



República Federativa do Brasil
Ministério da Economia
Instituto Nacional da Propriedade Industrial

(11) PI 0709986-0 B1



(22) Data do Depósito: 11/04/2007

(45) Data de Concessão: 15/10/2019

(54) Título: MÉTODO E SISTEMA PARA A ACELERAÇÃO DA ENTREGA DE UM AMBIENTE DE COMPUTAÇÃO PARA UM CLIENTE REMOTO

(51) Int.Cl.: H04L 29/08; H04L 29/06.

(52) CPC: H04L 67/06; H04L 67/34; H04L 67/2842; H04L 69/10; H04L 63/0272; (...).

(30) Prioridade Unionista: 12/04/2006 US 60/744,720.

(73) Titular(es): CITRIX SYSTEMS, INC..

(72) Inventor(es): BRAD J. PEDERSEN; PRABAKAR SUNDARRAJAN; RAJIV SINHA; TERRY TREDER.

(86) Pedido PCT: PCT US2007066433 de 11/04/2007

(87) Publicação PCT: WO 2007/121241 de 25/10/2007

(85) Data do Início da Fase Nacional: 10/10/2008

(57) Resumo: SISTEMAS E MÉTODOS PARA A ACELERAÇÃO DA ENTREGA DE UM AMBIENTE DE COMPUTAÇÃO PARA UM USUÁRIO REMOTO A presente invenção refere-se à aceleração de entrega de um ambiente de computação para um usuário remoto de um cliente em uma localização remota. O ambiente de computação pode incluir um aplicativo e um arquivo de dados usado ou processado pelo aplicativo. O aplicativo e o arquivo de dados podem ser armazenados ou providos através de um servidor remoto para o cliente. O usuário pode requisitar um ambiente de computação a partir do servidor que proveja a execução do aplicativo pelo usuário através do computador remoto. Por exemplo, o servidor pode transmitir continuamente o aplicativo para cliente remoto. O cliente e o servidor podem se comunicar através de uma aparelhagem que acelera as comunicações entre o cliente e o servidor. Por exemplo, a aparelhagem pode acelerar a transmissão contínua do aplicativo para o usuário remoto. Em alguns casos, a aparelhagem ou o usuário remoto também pode requisitar um arquivo de dados a partir do servidor, e a aparelhagem acelera a entrega do arquivo de dados para o usuário remoto. Como tal, os usuários em localizações remotas obtêm um acesso acelerado (...).

Relatório Descritivo da Patente de Invenção para
**"MÉTODO E SISTEMA PARA A ACELERAÇÃO DA ENTREGA DE
UM AMBIENTE DE COMPUTAÇÃO PARA UM CLIENTE REMOTO".**

Pedidos de Patente Relacionados

[001] O presente pedido reivindica o benefício de e a prioridade para o Pedido de Patente Provisória U.S. Nº 60/744.720, intitulado "Systems and Methods for Accelerating Delivery of a Computing Environment to a Remote User" e depositado em 12 de abril de 2006, o qual é incorporado aqui como referência.

Campo da Invenção

[002] A presente invenção refere-se a sistemas e métodos para aceleração da entrega de um ambiente de computação, incluindo um aplicativo e um arquivo de dados, para um usuário remoto de um cliente em uma localização remota para o servidor.

Antecedentes da Invenção

[003] A administração e o gerenciamento de ambientes de empresa consomem tempo, dinheiro e recursos. Em muitos casos, isto é porque o processo de gerenciamento de aplicativo e dados é descentralizado e de trabalho intenso. Por exemplo, uma porção significativa do tempo de um administrador pode ser gasta provendo mais armazenamento ou realizando backups para os dados corporativos, ou na atualização de servidores para lidarem com o crescimento nos dados corporativos. Também, um administrador pode precisar criar e aprovisionar novos servidores para lidarem com o crescimento nos dados. Adicionalmente, um administrador pode gastar tempo na atualização e no aprovisionamento de um servidor para um aplicativo de usuário em particular. Adicionalmente, uma porção significativa de dados corporativos pode residir fora do centro de dados corporativo. Por exemplo, documentos corporativos, arquivos e dados podem existir ou serem distribuídos para vários computadores

remotos do centro de dados.

[004] Em um esforço para redução de tempo, dinheiro e recursos requeridos para a administração e o gerenciamento de dados corporativos e aplicativos, muitas companhias consolidaram e centralizaram servidores, dados corporativos e aplicativos. Embora a consolidação e a centralização tenham reduzido alguns custos e produzido alguns benefícios, os dados e aplicativos centralizados introduzem desafios adicionais na provisão de acesso a dados e aplicativos. Um desses desafios envolve um usuário remoto tentando acessar um arquivo por uma conexão de rede de área ampla (WAN). Por exemplo, um usuário remoto em uma filial a qual tipicamente tem uma conexão de rede com o centro de dados corporativo que opera muito mais lentamente do que uma conexão de LAN pode tentar abrir pela WAN um documento do Microsoft Office armazenado em um centro de dados corporativo. O acesso do usuário remoto pela rede ao arquivo pode ser atrasado, devido à latência, à confiabilidade e à largura de banda com a WAN. Os atrasos podem ser maiores para arquivos maiores. Mais ainda, conforme a distância entre o usuário remoto e o centro de dados corporativo cresce, a frequência e a extensão de atrasos de rede no acesso a arquivos também podem aumentar. A adição de camadas de rede privada virtual, segurança e outras de rede na WAN pode reduzir mais a largura de banda disponível para os usuários remotos e aumentar os atrasos no acesso ao arquivo. A velocidade mais baixa e a largura de banda do escritório remoto podem causar atrasos inaceitáveis no acesso a arquivos remotos. Para se evitarem os atrasos em um acesso a arquivo remoto, os usuários remotos podem copiar e usar arquivos localmente, derrotando a finalidade de operações centralizadas. Adicionalmente, as conexões de WAN podem ser menos confiáveis do que as conexões de LAN, resultando em perda de pacote e desconexão de

rede. As interrupções de WAN podem ocorrer durante uma operação de arquivo, tal como um salvamento ou uma abertura de um documento, causando mais atrasos experimentados pelo usuário remoto.

[005] Portanto, são desejados sistemas e métodos para melhoria do acesso por usuários remotos a aplicativos e arquivos de dados centralizados, incluindo uma aceleração da entrega de aplicativos e arquivos de dados para usuários remotos.

Sumário da Invenção

[006] A presente invenção se refere a sistemas e métodos para a aceleração da entrega de um ambiente de computação de um aplicativo e arquivo de dados para um usuário remoto. O aplicativo e o arquivo de dados podem ser armazenados ou providos através de um servidor remoto para o cliente. Por exemplo, um usuário, tal como um empregado remoto, pode usar em uma filial um computador que não tenha o aplicativo e/ou o arquivo de dados disponíveis localmente. O usuário pode querer editar um documento corporativo com um aplicativo de processamento de texto não disponível no cliente remoto. O usuário pode requisitar um ambiente de computação a partir do servidor que proveja a execução do aplicativo desejado pelo usuário através do cliente remoto. Por exemplo, o servidor pode transmitir o aplicativo para o cliente remoto. O cliente remoto e o servidor podem se comunicar através de uma aplicação que acelere as comunicações entre o cliente remoto e o servidor. Por exemplo, a aplicação pode acelerar a transmissão do aplicativo para o usuário remoto. Em alguns casos, o aplicativo ou o usuário remoto também pode requisitar um arquivo de dados a partir do servidor, e a aplicação acelera a entrega do arquivo de dados para o usuário remoto. Como tal, a presente invenção provê a usuários em localizações remotas um acesso acelerado através de qualquer dispositivo conectado em rede a

aplicativos e arquivos de dados localizados remotamente do usuário.

[007] Em uma modalidade, a presente invenção está relacionada a um método para aceleração da entrega de um ambiente de computação de um aplicativo e um arquivo de dados para um usuário de um cliente em uma localização remota. O método inclui o recebimento pelo servidor de uma requisição a partir de um cliente remoto para execução de um aplicativo. O cliente remoto e o servidor se comunicam através de uma aplicação. O método também inclui a transmissão, pelo servidor, para o cliente remoto de um aplicativo para execução. O cliente transmite uma requisição para o servidor para um arquivo de dados usável pelo aplicativo, e a aplicação acelera a transmissão do arquivo de dados para o cliente remoto.

[008] Em uma modalidade da presente invenção, o método inclui a aceleração pela aplicação transmitindo o aplicativo para o cliente remoto. Em uma outra modalidade, a aplicação acelera a transmissão do arquivo de dados ou a transmissão dos aplicativos pela execução de uma das técnicas de aceleração a seguir: 1) compressão; 2) descompressão; 3) agrupamento de Protocolo de Controle de Transmissão; 4) multiplexação de Protocolo de Controle de Transmissão; 5) armazenamento em buffer de Protocolo de Controle de Transmissão; e 6) armazenamento em cache. Em uma outra modalidade, o método inclui a aceleração por um programa de aceleração no cliente remoto, de comunicações entre o cliente remoto e o servidor. Em algumas modalidades do método, a aplicação estabelece uma conexão de rede privada virtual ou uma conexão de Camada de Soquete Segura (SSL) com o cliente remoto. Em outras modalidades, o método inclui a aceleração, pela aplicação, da carga útil de um pacote de rede comunicado através de uma conexão de

camada de transporte entre um cliente remoto e o servidor.

[009] Em uma modalidade da presente invenção, o método inclui a transmissão pela aplicação de um programa de aceleração para o cliente remoto mediante uma requisição a partir do cliente remoto para o estabelecimento de uma conexão ou uma sessão com o servidor. Em algumas modalidades, o cliente remoto automaticamente instala e executa um programa de aceleração mediante o recebimento a partir da aplicação. Em outras modalidades, o método inclui a realização, por um programa de aceleração no cliente remoto, de uma das técnicas de aceleração a seguir: 1) compressão; 2) descompressão; 3) agrupamento de Protocolo de Controle de Transmissão; 4) multiplexação de Protocolo de Controle de Transmissão; 5) armazenamento em buffer de Protocolo de Controle de Transmissão; e 6) armazenamento em cache. Em algumas modalidades, o cliente remoto executa o programa de aceleração de forma transparente para o aplicativo ou o servidor.

[0010] Em algumas modalidades da presente invenção, o método inclui a determinação pela aplicação que o aplicativo é capaz de ser acelerado e a transmissão em resposta à determinação de um programa de aceleração para o cliente remoto. Em outras modalidades, a aplicação armazena em cache o arquivo de dados. Em uma modalidade, a aplicação intercepta a requisição para o arquivo de dados e transmite para o cliente remoto o arquivo de dados armazenado em cache em resposta à requisição.

[0011] Em um outro aspecto, a presente invenção está relacionada a um sistema para a aceleração da entrega para um usuário remoto de um ambiente de computação de um aplicativo e um arquivo de dados para um cliente em uma localização remota. O sistema inclui uma aplicação para aceleração das comunicações entre um ou mais clientes remotos e um ou mais servidores. O sistema também inclui

um servidor recebendo uma requisição a partir de um cliente remoto para execução de um aplicativo. O cliente remoto e o servidor se comunicam através da aplicação. O servidor transmite para o cliente remoto um aplicativo para execução. O cliente remoto transmite uma requisição para o servidor para um arquivo de dados usável pelo aplicativo, e a aplicação acelera a transmissão do artigo absorvente descartável para o cliente remoto.

[0012] Em algumas modalidades da presente invenção, a aplicação acelera a transmissão do aplicativo para o cliente remoto. Em uma modalidade, a aplicação acelera a transmissão do arquivo de dados ou a transmissão do aplicativo pela execução de uma das técnicas de aceleração a seguir: 1) compressão; 2) descompressão; 3) agrupamento de Protocolo de Controle de Transmissão; 4) multiplexação de Protocolo de Controle de Transmissão; 5) armazenamento em buffer de Protocolo de Controle de Transmissão; e 6) armazenamento em cache. Em uma outra modalidade, o sistema inclui um programa de aceleração no cliente remoto acelerando as comunicações entre o cliente remoto e o servidor. Em uma modalidade, a aplicação estabelece uma conexão de rede privada virtual ou uma conexão de Camada de Soquete Segura (SSL) com o cliente remoto.

[0013] Em algumas modalidades do sistema da presente invenção, a aplicação acelera uma carga útil de um pacote de rede comunicado através de uma conexão de camada de transporte entre o cliente remoto e o servidor. Em uma modalidade, a aplicação transmite um programa de aceleração para o cliente remoto mediante uma requisição a partir do cliente para o estabelecimento de uma conexão ou uma sessão com o servidor. Em outras modalidades, o cliente remoto automaticamente instala e executa um programa de aceleração mediante o recebimento a partir da aplicação.

O programa de aceleração no cliente remoto pode executar uma ou mais das técnicas de aceleração a seguir: 1) compressão; 2) descompressão; 3) agrupamento de Protocolo de Controle de Transmissão; 4) multiplexação de Protocolo de Controle de Transmissão; 5) armazenamento em buffer de Protocolo de Controle de Transmissão; e 6) armazenamento em cache. Em uma modalidade, o cliente remoto executa o programa de aceleração de forma transparente para o aplicativo ou o servidor.

[0014] Em uma outra modalidade do sistema da presente invenção, a aplicação determina que o aplicativo é capaz de ser acelerado, e transmite um programa de aceleração para o cliente remoto em resposta à determinação. Em uma modalidade, a aplicação compreende um cache para armazenamento em cache do arquivo de dados. Em algumas modalidades, a aplicação intercepta a requisição para o arquivo de dados e transmite para o cliente remoto o arquivo de dados armazenado em cache, em resposta à requisição.

Breve Descrição dos Desenhos

[0015] Estes e outros aspectos desta invenção serão prontamente evidentes a partir da descrição detalhada abaixo e dos desenhos em apenso, os quais têm por significado ilustrarem e não limitarem a invenção, e nos quais:

[0016] a figura 1A é um diagrama de blocos que descreve um ambiente de rede;

[0017] a figura 1B é um diagrama de blocos que descreve uma modalidade de um ambiente de computação de um remoto em um ambiente de rede;

[0018] as figura 1C e 1D são diagramas de blocos descrevendo modalidades de computadores úteis em relação a modalidades descritas;

[0019] a figura 1E é um diagrama de blocos que descreve um

ambiente adequado para a entrega de um ambiente de computação a um cliente;

[0020] a figura 1F é um diagrama de blocos que descreve uma modalidade de um sistema para a provisão de uma pluralidade de programas aplicativos disponíveis para a máquina local através da publicação de GUIs em um diretório de serviço da web;

[0021] a figura 2 é um fluxograma que descreve uma modalidade das etapas seguidas para a seleção de um método de execução de um programa aplicativo;

[0022] a figura 3A é um diagrama de blocos que descreve uma modalidade de uma máquina local iniciando uma execução de um aplicativo de Vizinhança de Programa através da Rede Mundial;

[0023] a figura 3B é um fluxograma que descreve uma modalidade das etapas seguidas por uma máquina local para acessar um programa aplicativo enumerado usando-se um diretório de serviço da web;

[0024] a figura 4A é um diagrama de blocos de uma modalidade de uma rede provendo acesso baseado em política a programas aplicativos para uma máquina local;

[0025] a figura 4B é um diagrama de blocos que descreve uma modalidade mais detalhada de um agente de política;

[0026] a figura 4C é um fluxograma que descreve uma modalidade das etapas seguidas por um agente de política para tomar uma decisão de controle de acesso com base em uma informação recebida sobre uma máquina local;

[0027] a figura 4D é um diagrama de blocos que descreve uma modalidade de uma rede de computadores na qual um acesso remoto autorizado a uma pluralidade de sessões de aplicativo é provido;

[0028] a figura 4E é um fluxograma que descreve uma modalidade das etapas seguidas por um servidor de sessão para conexão de uma

máquina local com suas sessões de aplicativo associadas;

[0029] a figura 5 é um fluxograma que descreve uma modalidade das etapas seguidas por um servidor de sessão para conexão de um nó cliente com suas sessões de aplicativo associadas;

[0030] a figura 6 é um diagrama de blocos que descreve uma modalidade de uma máquina remota incluindo um serviço de gerenciamento provendo uma enumeração de aplicativo;

[0031] a figura 7 é um fluxograma que descreve uma modalidade das etapas seguidas para acesso a uma pluralidade de arquivos compreendendo um programa aplicativo;

[0032] a figura 8A é um diagrama de blocos que descreve uma modalidade de um computador rodando sob o controle de um sistema operacional que tem compatibilidade de aplicativo reduzida e problemas de sociabilidade de aplicativo;

[0033] a figura 8B é um diagrama de blocos que descreve um programa de usuário múltiplo que tem compatibilidade de aplicativo reduzida e problemas de sociabilidade de aplicativo;

[0034] a figura 8C é um fluxograma que descreve uma modalidade das etapas seguidas em um método para associação de um processo a um escopo de isolamento;

[0035] a figura 9 é um fluxograma que descreve uma modalidade de etapas seguidas em um método para execução de um programa aplicativo;

[0036] a figura 10 é um fluxograma que descreve uma modalidade de uma pluralidade de arquivos aplicativos residentes em uma máquina remota;

[0037] a figura 11 é um fluxograma que descreve uma modalidade das etapas seguidas em um método para responder localmente a requisições por metadados de arquivo associados aos arquivos armazenados remotamente;

[0038] a figura 12 é um diagrama de blocos que descreve uma modalidade de um sistema para se responder localmente a requisições por metadados de arquivo associados a arquivos armazenados remotamente;

[0039] a figura 13 é um fluxograma que descreve uma modalidade das etapas seguidas em um método para acesso a um arquivo remoto em uma estrutura de diretório associada a um programa aplicativo em execução localmente;

[0040] a figura 14 é um diagrama de blocos que descreve uma modalidade de um sistema para acesso de um arquivo em uma estrutura de diretório associada a um aplicativo;

[0041] a figura 15 é um diagrama de blocos de uma modalidade de uma máquina remota incluindo um subsistema de gerenciamento de licença;

[0042] a figura 16 é um diagrama de blocos que descreve uma modalidade de componentes em um serviço de gerenciamento em uma máquina remota;

[0043] a figura 17 é um fluxograma que descreve uma modalidade das etapas seguidas para se requisitar e manter uma licença a partir de uma máquina remota;

[0044] a figura 18 é um diagrama de blocos que descreve uma modalidade de estados que podem ser associados a uma sessão monitorada por um serviço de gerenciamento;

[0045] a figura 19 é um diagrama de blocos que descreve um pacote incluindo dois alvos, cada alvo compreendendo uma pluralidade de arquivos aplicativos compreendendo um aplicativo;

[0046] a figura 20 é um fluxograma que descreve uma modalidade das etapas seguidas em um método baseado em política para instalação de um programa aplicativo sem rebotar um sistema operacional;

[0047] a figura 21 é um fluxograma que descreve uma modalidade das etapas seguidas em um método baseado em política para instalação de um programa aplicativo sem rebutar um sistema operacional;

[0048] a figura 22 é um instantâneo de tela que descreve uma modalidade de uma enumeração de scripts a serem executados na máquina local;

[0049] a figura 23 é um diagrama de blocos que descreve uma modalidade de um sistema que inclui um mecanismo de formação de pacote executando um programa instalador em um ambiente de isolamento;

[0050] a figura 24 é um fluxograma que descreve uma modalidade das etapas seguidas em um ambiente no qual uma execução de um programa instalador requer rebutar um sistema operacional;

[0051] a figura 25 é um diagrama de blocos que descreve uma modalidade de uma máquina remota na qual um mecanismo de formação de pacote instala um programa aplicativo;

[0052] a figura 26 é um fluxograma que descreve uma modalidade das etapas seguidas para instalação de um aplicativo em um ambiente de isolamento de aplicativo;

[0053] a figura 27 é um diagrama de blocos que ilustra uma modalidade de uma arquitetura de uma aplicação que realiza um armazenamento em cache integrado;

[0054] a figura 28A é um fluxograma de etapas seguidas em uma modalidade de um método para integração de operações de dispositivo com um processamento de pacote e o temporizador de processamento de pacote;

[0055] a figura 28B é um fluxograma de etapas seguidas em uma modalidade de um método para a prática de técnicas de granularidade de invalidação tendo em vista a figura 3A;

[0056] a figura 29A é um fluxograma de etapas seguidas em uma modalidade de um método usando comandos de invalidação para invalidação de objetos vencidos;

[0057] a figura 29B é um fluxograma de etapas seguidas em uma modalidade de um método que incorpora uma invalidação de grupos de objetos;

[0058] a figura 29C é um fluxograma de etapas seguidas em uma modalidade de um método que incorpora uma invalidação de grupos de objetos usando-se determinantes de objeto;

[0059] a figura 30 é um fluxograma de etapas seguidas em uma modalidade de um método de conexão de agrupamento;

[0060] a figura 31 é um fluxograma de etapas seguidas em uma modalidade de um método de tradução de requisições de cliente e servidor;

[0061] a figura 32 ilustra uma modalidade de um parâmetro de comprimento de conteúdo;

[0062] a figura 33 ilustra uma modalidade de campos de tamanho de fragmento;

[0063] a figura 34 é um fluxograma de mensagem que descreve uma modalidade de agrupamento de conexão;

[0064] a figura 35 é um fluxograma detalhado que ilustra uma modalidade das etapas seguidas para uso do parâmetro de comprimento de conteúdo para aumento da eficiência de agrupamento de conexão entre clientes e servidores;

[0065] a figura 36 é um fluxograma que descreve uma modalidade das etapas seguidas para uso do parâmetro de comprimento de conteúdo para aumento da eficiência de agrupamento de conexão entre clientes e servidores;

[0066] a figura 37 é um fluxograma detalhado que ilustra uma modalidade das etapas seguidas para uso de campos de tamanho de

fragmento para aumento da eficiência de conexão entre clientes e servidores;

[0067] a figura 38 é um fluxograma que descreve uma modalidade das etapas seguidas para uso de campos de tamanho de fragmento para aumento da eficiência de conexão entre clientes e servidores;

[0068] a figura 39 é um fluxograma de uma modalidade das etapas seguidas para a provisão de uma funcionalidade de armazenamento em cache integrada;

[0069] a figura 40A é um diagrama de blocos de uma modalidade de um programa de aceleração de lado de cliente;

[0070] a figura 40B é um diagrama de blocos de uma modalidade de uma aplicação para a provisão de um programa de aceleração de lado de cliente;

[0071] a figura 41A é um diagrama de etapa de uma modalidade de um método para a provisão dinâmica e a instalação automática e a execução de um programa de aceleração de lado de cliente;

[0072] a figura 41B é um diagrama de etapa de uma modalidade de um método para se determinar que um aplicativo pode ser acelerado;

[0073] a figura 41C é um diagrama de etapa de uma outra modalidade de um método de execução de uma pluralidade de técnicas de arquitetura de entrega de conteúdo pelo programa de aceleração para interceptação na camada de transporte e usando-se uma estrutura de dados de nível de kernel;

[0074] a figura 42A é um diagrama de etapa de uma outra modalidade de um método para a instalação automática e a execução do programa de aceleração no cliente através de um primeiro programa;

[0075] a figura 42B é um diagrama de etapa de uma modalidade de um método para um primeiro programa e do programa de

aceleração para a provisão de uma conectividade de rede privada virtual e a execução de uma ou mais técnicas de aceleração;

[0076] a figura 43 é um diagrama de etapa de uma modalidade de um método para redirecionamento de uma com de cliente para um soe para se desviar de um intermediário determinado não usável para transmissão da comunicação para o servidor;

[0077] a figura 44 é um diagrama de etapa de uma modalidade de um método para a execução de uma técnica de aceleração de lado de cliente de armazenamento em buffer de protocolo de controle de transporte;

[0078] a figura 45A é um diagrama de etapa de uma modalidade de um método para a execução de uma técnica de aceleração de lado de cliente de agrupamento de conexão de protocolo de controle de transporte;

[0079] a figura 45B é uma vista diagramática de um conjunto de transações de HTTP realizadas por uma pluralidade de aplicativos através de um grupo de uma ou mais conexões de camada de transporte em uma modalidade;

[0080] a figura 46 é um diagrama de etapa de uma modalidade de um método para a execução de uma técnica de aceleração de lado de cliente de multiplexação de protocolo de controle de transporte;

[0081] a figura 47 é uma vista diagramática de uma modalidade de um identificador de comprimento de conteúdo de um pacote de camada de transporte;

[0082] a figura 48 é uma vista diagramática de uma outra modalidade de um identificador de comprimento de conteúdo de uma mensagem transmitida através de múltiplos fragmentos;

[0083] a figura 49A é um diagrama de blocos que descreve uma modalidade de exemplo de um sistema de computador em rede para aceleração da entrega de um ambiente de computação para um

cliente remoto; e

[0084] a figura 49B é um fluxograma que descreve uma modalidade de etapas de um método para a aceleração da entrega de um ambiente de computação para um cliente remoto.

Descrição Detalhada da Invenção

[0085] Para fins de leitura da descrição das várias modalidades abaixo, as descrições a seguir das seções do relatório descritivo e seus respectivos conteúdos pode ser útil:

- a Seção A descreve um ambiente de rede e um ambiente de computação os quais podem ser úteis para a prática de modalidades descritas aqui;

- a Seção B descreve modalidades de sistemas e métodos para a entrega de um ambiente de computação processador uma usuário remoto;

- a Seção C descreve modalidades de sistemas e métodos para aceleração das comunicações entre um cliente e um servidor; e

- a Seção D descreve uma modalidade de exemplo ilustrativa da aceleração da entrega de um ambiente de computação para um usuário remoto usando os sistemas e métodos descritos aqui.

A. AMBIENTE DE REDE E DE COMPUTAÇÃO

[0086] Antes da discussão das especificidades da presente invenção, pode ser útil discutir alguns dos ambientes de rede nos quais a modalidade ilustrativa da presente invenção pode ser empregada. Com referência, agora, à Figura 1A, um ambiente de rede 5 é descrito. Em uma breve visão geral, o ambiente de rede 5 compreende um ou mais clientes 10-10'' (geralmente referidos como os clientes 10 ou as máquinas locais 10) em comunicação com um ou mais servidores 30-30'' (também referidos geralmente como os servidores 30 ou as máquinas remotas 30) através de uma ou mais redes 40, 40''. Em algumas modalidades, um cliente 10 se comunica

com um servidor 30 através de uma aplicação 1250.

[0087] Embora a figura 1A mostre uma rede 40 e uma rede 40' entre os clientes 10-10'-10'' e os servidores 30-30'', os clientes 10-10' e os servidores 30-30'' podem estar na mesma rede 40. As redes 40 e 40' podem ser do mesmo tipo de rede ou de tipos diferentes de redes. A rede 40 e/ou a rede 40' podem ser uma rede de área local (LAN), tal como uma Intranet de companhia, uma rede de área metropolitana (MAN) ou uma rede de área ampla (WAN), tal como a Internet ou a Rede Mundial. Em uma modalidade, a rede 40' pode ser uma rede privada e a rede 40 pode ser uma rede pública. Em uma outra modalidade, as redes 40 e 40' podem ser ambas redes privadas. Em algumas modalidades, os clientes 10-10'' podem estar localizados em uma filial de uma empresa corporativa em comunicação através de uma conexão de WAN pela rede 40 com os servidores 30-30'' localizados em um centro de dados corporativo.

[0088] A rede 40 e/ou 40' pode ser qualquer tipo e/ou forma de rede, e pode incluir qualquer uma das seguintes: uma rede de ponto a ponto, uma rede de difusão, uma rede de área ampla, uma rede de área local, uma rede de telecomunicações, uma rede de comunicação de dados, uma rede de computadores, uma rede de ATM (Modo de Transferência Assíncrono), uma rede SONET (Rede Ótica Síncrona), uma rede de SDH (Hierarquia Digital Síncrona), uma rede sem fio e uma rede com fio. A topologia da rede 40 e/ou 40' pode ser uma topologia de rede de barramento, de estrela ou de anel. A rede 40 e/ou 40' e a topologia de rede podem ser qualquer uma dessas redes ou topologia de rede, conforme conhecido por aqueles versados na técnica, capazes de suportarem as operações descritas aqui.

[0089] Conforme mostrado na figura 1A, a aplicação 1250 (também referida aqui como uma unidade de interface 1250) é mostrada entre as redes 40 e 40'. Em algumas modalidades, a

aplicação 1250 pode estar localizada na rede 40. Por exemplo, uma filial de uma empresa corporativa pode empregar uma aplicação 1250 na filial. Em outras modalidades, a aplicação 1250 pode estar localizada na rede 40'. Por exemplo, uma aplicação 1250 pode estar localizada em um centro de dados corporativo. Ainda em uma outra modalidade, uma pluralidade de aparelhagens 1250 pode ser empregada na rede 40. Em algumas modalidades, uma pluralidade de aparelhagens 1250 pode ser empregada na rede 40'. Em uma modalidade, uma primeira aplicação 1250 se comunica com uma segunda aplicação 1250'. Em outras modalidades, a aplicação 1250 poderia ser parte de qualquer cliente 10-10' ou servidor 30-30'' na mesma rede ou em uma diferente 40, 40' como o cliente 10-10''. Uma ou mais aparelhagens 1250 podem estar localizadas em qualquer ponto na rede ou em caminhos de comunicações de rede entre um cliente 10-10'' e um servidor 30-30''.

[0090] Em uma modalidade, o sistema pode incluir múltiplas máquinas remotas logicamente agrupadas 30, uma ou mais das quais estando disponível para execução de aplicativos em nome de uma máquina local 10. Nestas modalidades, o grupo lógico de máquinas remotas pode ser referido como um banco de servidores 38 ou um banco 38. Em algumas destas modalidades, as máquinas remotas 30 podem ser geograficamente dispersas. Um banco 38 pode ser administrado como uma entidade única.

[0091] As máquinas remotas 30 dentro de cada banco 38 podem ser heterogêneas. Isto é, uma ou mais das máquinas remotas 30 podem operar de acordo com um tipo de plataforma de sistema operacional (por exemplo, WINDOWS NT, fabricado pela Microsoft Corp. de Redmond, Washington), enquanto uma ou mais das outras máquinas remotas 30 podem operar de acordo com um outro tipo de plataforma de sistema operacional (por exemplo, Unix ou Linux). As

máquinas remotas 30 compreendendo cada banco 38 não precisam estar fisicamente próximas de cada outra máquina remota 30 neste banco 38. Assim, o grupo de máquinas remotas 30 logicamente agrupado como o banco 38 pode ser interconectado usando-se uma conexão de rede de área ampla (WAN) ou uma conexão de rede de área média (MAN). Por exemplo, um banco 38 pode incluir máquinas remotas 30 fisicamente localizadas em continentes diferentes ou em regiões diferentes de um continente, país, estado, cidade, campus ou sala. As velocidades de transmissão de dados entre as máquinas remotas 30 no banco 38 podem ser aumentadas, se as máquinas remotas 30 forem conectadas usando-se uma conexão de rede de área local (LAN) ou alguma forma de conexão direta.

[0092] As máquinas remotas 30 podem ser referidas como servidores, servidores de arquivo, servidores de aplicativo ou máquinas remotas. Em algumas modalidades, as máquinas remotas 30 podem ter a capacidade de funcionarem como servidores de aplicativo ou como um servidor de aplicativo mestre. Em uma modalidade, uma máquina remota 30 pode incluir um Diretório Ativo. As máquinas locais 10 também podem ser referidas como nós clientes ou pontos finais. Em algumas modalidades, as máquinas locais 10 têm a capacidade de funcionarem como nós clientes buscando acesso a aplicativos e como servidores de aplicativo provendo acesso a aplicativos hospedados para outras máquinas locais 10.

[0093] Em uma modalidade, a máquina local 10 se comunica diretamente com uma das máquinas remotas 30 em um banco 38. Em uma outra modalidade, a máquina local 10 executa um aplicativo de vizinhança de programa para comunicação com a máquina remota 30 em um banco 38. Ainda em uma outra modalidade, a máquina remota 30 provê a funcionalidade de um nó mestre. Em algumas modalidades, a máquina local 10 se comunica com a máquina remota 30 no banco

38 através de uma rede 40. Pela rede 40, a máquina local 10 pode, por exemplo, requisitar uma execução de vários aplicativos hospedados pelas máquinas remotas 30, 30', 30'' e 30''' no banco 38 e receber uma saída de resultados da execução de aplicativo para exibição. A rede 40 pode compreender conexões síncronas ou assíncronas e pode ser uma LAN, uma MAN (Rede de Área Média) ou uma WAN. Adicionalmente, ímã rede 40 pode compreender um enlace sem fio, tal como um canal de infravermelho ou uma banda de satélite. Em algumas modalidades, apenas o nó mestre provê a funcionalidade requerida para a identificação e a provisão de uma informação de endereço associada a uma máquina remota 30' hospedando um aplicativo requisitado.

[0094] Em algumas modalidades, uma máquina local 10 se comunica com uma máquina remota 30'''. Em uma destas modalidades, a máquina remota 30''' provê uma funcionalidade de um servidor da web. Em uma outra destas modalidades, a máquina remota 30''' recebe requisições da máquina local 10, encaminha as requisições para uma máquina remota 30 e responde à requisição pela máquina local 10 com uma resposta à requisição a partir da máquina remota 30. Ainda em uma outra destas modalidades, a máquina remota 30 adquire uma enumeração de aplicativos disponíveis para a máquina local 10 e uma informação de endereço associada a uma máquina remota 30' hospedando um aplicativo identificado pela enumeração de aplicativos. Ainda em uma outra destas modalidades, a máquina remota 30''' apresenta a resposta à requisição para a máquina local 10 usando uma interface da web. Em uma modalidade, a máquina local 10 se comunica diretamente com a máquina remota 30' para acessar o aplicativo identificado. Em uma outra modalidade, a máquina local 10 recebe dados de saída de aplicativo a partir da máquina remota 30''', os dados de saída de aplicativo gerados por

uma execução do aplicativo identificado na máquina remota 30'.

[0095] Com referência, agora, à figura 1B, um ambiente de rede para entrega e/ou operação de um ambiente de computação em um cliente 10 é descrito. Em uma visão geral breve, um servidor 30 inclui um sistema de entrega de aplicativo 500 para entrega de um ambiente de computação ou um aplicativo e arquivo de dados para um ou mais clientes. O cliente 10 pode incluir um ambiente de computação 15 para execução de um aplicativo que usa ou processa um arquivo de dados. O cliente 10 em comunicação com o servidor 30 através das redes 40, 40' e da aplicação 1250 pode requisitar um aplicativo e um arquivo de dados a partir do servidor 30, ou a aplicação 1250 pode encaminhar uma requisição a partir do cliente 10 para o servidor 30. Por exemplo, o cliente 10 pode não ter localmente o aplicativo e o arquivo de dados armazenados ou acessíveis localmente. Em resposta à requisição, o servidor 30 pode entregar o aplicativo e o arquivo de dados para o cliente 10. Por exemplo, em uma modalidade, o servidor 30 pode transmitir o aplicativo como uma transmissão de aplicativo para operação no ambiente de computação 15 no cliente 10.

[0096] As Figuras 1C e 1D são diagramas de blocos que descrevem modalidades da arquitetura do computador de finalidade geral 135 útil como dispositivos de computação clientes 10 e dispositivos de computação servidores 30. Conforme mostrado nas figuras 1C e 1D, cada computador 135 inclui uma unidade de processamento central 102 e uma unidade de memória principal 122. Cada computador 135 também pode incluir outros elementos opcionais, tais como um ou mais dispositivos de entrada/saída 130a – 130b (geralmente referidos usando-se o número de referência 130), e uma memória cache 140 em comunicação com a unidade de processamento central 102.

[0097] A unidade de processamento central 102 é qualquer circuito

lógico que responda a e processe instruções buscadas a partir da unidade de memória principal 122. Em muitas modalidades, a unidade de processamento central é provida por uma unidade de microprocessador, tais como aquelas fabricadas pela Intel Corporation de Mountain View, Califórnia; aquelas fabricadas pela Motorola Corporation de Schaumburg, Illinois; as linhas Crusoe e Efficeon de processadores fabricadas pela Transmeta Corporation, de Santa Clara, Califórnia; as linhas de processadores fabricadas pela International Business Machines de White Plains, New York; ou as linhas de processadores fabricadas pela Advanced Micro Devices de Sunnyvale, Califórnia.

[0098] A unidade de memória principal 122 pode ser um ou mais chips de memória capazes de armazenarem dados e permitindo que qualquer localização de armazenamento seja diretamente acessada pelo microprocessador 102, tal como uma memória de acesso randômico Estática (SRAM), uma SRAM de Rajada ou uma SRAM de Rajada Síncrona (BSRAM), uma memória de acesso randômico Dinâmica (DRAM), uma DRAM de Modo de Radiochamada Rápida (FDM DRAM), uma DRAM Melhorada (EDRAM), uma RAM de Saída de Dados Estendida (EDO RAM), uma DRAM de Saída de Dados Estendida (EDO DRAM), uma DRAM de Saída de Dados Estendida de Rajada (BEDO DRAM), uma DRAM Estendida (EDRAM), uma DRAM síncrona (SDRAM), JEDEC SRAM, PC100 SDRAM, SDRAM de Taxa de Dados Dupla (DDR SDRAM), SDRAM Melhorada (ESDRAM), DRAM de Enlace Síncrono (SLDRAM), DRAM Rambus Direta (DRDRAM) ou RAM Ferroelétrica (FRAM). Na modalidade mostrada na figura 1C, o processador 102 se comunica com a memória principal 122 através de um barramento de sistema 120 (descrito em maiores detalhes abaixo). A figura 1B descreve uma modalidade de um sistema de computador 135 no qual o processador se comunica

diretamente com a memória principal 122 através de uma porta de memória. Por exemplo, na figura 1C, a memória principal 122 pode ser uma DRDRAM.

[0099] As figura 1C e 1D descrevem modalidades nas quais o processador principal 102 se comunica diretamente com a memória cache 140 através de um barramento secundário, referido às vezes como barramento "backside". Em outras modalidades, o processador principal 102 se comunica com a memória cache 140 usando o barramento de sistema 120. A memória cache 140 tipicamente tem um tempo de resposta mais rápido do que a memória principal 122 e tipicamente é provida por SRAM, BSRAM ou EDRAM.

[00100] Na modalidade mostrada na figura 1C, o processador 102 se comunica com vários dispositivos de I/O 130 através do barramento de sistema local 120. Vários barramentos podem ser usados para a conexão da unidade de processamento central 102 aos dispositivos de I/O 130, incluindo um barramento VESA VL, um barramento ISA, um barramento EISA, um barramento de Arquitetura de Microcanal (MCA), um barramento PCI, um barramento PCI-X, um barramento PCI-Express ou um NuBus. Para modalidades nas quais o dispositivo de I/O é um visor de vídeo, o processador 102 pode usar uma Porta Gráfica Avançada (AGP) para comunicação com o visor. A figura 1D descreve uma modalidade de um sistema de computador 135 no qual o processador principal 102 se comunica diretamente com o dispositivo de I/O 130b através de HyperTransport, Rapid I/O ou InfiniBand. A figura 1C também descreve uma modalidade na qual os barramentos locais e a comunicação direta são misturados: o processador 102 se comunica com o dispositivo de I/O 130a usando um barramento de interconexão local enquanto se comunica com o dispositivo de I/O 130b diretamente.

[00101] Uma ampla variedade de dispositivos de I/O 130 pode estar

presente no sistema de computador 135. Os dispositivos de entrada incluem teclados, mouses, "trackpads", "trackballs", microfones e mesas de desenho. Os dispositivos de saída incluem visores de vídeo, alto-falantes, impressoras de jato de tinta, impressoras a laser, e impressoras de sublimação de corante. Um dispositivo de I/O também pode prover um armazenamento de massa para o sistema de computador 135, tais como uma unidade de disco rígido, uma unidade de disco flexível para o recebimento de disquetes tais como discos de 3,5", 5,25" ou discos ZIP, uma unidade de CD-ROM, uma unidade de CD-R/RW, uma unidade de DVD-ROM, unidades de fita de vários formatos, e dispositivos de armazenamento USB, tal como uma linha de dispositivos USB Flash Drive fabricados pela Twintech Industry, Inc. de Los Alamitos, Califórnia.

[00102] Em modalidades adicionais, um dispositivo de I/O 130 pode ser uma ponte entre o barramento de sistema 120 e um barramento de comunicação externo, tal como um barramento USB, um Barramento Apple Desktop, uma conexão serial RS-232, um barramento SCSI, um barramento FireWire, um barramento FireWire 800, um barramento de Ethernet, um barramento AppleTalk, um barramento de Gigabit Ethernet, um barramento de Modo de Transferência Assíncrono, um barramento HIPPI, um barramento Super HIPPI, um barramento SerialPlus, um barramento SCI/LAMP, um barramento FibreChannel ou um barramento de interface de sistema de computador pequeno Serial Attached.

[00103] Os computadores de mesa de finalidade geral do tipo descrito na figura 1C e na figura 1D tipicamente operam sob o controle de sistemas operacionais, os quais controlam a programação de tarefas e o acesso a recursos do sistema. Os sistemas operacionais típicos incluem: MICROSOFT WINDOWS, fabricado pela Microsoft Corp. de Redmond, Washington; MacOS, fabricado pela Apple

Computer, Inc. de Cupertino, Califórnia; OS/2, fabricado pela International Business Machines de Armonk, New York; e Linux, um sistema operacional disponível livremente pela Caldera Corp. de Salt Lake City, Utah, dentre outras.

[00104] Para modalidades nas quais uma máquina de cliente 10 ou um servidor 30 compreende um dispositivo móvel, o dispositivo pode ser um telefone celular com JAVA habilitado, tais como i55sr, i58sr, i85s, ou o i88s, todos os quais sendo fabricados pela Motorola Corp. de Schaumburg, Illinois; o 6035 ou 7135 fabricado pela Kyocera de Kyoto, Japão; ou o i300 ou i330 fabricado pela Samsung Electronics Co., Ltd., de Seul, Coréia. Em outras modalidades compreendendo dispositivos móveis, um dispositivo móvel pode ser um assistente digital pessoal (PDA) operando sob o controle do sistema operacional PalmOS, tais como o Tungsten W, o VII, o VIIx, o i705, todos os quais sendo fabricados pela palmOne, Inc. de Milpitas, Califórnia. Em outras modalidades, o cliente 113 pode ser um assistente digital pessoal (PDA) operando sob o controle do sistema operacional PocketPC, tais como iPAQ 4155, iPAQ 5555, iPAQ 1945, iPAQ 2215, e iPAQ 4255, todos os quais fabricados pela Hewlett-Packard Corporation de Palo Alto, Califórnia; o ViewSonic V36, fabricado pela ViewSonic de Walnut, Califórnia; ou o Toshiba PocketPC e405, fabricado pela Toshiba America, Inc. de New York, New York. Ainda em outras modalidades, o dispositivo móvel é uma combinação de PDA/dispositivo de telefone, tais como os Treo 180, Treo 270, Treo 600, Treo 650, ou o Treo 700w, todos os quais sendo fabricados pela palmOne, Inc. de Milpitas, Califórnia. Ainda em outras modalidades, o dispositivo móvel é um telefone celular que opera sob o controle do sistema operacional PocketPC, tal como o MPx200, fabricado pela Motorola Corp. Um dispositivo móvel típico pode compreender muitos dos elementos descritos acima nas figuras 1C e 1D, incluindo o processador 102 e a

memória principal 104.

B. SISTEMAS E MÉTODOS PARA ENTREGA DE UM AMBIENTE DE COMPUTAÇÃO

[00105] Uma modalidade é dirigida a sistemas e métodos para a entrega de um ambiente de computação para um usuário remoto em um cliente 10 localizado em uma localização remota do servidor 30. Embora os métodos e sistemas nesta seção geralmente falem de servidores 30, os métodos e sistema abaixo podem utilizar os servidores 30, as aparelhagens de rede 1250, ou qualquer combinação dos mesmos.

[00106] Com referência, agora, à figura 1E, uma modalidade de um sistema na qual as máquinas remotas 30 compreendem um banco 38, conforme descrito na figura 1A, é mostrada. Cada máquina remota 30 inclui uma interface de lado de rede 202 e uma interface de lado de banco 204. A interface de lado de rede 202 da máquina remota 30 pode estar em comunicação com uma ou mais máquinas locais 10 ou uma rede 210. A rede 210 pode ser uma WAN, uma LAN ou uma rede internacional, tal como a Internet ou a Rede Mundial. As máquinas locais 10 podem estabelecer conexões com as máquinas remotas 30 usando a rede 210.

[00107] As interfaces de lado de banco 204 das máquinas remotas 30 são interconectadas umas às outras por enlaces de comunicação 200, de modo que as máquinas remotas 30 possam se comunicar umas com as outras. Em cada máquina remota 30, a interface de lado de banco 204 se comunica com a interface de lado de rede 202. As interfaces de lado de banco 204 também se comunicam (designado pelas setas 220) com um armazenamento persistente 230 e, em algumas modalidades, com um armazenamento dinâmico 240. A combinação de máquinas remotas 30, do armazenamento persistente 230 e do armazenamento dinâmico 240, quando provido, é

coletivamente referida como um banco 38. Em algumas modalidades, uma máquina remota 30 se comunica com o armazenamento persistente 230 e as outras máquinas remotas 30' se comunicam com a máquina remota 30 para acessar uma informação armazenada no armazenamento persistente.

[00108] O armazenamento persistente 230 pode ser fisicamente implementado em um disco, um banco de disco, um arranjo redundante de discos independentes (RAID), um disco compacto gravável ou qualquer outro dispositivo que permita que dados sejam lidos e escritos e que mantenha dados escritos, se a potência for removida do dispositivo de armazenamento. Um dispositivo físico único pode prover armazenamento para uma pluralidade de armazenamentos persistentes, isto é, um único dispositivo físico pode ser usado para a provisão do armazenamento persistente 230 para mais de um banco 38. O armazenamento persistente 230 mantém dados estáticos associados a cada máquina remota 30 no banco 38 e dados globais usados por todas as máquinas remotas 30 no banco 38. Em uma modalidade, o armazenamento persistente 230 pode manter os dados de máquina remota em um modelo de dados de Protocolo de Acesso de Diretório de Peso Leve (LDAP). Em outras modalidades, o armazenamento persistente 230 armazena dados de máquina remota em um banco de dados em conformidade com ODBC. Para as finalidades desta descrição, o termo "dados estáticos" se refere àquilo que não muda freqüentemente, isto é, dados que mudam apenas em uma base horária, diária ou semanal, ou dados que não mudam nunca. Cada máquina remota usa um subsistema de armazenamento persistente para a leitura de dados e escrita de dados no armazenamento persistente 230.

[00109] Os dados armazenados no armazenamento persistente 230 podem ser replicados para fins de confiabilidade de forma física ou

lógica. Por exemplo, uma redundância física pode ser provida usando-se um conjunto de discos espelhados redundantes, cada um provendo uma cópia de dados. Em outras modalidades, o banco de dados em si pode ser replicado usando-se técnicas padronizadas de banco de dados para a provisão de múltiplas cópias do banco de dados. Em modalidades adicionais, uma replicação física e lógica pode ser usada concorrentemente.

[00110] O armazenamento dinâmico 240 (isto é, a coleção de todas as tabelas de gravação) pode ser concretizado de várias formas. Em uma modalidade, o armazenamento dinâmico 240 é centralizado; isto é, todos os dados de tempo de execução são armazenados na memória de uma máquina remota 30 no banco 38. A máquina remota opera como um nó de rede mestre com o qual todas as outras máquinas remotas 30 no banco 38 se comunicam, quando buscando acesso àqueles dados de tempo de execução. Em uma outra modalidade, cada máquina remota 30 na banco 38 mantém uma cópia plena do armazenamento dinâmico 240. Aqui, cada máquina remota 30 se comunica com cada outra máquina remota 30 para manter sua cópia de armazenamento dinâmico 240 atualizada.

[00111] Em uma outra modalidade, cada máquina remota 30 mantém seus próprios dados de tempo de execução e se comunica com cada outra máquina remota 30 quando buscando obter dados de tempo de processamento a partir deles. Assim, por exemplo, uma máquina remota 30 tentando encontrar um programa aplicativo requisitado pela máquina local 10 pode se comunicar diretamente com cada outra máquina remota 30 na banco 38 para encontrar uma ou mais máquinas remotas hospedando o aplicativo requisitado.

[00112] Para bancos 38 tendo um grande número de máquinas remotas 30, o tráfego de rede produzido por estas modalidades pode se tornar pesado. Uma modalidade elimina o tráfego pesado de rede

pela designação de um subconjunto das máquinas remotas 30 em um banco 38, tipicamente duas ou mais, como "pontos de coletor". Geralmente, um ponto de coletor é uma máquina remota que coleta dados de tempo de execução. Cada ponto de coletor armazena dados de tempo de execução coletados a partir de certas outras máquinas remotas 30 no banco 38. Cada máquina remota 30 no banco 38 é capaz de operar como e, conseqüentemente, é capaz de ser projetada como um ponto de coletor. Em uma modalidade, cada ponto de coletor armazena uma cópia do armazenamento dinâmico inteiro 240. Em uma outra modalidade, cada ponto de coletor armazena uma porção do armazenamento dinâmico 240, isto é, mantém dados de tempo de execução de um tipo de dados em particular. O tipo de dados armazenado por uma máquina remota 30 pode ser predeterminado de acordo com um ou mais critérios. Por exemplo, as máquinas remotas 30 podem armazenar tipos diferentes de dados com base em sua ordem de boot. Alternativamente, o tipo de dado armazenado processamento uma máquina remota 30 pode ser configurado por um administrador usando a ferramenta de administração 140. Nestas modalidades, o armazenamento dinâmico 240 é distribuído dentre duas ou mais máquinas remotas 30 no banco 38. Em uma outra modalidade, uma aplicação 1250 pode aliviar o tráfego de rede pesado pela aceleração de dados passados entre as máquinas remotas 30, o armazenamento dinâmico 16240 e o armazenamento persistente 230. Essa aceleração pode ser provida por qualquer uma das técnicas discutidas adicionalmente na Seção C. Por exemplo, a aplicação 1250 pode ser usada para alívio do tráfego de rede pesado.

[00113] As máquinas remotas 30 não designadas como pontos de coletor sabem as máquinas remotas 30 em um banco 38 que são designadas como pontos de coletor. Uma máquina remota 180 não designada como um ponto de coletor pode se comunicar com um

ponto de coletor em particular quando entregando e requisitando dados de tempo de execução. Conseqüentemente, os pontos de coletor escutam o tráfego de rede porque cada máquina remota 30 no banco 38 se comunica com uma única máquina remota de ponto de coletor 30, ao invés de com cada outra máquina remota 30, quando buscando acessar os dados de tempo de execução.

[00114] Cada máquina remota 30 pode operar como um ponto de coletor para mais de um tipo de dados. Por exemplo, a máquina remota 30'' pode operar como um ponto de coletor para uma informação de licenciamento e para uma informação de carregamento. Nestas modalidades, cada ponto de coletor pode acumular um tipo diferente de dados de tempo de execução. Por exemplo, para ilustração deste caso, a máquina remota 30''' pode coletar uma informação de licenciamento, enquanto a máquina remota 30'' pode coletar uma informação de carregamento.

[00115] Em algumas modalidades, cada ponto de coletor armazena dados que são compartilhados entre todas as máquinas remotas 30 em um banco 38. Nestas modalidades, cada ponto de coletor de um tipo em particular de dados troca os dados coletados por aquele ponto de coletor com cada outro ponto de coletor para aquele tipo de dados na banco 38. Assim, mediante a conclusão da troca desses dados, cada máquina remota 30'' e 30 possui os mesmos dados. Também, nestas modalidades, cada ponto de coletor 30 e 30'' também mantém todo outro ponto de coletor bem informado de quaisquer atualizações nos dados de tempo de execução.

[00116] Uma navegação permite que uma máquina local 10 veja os bancos 38, as máquinas remotas 30 e os aplicativos nos bancos 38 e acesse uma informação disponível, tais como sessões através do banco 38. Cada máquina remota 30 inclui um subsistema de navegação de ICA 260 para a provisão à máquina local 10 de uma capacidade de

navegação. Após a máquina local 10 estabelecer uma conexão com o subsistema de navegador de ICA 260 de qualquer uma das máquinas remotas 30, aquele subsistema de navegador suporta uma variedade de requisições de máquina local. Essas requisições de máquina local incluem: (1) enumeração de nomes de máquinas remotas no banco, (2) enumeração de nomes de aplicativos publicados no banco, (3) resolução de um nome e/ou de um nome de aplicativo de máquina remota para um endereço de máquina remota que é útil para a máquina local 10. O subsistema de navegador de ICA 260 também suporta requisições feitas por máquinas locais 10 rodando um aplicativo de vizinhança de programa que provê à máquina local 10, mediante uma requisição, uma visão daqueles aplicativos no banco 38 aos quais o usuário está autorizado. O subsistema de navegador de ICA 260 encaminha todas as requisições de máquina local mencionadas acima para o subsistema apropriado na máquina remota 30.

[00117] Em uma modalidade, cada máquina remota 30 no banco 38 que tem um subsistema de vizinhança de programa 270 pode prover ao usuário de uma máquina local 10 uma visão de aplicativos no banco 38. O subsistema de vizinhança de programa 270 pode limitar a visão àqueles aplicativos para os quais o usuário da máquina local 10 tem autorização para acessar. Tipicamente, este serviço de vizinhança de programa apresenta ao usuário os aplicativos como uma lista ou um grupo de ícones.

[00118] A funcionalidade provida pelo subsistema de vizinhança de programa 270 está disponível para dois tipos de máquinas locais, (1) máquinas locais habilitadas com vizinhança de programa que podem acessar a funcionalidade diretamente a partir de uma área de trabalho de máquina local e (2) máquinas locais não habilitadas com vizinhança de programa (por exemplo, máquinas locais de legado) que podem acessar a funcionalidade ao rodarem uma área de trabalho habilitada

com vizinhança de programa na máquina remota.

[00119] Uma comunicação entre uma máquina local com vizinhança de programa habilitada e o subsistema de vizinhança de programa 270 pode ocorrer por um canal virtual dedicado que é estabelecido no topo de um canal virtual de ICA. Em outras modalidades, a comunicação ocorre usando-se um serviço de XML. Em uma destas modalidades, a máquina local com vizinhança de programa habilitada se comunica com um subsistema de XML, tal como o serviço de XML 516 descrito em relação à figura 6 abaixo, provendo uma funcionalidade de vizinhança de programa em uma máquina remota 30.

[00120] Em uma modalidade, a máquina local com vizinhança de programa habilitada não tem uma conexão com a máquina remota com um subsistema de vizinhança de programa 270. Para esta modalidade, a máquina local 10 envia uma requisição para o subsistema de navegador de ICA 260 para o estabelecimento de uma conexão de ICA com a máquina remota 30, de modo a se identificarem aplicativos disponíveis para a máquina local 10. A máquina local 10 então roda um diálogo de lado de cliente que adquire as credenciais de um usuário. As credenciais são recebidas pelo subsistema de navegador de ICA 260 e enviadas para o subsistema de vizinhança de programa 270. Em uma modalidade, o subsistema de vizinhança de programa 270 envia as credenciais para um subsistema de gerenciamento de usuário para autenticação. O subsistema de gerenciamento de usuário pode retornar um conjunto de nomes distinguidos representando uma lista de contas às quais o usuário pertence. Mediante uma autenticação, o subsistema de vizinhança de programa 270 estabelece o canal virtual de vizinhança de programa. Este canal permanece aberto até a filtração de aplicativo estar completada. Em algumas modalidades, um programa de aceleração 6120, conforme descrito na Seção C, também pode ser transmitido

para a máquina local 10, em resposta a uma requisição de máquina local 10.

[00121] O subsistema de vizinhança de programa 270 então requisita a informação de vizinhança de programa a partir do subsistema de aplicativo comum 524 associado àquelas contas. O subsistema de aplicativo comum 524 obtém a informação de vizinhança de programa a partir do armazenamento persistente 230. Ao receber a informação de vizinhança de programa, o subsistema de vizinhança de programa 270 formata e retorna a informação de vizinhança de programa para a máquina local pelo canal virtual de vizinhança de programa. Então, a conexão de ICA parcial é fechada.

[00122] Para um outro exemplo no qual a máquina local com vizinhança de programa habilitada estabelece uma conexão de ICA parcial com uma máquina remota, considere o uso da máquina local 10 que seleciona um banco 38. A seleção do banco 38 envia uma requisição a partir da máquina local 10 para o subsistema de navegador de ICA 260 para o estabelecimento de uma conexão de ICA com uma das máquinas remotas 30 no banco selecionado 38. O subsistema de navegador de ICA 260 envia a requisição para o subsistema de vizinhança de programa 270, o qual seleciona uma máquina remota 30 no banco 38. Uma informação de endereço associada à máquina remota 30 é identificada e retornada para a máquina local 10 por meio do subsistema de navegador de ICA 260. A máquina local 10 então pode subsequente se conectar à máquina remota 30 correspondente à informação de endereço recebida.

[00123] Em uma outra modalidade, a máquina local com vizinhança de programa habilitada 10 tem uma conexão de ICA pela qual o canal virtual de vizinhança de programa é estabelecido e permanece aberto por quanto tempo a conexão de ICA persistir. Por este canal virtual de

vizinhança de programa, o subsistema de vizinhança de programa 270 empurra as atualizações de informação de vizinhança de programa para a máquina local 10. Para a obtenção de atualizações, o subsistema de vizinhança de programa 270 assina eventos a partir do subsistema de aplicativo comum 524 para permitir que o subsistema de vizinhança de programa 270 detecte mudanças em aplicativos publicados.

[00124] Com referência à figura 1F, um diagrama de blocos descreve uma outra modalidade de uma arquitetura de sistema para a provisão de uma pluralidade de programas aplicativos disponíveis para a máquina local através da publicação de GUIs em um diretório de serviço da web. O sistema inclui a máquina local 10 e uma pluralidade de máquinas remotas 30. Uma máquina remota 30 funciona como um servidor de conteúdo. Uma máquina remota 30' provê uma funcionalidade de servidor da web. Uma máquina remota 30'' provê uma funcionalidade para a provisão de acesso a arquivos aplicativos e atua como um servidor de aplicativo ou um servidor de arquivo. A máquina local 10 pode transferir (via download) conteúdo do servidor de conteúdo 30, do servidor da web 30' e do servidor de aplicativo 30'' pela rede 155. Em uma modalidade, a máquina local 10 pode transferir conteúdo (por exemplo, um aplicativo) a partir do servidor de aplicativo 30'' pelo canal de comunicação de cliente – servidor de aplicativo 1150.

[00125] Em uma modalidade, o navegador da web 11 na máquina local 10 usa um suporte de Camada de Soquete Seguro (SSL) para comunicações com o servidor de conteúdo 30 e/ou o servidor da web 30'. O SSL é um protocolo seguro desenvolvido pela Netscape Communication Corporation de Mountain View, Califórnia, e agora é um padrão promulgado pela Internet Engineering Task Force (IETF). O navegador da web 11 alternativamente pode se conectar ao servidor de

conteúdo 30 e/ou ao servidor da web 30' usando outros protocolos de segurança, tais como, mas não limitando, um Protocolo de Transferência de Hipertexto Seguro (SHTTP) desenvolvido pela Terisa Systems de Los Altos, CA, HTTP over SSL (HTTPS), Tecnologia de Comunicação Privada (PCT) desenvolvido pela Microsoft Corporation de Redmond, Washington, e o padrão de Segurança de Nível de Transporte (TLS) promulgado pela IETF. Em outras modalidades, o navegador da web 11 se comunica com os servidores 30 usando um protocolo de comunicações sem encriptação, tal como o Protocolo de Transferência de Hipertexto (HTTP).

[00126] Adicionalmente, a máquina local 10 inclui um cliente de aplicativo 13 para o estabelecimento e a troca de comunicações com o servidor de aplicativo 30'' pelo canal de comunicação de cliente – servidor de aplicativo 1150. Em uma modalidade, o cliente de aplicativo 13 é um aplicativo de GUI. Em algumas modalidades, o cliente de aplicativo 13 é um cliente de Arquitetura de Computação Independente (ICA), desenvolvido pela Citrix Systems, Inc. de Fort Lauderdale, Flórida, e também é referido abaixo como o cliente de ICA 13. Outras modalidades do cliente de aplicativo 13 incluem um cliente de Protocolo de Exibição Remota (RDP), desenvolvido pela Microsoft Corporation de Redmond, Washington, um cliente de X-Windows 13, um tocador de lado de cliente, um intérprete ou um simulador capaz de executar aplicativos de multimídia, e-mail, Java, ou código .NET. Mais ainda, em uma modalidade, a saída de um aplicativo em execução no servidor de aplicativo 30'' pode ser exibida na máquina local 10 através do cliente de aplicativo 13. Em algumas modalidades, o cliente de aplicativo 13 é um cliente de aplicativo tal como o cliente de transmissão de aplicativo 552, descrito em maiores detalhes em relação à figura 5. Em algumas modalidades, o cliente de aplicativo 13 compreende um programa de aceleração de acordo com qualquer

uma das modalidades descritas aqui 6120, para aceleração das comunicações entre o cliente 10 e o servidor 30.

[00127] A máquina local 10 busca o diretório de serviço da web 160 quanto a um serviço da web. Em uma modalidade, a busca é uma busca manual. Alternativamente, a busca é uma busca automática. O diretório de serviço da web 160 também pode prover uma vista baseada em serviço, tal como um catálogo de endereços da Internet ou páginas amarelas, para busca por serviços da web no diretório de serviço da web. Em uma outra modalidade, o diretório de serviço da web 160 suporta uma navegação hierárquica baseada em um nome de serviço estruturado e tipo de serviço para aplicativos de GUI. Em uma modalidade, o diretório de serviço da web 160 é executado em uma máquina remota independente do servidor de conteúdo 30, tal como um servidor de diretório. Em outras modalidades, o diretório de serviço da web 160 é executado em múltiplos servidores.

[00128] Em algumas modalidades, o servidor de conteúdo 30 permite que a máquina local 10 selecione serviços da web com base em uma análise adicional ou informação pela provisão desta informação ou análise no diretório de serviço da web 160. Os exemplos de informação de serviço que o diretório de serviço da web 160 pode listar incluem, mas não estão limitados ao nome do negócio oferecendo o serviço, o tipo de serviço, uma descrição textual do serviço, um ou mais pontos de acesso de serviço (SAPs), o tipo de rede, o caminho a usar (por exemplo, TCP ou HTTPS), e a informação de qualidade de serviço (QoS). Mais ainda, a informação de serviço pode ser específica de tipo ou usuário (por exemplo, papel) de dispositivo cliente. Assim, a seleção de serviço pode ser baseada em um ou mais dos atributos acima.

[00129] Em uma modalidade, o tipo de serviço denota uma interface de programação que a máquina local 10 deve usar para acessar o serviço da web. Por exemplo, o tipo de serviço pode declarar que o

serviço é codificado por uma linguagem de descrição de interface, tal como a Linguagem de Descrição de Serviços da Web (WSDL).

[00130] O ponto de acesso de serviço, ou SAP, é um endereço único para um aplicativo. Os SAPs permitem que o sistema de computador suporte múltiplos aplicativos na máquina local 10 e em cada máquina remota 30. Por exemplo, o servidor de aplicativo 30'' pode suportar um aplicativo de correio eletrônico (isto é, e-mail), um aplicativo de transferência de arquivo e/ou um aplicativo de GUI. Em uma modalidade, cada um destes aplicativos teria um SAP que seria único no servidor de aplicativo 30''. Em uma modalidade, o SAP é um nome de endereço da web ou Internet (por exemplo, um Sistema de Nome de Domínio (DNS), um IP/porta, ou um Localizador de Recurso Uniforme (URL)). Assim, em uma modalidade, o SAP identifica o endereço do servidor da web 30' como parte do endereço para um aplicativo armazenado no servidor da web 30'. Em algumas modalidades, o SAP identifica o endereço de um plug-in de servidor de publicação 165 como parte do endereço para um aplicativo armazenado no servidor da web 30', conforme descrito abaixo. Em uma modalidade, o SAP é um "Ponto de acesso" ("accessPoint") do registro de UDDI.

[00131] Para preparação de um item para publicação no diretório de serviço da web 160, o servidor de conteúdo 30 inclui uma ferramenta de publicação da web 170. Em uma modalidade, a ferramenta de publicação da web 170 é um módulo de software. Alternativamente, a ferramenta de publicação da web 170 é um outro servidor que pode estar externamente localizado ou internamente localizado no servidor de conteúdo 30.

[00132] Em uma modalidade, o servidor da web 30' entrega páginas da web para a máquina local 10. O servidor da web 30' pode ser qualquer máquina remota 30 capaz de prover páginas da web para a

máquina local 105. Em uma outra modalidade, o servidor da web 30' é um Portal de Informação de Empresa (por exemplo, uma Intranet corporativa ou uma extranet de negócio para negócio seguro). Os portais de empresa são websites de companhia que agregam, personalizam e servem aplicativos, dados e conteúdo para usuários, enquanto oferecem ferramentas para organização e uso de uma informação de forma mais eficiente. Em algumas companhias, os portais substituíram o software de área de trabalho tradicional por um acesso baseado em navegador para um local de trabalho virtual. Em algumas modalidades, uma aplicação 1250 acelera a entrega da provisão de páginas da web que é acelerada usando-se qualquer uma das técnicas de aceleração usadas aqui. Em outras modalidades, o programa de aceleração 6120 acelera a entrega das páginas da web.

[00133] O servidor da web 30' também inclui um plug-in de servidor de publicação 165 para permitir a publicação de aplicativos de interface gráfica de usuário (GUI). Mais especificamente, o plug-in de servidor de publicação 165 traduz um URL de entrada de serviço da web novo em um serviço de aplicativo de GUI, de modo que a GUI possa ser acessada através do diretório de serviço da web 160. Em uma modalidade, o plug-in de servidor de publicação 165 é um script de Interface de Gateway Comum (CGI), o qual é um programa projetado para aceitar e retornar dados que se conformem à especificação de CGI. O programa pode ser escrito em qualquer linguagem de programação, tal como C, Perl, Java ou Visual Basic. Em uma outra modalidade, o plug-in de servidor de publicação 165 é uma Página de Servidor Java (JSP). Usando o plug-in de servidor de publicação 165 para facilitar a publicação de aplicativos de GUI remotos, a máquina local 10 desse modo pode acessar o serviço da web, não através de uma interface de programação ou página da web, mas através de uma interface de GUI plena, tal como a ICA da Citrix

ou o RDP da Microsoft. Em algumas modalidades, uma aplicação 1250 ou um programa de aceleração 6120 acelera a entrega da referida GUI para o cliente, que é acelerada usando-se qualquer uma das técnicas de aceleração discutidas aqui na Seção C.

[00134] O servidor de aplicativo 30'' hospeda um ou mais aplicativos que estão disponíveis para a máquina local 10. Os exemplos desses aplicativos incluem programas de processamento de texto, tal como o MICROSOFT WORD, e programas de planilha, tal como o MICROSOFT EXCEL, ambos Microsoft Corporation de Redmond, Washington, programas de relatório financeiro, programas de registro de consumidor, programas provendo uma informação de suporte técnico, aplicativos de banco de dados de consumidor ou gerenciadores de conjunto de aplicativo.

[00135] Em uma modalidade, a ferramenta de publicação da web 170 armazena uma informação sobre um aplicativo que a ferramenta de publicação da web 170 está publicando no diretório de serviço da web 160 em um armazenamento de massa persistente 225. Em uma modalidade, a informação é um URL para o plug-in de servidor de publicação dinâmico 165. O armazenamento de massa persistente 225 pode ser um disco magnético ou um disco magneto-ótico. Em uma modalidade, o armazenamento de massa persistente 225 é um servidor de banco de dados, o qual armazena dados relacionados ao aplicativo publicado em um ou mais bancos de dados de serviço local. O armazenamento de massa persistente 225 pode ser um componente localizado internamente ou localizado externamente em relação a quaisquer ou todas as máquinas remotas 30.

[00136] Em outras modalidades, o servidor de conteúdo 30 ou o servidor da web 30' se comunica com uma máquina remota 30 no banco 38 para a recuperação da lista de aplicativos. Em uma destas modalidades, o servidor de conteúdo 30 ou o servidor da web 30' se

comunica com o banco 38, ao invés de com o armazenamento de massa persistente 225.

[00137] Com referência, agora, à figura 2, um fluxograma descreve uma modalidade das etapas seguidas para a seleção de um método de execução de um programa aplicativo. Em uma visão geral breve, as credenciais associadas à máquina local ou a um usuário da máquina local são recebidas, com uma requisição para uma enumeração de aplicativos disponíveis para execução pela máquina local (etapa 202). Uma enumeração de uma pluralidade de programas aplicativos disponível para a máquina local é provida, em resposta às credenciais recebidas (etapa 204). Uma requisição é recebida para execução do aplicativo enumerado (etapa 206). Um de um número predeterminado de métodos para execução do aplicativo enumerado é selecionado, em resposta a uma política, o número predeterminado de métodos incluindo um método para transmissão de aplicativo do aplicativo enumerado (etapa 208).

[00138] As credenciais associadas à máquina local ou a um usuário da máquina local são recebidas, com a requisição por uma enumeração de aplicativos disponíveis para execução pela máquina local (etapa 202). Em uma modalidade, a máquina remota recebe uma requisição por enumeração de aplicativos disponíveis a partir da máquina local 10 com credenciais. Em uma outra modalidade, um serviço de XML na máquina remota 30 recebe a requisição e as credenciais e transmite a requisição e as credenciais para um serviço de gerenciamento na máquina remota 30.

[00139] Em algumas modalidades, uma máquina remota 30 funcionando como um servidor da web recebe comunicações a partir da máquina local 10 e encaminha as comunicações para uma máquina remota 30'. Em uma destas modalidades, o servidor da web encaminha as comunicações para um serviço de XML na máquina

remota 30'. Em uma outra destas modalidades, o servidor da web reside na máquina local. Em outras modalidades em que as comunicações a partir da máquina local 10 são roteadas para uma máquina remota 30' pelo servidor da web, a máquina remota 30 pode ser selecionada em resposta a um endereço de Protocolo da Internet (IP) da máquina local 10.

[00140] Em algumas modalidades, uma máquina local 10 requisita acesso a um aplicativo residente em uma máquina remota 30. Em uma destas modalidades, a máquina local 10 requisita uma execução pela máquina remota 30 do aplicativo residente na máquina remota 30. Em uma outra destas modalidades, a máquina local 10 requisita uma recuperação de uma pluralidade de arquivos aplicativos que compreendem o aplicativo.

[00141] Em algumas modalidades, o usuário provê credenciais para a máquina remota 30 através de uma interface gráfica de usuário apresentada para a máquina local 10 pela máquina remota 30. Em outras modalidades, uma máquina remota 30''' tendo a funcionalidade de um servidor da web provê a interface gráfica de usuário para a máquina local 10. Ainda em outras modalidades, um agente de coleta transmitido para a máquina local 10 pela máquina remota 30 acumula as credenciais a partir da máquina local 10. Em uma modalidade, uma credencial se refere a um nome de usuário e uma senha. Em uma outra modalidade, uma credencial não está limitada a um nome de usuário e uma senha, mas inclui, sem limitação, uma ID de máquina da máquina local 10, o tipo de sistema operacional, a existência de um "patch" para um sistema operacional, endereços de MAC de placas de rede instaladas, uma marca d'água digital no dispositivo cliente, uma participação em um Diretório Ativo, a existência de uma varredura de vírus, a existência de um firewall pessoal, um cabeçalho de http, um tipo de navegador, um tipo de dispositivo, uma informação de conexão

de rede, tal como um endereço de protocolo de internet ou uma faixa de endereços, uma ID de máquina da máquina remota 30, data ou horário de requisição de acesso incluindo ajustes quanto a fusos-horários variáveis, e credenciais de autorização.

[00142] Em algumas modalidades, uma credencial associada a uma máquina local está associada a um usuário da máquina local. Em uma destas modalidades, a credencial é uma informação possuída pelo usuário. Em uma outra destas modalidades, a credencial é uma informação de autenticação de usuário. Em outras modalidades, uma credencial associada a uma máquina local está associada a uma rede. Em uma destas modalidades, a credencial é uma informação associada a uma rede à qual a máquina local pode se conectar. Em uma outra destas modalidades, a credencial é uma informação associada a uma rede coletando uma informação sobre a máquina local. Ainda em outras modalidades, uma credencial associada a uma máquina local é uma característica da máquina local.

[00143] Uma enumeração de uma pluralidade de programas aplicativos disponíveis para a máquina local é provida, em resposta às credenciais recebidas (etapa 204). Em uma modalidade, um usuário de uma máquina local 10 pode aprender a disponibilidade de programas aplicativos hospedados pelas máquinas remotas 30 na rede 40, sem saber onde encontrar tais aplicativos e sem uma informação técnica necessária para se ligar a esses aplicativos. Estes programas aplicativos disponíveis compreendem a "vizinhança de programa" do usuário. Um sistema para determinação de uma vizinhança de programa para uma máquina local inclui um programa aplicativo (a partir deste ponto referido como o aplicativo de "Vizinhança de Programa"), uma memória para o armazenamento de componentes do programa aplicativo, e um processador para execução do programa aplicativo. O aplicativo de Vizinhança de Programa (PN) pode ser

instalado na memória da máquina local 10 e/ou em uma máquina remota 30, conforme descrito abaixo.

[00144] Uma máquina remota 30 operando de acordo com o aplicativo de Vizinhança de Programa coleta uma informação relacionada a aplicativo a partir de cada uma das máquinas remotas 30 em um banco 38. A informação relacionada a aplicativo para cada aplicativo hospedado pode ser uma variedade de informação incluindo, por exemplo, um endereço da máquina remota hospedando aquele aplicativo, o nome de aplicativo, os usuários ou grupos de usuários que estão autorizados a usarem aquele aplicativo, e as capacidades mínimas requeridas da máquina local 10, antes do estabelecimento de uma conexão para rodar o aplicativo. Por exemplo, o aplicativo pode transmitir dados de vídeo e, portanto, uma capacidade mínima requerida pode ser que a máquina local suporte dados de vídeo. Outros exemplos são exigências para que a máquina local suporte dados de áudio ou tenha a capacidade de processar dados encriptados. A informação relacionada a aplicativo pode ser armazenada em um banco de dados.

[00145] Quando uma máquina local 10 se conecta à rede 40, o usuário da máquina local 10 provê credenciais de usuário. As credenciais de usuário podem incluir o nome de usuário de um usuário da máquina local 10, a senha do usuário, e o nome de domínio para o qual um usuário está autorizado. Alternativamente, as credenciais de usuário podem ser obtidas a partir de smart cards, fichas baseadas em tempo, números de inscrição social, senhas de usuário, números de identificação pessoal (PIN), certificados digitais com base em chave simétrica ou criptografia de curva elíptica, características biométricas do usuário ou qualquer outro meio pelo qual a identificação do usuário da máquina local 10 possa ser obtida e submetida para autenticação. A máquina remota 30 respondendo à máquina local 10 pode autenticar

o usuário com base nas credenciais de usuário. As credenciais de usuário podem ser armazenadas em qualquer lugar em que o aplicativo de Vizinhança de Programa esteja em execução. Para modalidades nas quais uma máquina remota 30 executa a Vizinhança de Programa, as credenciais podem ser armazenadas naquela máquina remota 30.

[00146] A partir das credenciais de usuário e da informação relacionada a aplicativo, a máquina remota 30 também pode determinar quais programas aplicativos hospedados pelas máquinas remotas 30 estão disponíveis para uso pelo usuário da máquina local 10. A máquina remota 30 transmite uma informação representando os programas aplicativos disponíveis para a máquina local 10. Este processo elimina a necessidade de um usuário da máquina local 10 estabelecer conexões de aplicativo. Adicionalmente, um administrador da máquina remota 30 pode controlar o acesso a aplicativos dentre múltiplos usuários de uma máquina local 10.

[00147] Em algumas modalidades, a autenticação de usuário realizada pela máquina remota 30 pode ser suficiente para autorizar o uso de cada programa aplicativo hospedado apresentado para a máquina local 10, embora esses aplicativos possam residir em uma outra máquina remota 30'. Assim sendo, quando a máquina local 10 abre (isto é, inicia a execução de) um dos aplicativos hospedados, uma entrada adicional de credenciais de usuário pela máquina local 10 pode ser desnecessária para a autenticação do uso daquele aplicativo. Assim, uma entrada única de credenciais de usuário pode servir para a determinação dos aplicativos disponíveis e para autorização da abertura desses aplicativos, sem um processo de autenticação de logon manual adicional pelo usuário.

[00148] Uma máquina local 10 ou uma máquina remota 30 pode abrir o aplicativo de Vizinhança de Programa. Os resultados são

exibidos na tela de exibição 12, 22 da máquina local 10. Em uma implementação baseada em janela gráfica, os resultados podem ser exibidos em uma janela gráfica de Vizinhança de Programa e cada programa aplicativo autorizado pode ser apresentado por um ícone gráfico naquela janela.

[00149] Em uma modalidade, o aplicativo de Vizinhança de Programa filtra os programas aplicativos que a máquina local 10 não está autorizada a executar e exibe apenas os programas autorizados (isto é, disponíveis). Em outras modalidades, o aplicativo de Vizinhança de Programa pode exibir aplicativos autorizados e não autorizados. Quando aplicativos não autorizados são filtrados a partir da exibição, uma notificação pode ser provida indicando que esses aplicativos não estão disponíveis. Alternativamente, o aplicativo de Vizinhança de Programa pode reportar todos os aplicativos hospedados pelas máquinas remotas 30 para o usuário de uma máquina local 10, sem a identificação de quais aplicativos a máquina local 10 está autorizada ou não está autorizada a executar. Uma autorização pode ser subsequente determinada quando a máquina local 10 tentar rodar um destes aplicativos.

[00150] A máquina local 10 pode requisitar uma enumeração de aplicativo a partir de uma máquina remota 30. A enumeração de aplicativo permite que um usuário da máquina local 10 veja os nomes de todo aplicativo publicado. Em uma modalidade, o usuário da máquina local 10 pode ver os nomes de aplicativo independentemente de se o usuário tem autorização para executar o aplicativo. Em uma outra modalidade, o usuário vê apenas aqueles nomes de aplicativo que o usuário está autorizado a executar.

[00151] As requisições por enumeração de aplicativo passam para o subsistema de navegador de ICA 260, para o subsistema de vizinhança de programa 270 ou para um subsistema de aplicativo

comum 524, dependendo do processo em particular sendo rodado pela máquina local 10. Por exemplo, quando a máquina local 10 está rodando um aplicativo de vizinhança de programa, as requisições para enumeração de aplicativo são enviadas para o subsistema de vizinhança de programa 270 em uma máquina remota 30. Quando a máquina local 10 submete uma requisição de enumeração através de uma página da web, as requisições passam para o subsistema de ponto de acesso comum 524. Para estas modalidades, o subsistema de aplicativo comum 524 serve como um ponto de acesso inicial para o subsistema de vizinhança de programa 270, o subsistema de navegador de ICA 260 e os subsistemas de aplicativo comum, quando a máquina local 10 quiser enumerar os aplicativos. Em algumas modalidades, quando a máquina local 10 submete a requisição de enumeração através de uma página da web, uma máquina remota intermediária 30 hospedando um servidor da web recebe a requisição e encaminha a requisição para uma máquina remota 30'.

[00152] Mediante o recebimento das requisições de enumeração, um subsistema de aplicativo comum 524 consulta o armazenamento persistente 230 quanto a uma lista de todos os aplicativos. Para requisições recebidas a partir do subsistema de vizinhança de programa 270 e de subsistemas de ponto de acesso comum 645, esta lista de aplicativos é filtrada de acordo com as credenciais do usuário da máquina local 10 (isto é, o usuário vê apenas aqueles aplicativos aos quais o usuário está autorizado).

[00153] A máquina local 10 também pode requisitar uma enumeração de máquina remota. A enumeração de máquina remota permite que um usuário da máquina local 10 veja uma lista de máquinas remotas no banco 38. Em uma modalidade, a lista de máquinas remotas pode ser filtrada de acordo com o tipo de máquina remota, conforme determinado pelo subsistema de máquina remota

especializado naquela máquina remota.

[00154] As requisições para enumeração de máquina remota passam para o subsistema de navegador de ICA 260 ou para o subsistema de ponto de acesso comum 645, dependendo do processo em particular sendo rodado pela máquina local 120. Por exemplo, quando a máquina local 120 submete a requisição de enumeração de máquina remota através de uma página da web, as requisições passam para o subsistema de ponto de acesso comum 645. Para estas modalidades, o subsistema de máquina remota comum 300 serve como um ponto de acesso inicial para o subsistema de navegador de ICA 260 e os subsistemas de ponto de acesso comum 645. Mediante o recebimento das requisições de enumeração de máquina remota, o subsistema de máquina remota comum consulta o armazenamento persistente 230 quanto a uma lista de todas as máquinas remotas. Opcionalmente, a lista de máquinas remotas é filtrada de acordo com o tipo de máquina remota.

[00155] A Figura 3A é um diagrama de blocos que descreve uma outra modalidade do processo pelo qual uma máquina local 10 inicia uma execução do aplicativo de Vizinhaça de Programa, neste exemplo, através da Rede Mundial. Uma máquina local 10 executa um aplicativo de navegador da web 80, tal como o NETSCAPE NAVIGATOR, fabricado pela Netscape Communications, Inc. de Mountain View, Califórnia ou o MICROSOFT INTERNET EXPLORER, fabricado pela Microsoft Corporation de Redmond, Washington, ou FIREFOX, fabricado pela Mozilla Foundation de Mountain View, Califórnia, ou OPERA, fabricado pela Opera Software ASA, de Oslo, Noruega, ou SAFARI, fabricado pela Apple Computer, Inc., de Cupertino, Califórnia.

[00156] A máquina local 10, através do navegador da web 80, transmite uma requisição 82 para acesso a um endereço de

Localizador de Recurso Uniforme (URL) correspondente a uma página em HTML residente na máquina remota 30. Em algumas modalidades, a primeira página em HTML retornada 84 para a máquina local 10 pela máquina remota 30 é uma página de enumeração que busca identificar a máquina local 10.

[00157] Ainda com referência à figura 3A, uma vez que a máquina local 10 seja autenticada pela máquina remota 30, a máquina remota 30 prepara e transmite para a máquina local 10 uma página em HTML 88 que inclui uma janela de Vizinhança de Programa 58 na qual aparecem os ícones gráficos 57, 57' representando programas aplicativos aos quais a máquina local 10 tem acesso. Um usuário de máquina local 10 invoca uma execução de um aplicativo representado pelo ícone 57 ao clicar naquele ícone 57.

[00158] Em algumas modalidades, a máquina remota 30 executa o aplicativo de Vizinhança de Programa em nome de um usuário da máquina local 10. Em uma destas modalidades, a máquina remota 30 é uma máquina remota intermediária residente entre a máquina local 10 e uma máquina remota 30'.

[00159] Com referência à figura 3B, um fluxograma descreve uma modalidade das etapas seguidas para a provisão de uma pluralidade de programas aplicativos disponíveis para a máquina local através da publicação de GUIs em um diretório de serviço da web. A ferramenta de publicação da web 170 recebe uma descrição de serviço da web e uma informação de acesso para um aplicativo (por exemplo, um aplicativo de GUI) para publicação (etapa 300). Em uma modalidade, a descrição de serviço da web inclui uma informação de serviço descrita acima (por exemplo, o nome do escritório oferecendo o serviço da web, o tipo de serviço, a descrição textual do serviço e um SAP). A informação de acesso pode incluir, por exemplo, um nome de aplicativo publicado, um endereço de banco de servidor de navegação

de Protocolo de Controle de Transmissão (TCP), e um endereço de IP de servidor de Metaquadro. Em algumas modalidades, a informação de acesso especifica o endereço a usar e um tíquete a usar para atravessar a rede ou "gateways" de segurança ou dispositivos de ponte.

[00160] A ferramenta de publicação da web 170 então constrói uma requisição de publicação de serviço para requisitar a publicação do serviço da web (isto é, um aplicativo de GUI) (etapa 305). Em uma modalidade, a requisição de publicação de serviço inclui um SAP. Em algumas modalidades, o SAP é um URL incluindo um endereço da web do servidor da web 30' e o plug-in de servidor de publicação 165. Ainda, o endereço da web pode ser um Identificador de Recurso Uniforme (URI), o qual é o termo genérico para os tipos de nomes e endereços que se referem aos objetos na web. Um URL é um tipo de URI. Um exemplo do URI é o nome do servidor da web 30' (por exemplo, "servidor da web") e o nome de script de CGI (por exemplo, "componente dinâmico") para o plug-in de servidor de publicação 165.

[00161] A ferramenta de publicação da web 170 armazena uma entrada de SAP associada ao SAP no armazenamento de massa persistente 225 (etapa 310). Em algumas modalidades, a ferramenta de publicação da web 170 também associa a informação de aplicativo publicada (por exemplo, ICA-published-app-info) para o aplicativo de GUI. Em modalidades adicionais, a ferramenta de publicação da web 170 também inclui uma chave na requisição de publicação de serviço para a identificação da entrada de SAP que o servidor de conteúdo 30 armazena no armazenamento de massa persistente 225. Por exemplo, a chave pode ser o valor de "123456677". Um exemplo de um SAP identificando o servidor da web 30', o nome de script de CGI do plug-in de servidor de publicação 165 e a chave descrita acima é <http://web-server/dynamic-component/?app=123456677>.

[00162] Um exemplo da entrada de SAP associada ao SAP descrito acima é "key= 123456677, value=ICA-published-app-info". A chave (key) pode ser de qualquer comprimento (por exemplo, uma chave de 56 bits, uma chave de 128 bits). Em uma modalidade, a chave é um número randômico criptografado. A chave também pode prover acesso ao mantenedor de chave. Embora ilustrado com uma chave, qualquer meio pode ser usado para a provisão de uma forma de segurança à entrada de SAP armazenada no armazenamento de massa persistente 225.

[00163] A ferramenta de publicação da web 170 provê a requisição de publicação de serviço para o servidor de conteúdo 30 para publicação no diretório de serviço da web 160 (etapa 315). Mais ainda, em uma modalidade, o servidor de conteúdo 30 transmite a chave do SAP para a máquina local 10 requisitando o serviço da web em particular para uso subsequente na localização da entrada de SAP. Em uma modalidade, a publicação da requisição de publicação de serviço permite que usuários da máquina local 10 acessem o serviço. Em uma modalidade, as aplicações de GUI são publicadas no diretório de serviço da web 160 usando-se NFUSE desenvolvido pela Citrix Systems, Inc. de Fort Lauderdale, Flórida. Em algumas modalidades, um editor de um aplicativo de GUI personaliza a publicação do aplicativo de GUI no diretório de serviço da web 160 usando-se o Application Launching And Embedding (ALE), também desenvolvido pela Citrix Systems, Inc. O ALE permite a abertura de um aplicativo de GUI a partir de ou a inserção do aplicativo em uma página em HTML.

[00164] A máquina local 10 então consulta um nome de serviço a partir do diretório de serviço da web 160 (etapa 320). O servidor de conteúdo 30 recebe a consulta a partir da máquina local 10 (etapa 325) e encontra o nome de serviço requisitado no diretório de serviço da web 160. Em uma outra modalidade, o usuário da máquina local 10

navega pelo diretório de serviço da web 160 até localizar um nome de serviço em particular que o usuário da máquina local 10 estava tentando encontrar. Embora ilustrado com a máquina local 10, qualquer cliente de diretório de serviço da web (por exemplo, um cliente UDDI ou um navegador LDAP) pode consultar ou navegar pelo diretório de serviço da web 160 para descobrir serviços da web publicados.

[00165] Mediante a localização do SAP associado à consulta recebida, o servidor de conteúdo 30 transmite o SAP para a máquina local 10 (etapa 330). A máquina local 10 recebe o SAP (etapa 335) e determina o endereço do plug-in de servidor de publicação 165 a partir do SAP. A máquina local 10 subsequente transmite uma requisição para o aplicativo de GUI para o servidor da web 30' (etapa 340). Em algumas modalidades, a requisição a partir da máquina local 10 é uma requisição em HTTP transmitida a partir do navegador da web 11 para o servidor da web 30'. Em outras modalidades, um aplicativo (por exemplo, navegador de diretório geral ou HTML UI) em execução na máquina local 10 recebe o SAP a partir do servidor de conteúdo 30 e provê o SAP como um argumento para o navegador da web 11. O navegador da web 11 então pode transmitir automaticamente uma requisição em HTTP (para o aplicativo de GUI) para o servidor da web 30'. Seguindo ao longo das linhas dos exemplos prévios, um exemplo em particular de requisição de aplicativo para o servidor da web 30' é <http://web-server/dynamic-component/?app=123456677>.

[00166] O servidor da web 30' e, mais particularmente, o plug-in de servidor de publicação 165 recebem a requisição de aplicativo associada ao SAP (etapa 345) e determinam a entrada de SAP associada à requisição (etapa 350). Em uma modalidade, o plug-in de servidor de publicação 165 recebe a requisição a partir da máquina

local 10 e recupera a informação de aplicativo publicado associada à requisição que fora armazenada (como parte da entrada de SAP) no armazenamento de massa persistente 225. Em algumas modalidades, o plug-in de servidor de publicação 165 usa o SAP (ou parte do SAP) que a máquina local 10 recebeu a partir do servidor de conteúdo 30 como a chave para acessar a entrada de serviço apropriada (por exemplo, a informação de aplicativo publicado) armazenada no armazenamento de massa persistente 225.

[00167] O plug-in de servidor de publicação 165 então constrói um arquivo ou documento tendo a informação de aplicativo publicada (por exemplo, um endereço em HTTP do servidor de aplicativo 30”) (etapa 352) e transmite este documento para a máquina local 10 (etapa 355). O plug-in de servidor de publicação 165 constrói o arquivo de modo que o arquivo tenha um formato compatível com o cliente de aplicativo 13. Em uma modalidade, o documento é um documento de Extensões de Correio de Internet de Finalidade Múltipla (MIME) ou de MIME seguro (S/MIME). Em uma outra modalidade, o documento é um documento em HTML contendo um tag de HTML de objeto inserido de cliente da web de ICA. Ainda em uma outra modalidade, o documento é um documento em HTML contendo um tag de HTML de objeto inserido de cliente de transmissão de aplicativo.

[00168] O navegador da web 11 subseqüentemente recebe o documento e tenta abrir o documento. Em uma modalidade, se o cliente de aplicativo 13 não estiver instalado na máquina local 10, a máquina local 10 se comunicará com o servidor de aplicativo 30” para transferir (via download) e instalar o cliente de aplicativo 13. Mediante a instalação do cliente de aplicativo 13 ou, alternativamente, se o cliente de aplicativo 13 já tiver sido instalado na máquina local 10, a máquina local 10 abrirá o cliente de aplicativo 13 para ver o documento recebido a partir do servidor da web 30’ (etapa 360).

[00169] Uma vez que o cliente de aplicativo 13 esteja instalado e em execução na máquina local 10, o servidor de aplicativo 30'' então executa o aplicativo e exibe o aplicativo no cliente de aplicativo 13 (etapa 365). Em uma modalidade alternativa, o servidor de aplicativo 30'' transmite uma pluralidade de arquivos aplicativos compreendendo o aplicativo para o cliente de aplicativo 13 para execução na máquina local 10, conforme descrito adicionalmente em detalhes abaixo em relação à figura 7. Em uma outra modalidade, a máquina local 10 vê o documento (mesmo antes da abertura do cliente de aplicativo 13) e usa a informação no documento para obter o aplicativo de GUI a partir do servidor de aplicativo 30''. Nesta modalidade, a exibição do aplicativo de GUI inclui a instalação e a execução do servidor de aplicativo 30''. Mais ainda, a visualização do documento pode ser transparente para o usuário da máquina local 10. Por exemplo, a máquina local 10 pode receber o documento a partir do servidor de aplicativo 30'' e interpretar o documento antes de requisitar automaticamente o aplicativo de GUI a partir do servidor de aplicativo 30''.

[00170] Assim, o cliente de aplicativo 13 provê acesso baseado em serviço para aplicativos publicados, áreas de trabalho, documentos de área de trabalho e qualquer outro aplicativo que seja suportado pelo cliente de aplicativo 13. Os exemplos de aplicativos a que o cliente de aplicativo 13 pode prover acesso incluem, mas não estão limitados às áreas de trabalho de WINDOWS, documentos de WINDOWS, tais como MICROSOFT EXCEL, WORD e POWERPOINT, todos os quais tendo sido desenvolvidos pela Microsoft Corporation de Redmond, Washington, áreas de trabalho de Unix tais como SUN SOLARIS desenvolvido pela Sun Microsystems de Palo Alto, Califórnia, e GNU/Linux distribuído pela Red Hat, Inc. de Durham, North Carolina, dentre outros.

[00171] Em algumas modalidades, a enumeração de uma pluralidade de programas aplicativos disponíveis para a máquina local 10 é provida (etapa 204) em resposta a uma determinação por um agente de política com referência a se e como uma máquina local pode acessar um aplicativo. O agente de política pode coletar uma informação sobre a máquina local, antes de fazer da determinação. Com referência, agora, à figura 4A, uma modalidade de uma rede de computadores construída de acordo com a invenção é descrita, a qual inclui uma máquina local 10, um agente de coleta 404, um agente de política 406, um banco de dados de política 408, um banco 38 e um servidor de aplicativo 30". Em uma modalidade, o agente de política 406 é uma máquina remota 30. Em uma outra modalidade, o servidor de aplicativo 30" é uma máquina remota 30'. Embora apenas uma máquina local 10, um agente de coleta 404, um agente de política 406, um banco 38 e um servidor de aplicativo 30' são descritos na modalidade mostrada na Figura 4A, deve ser entendido que o sistema pode prover múltiplos de qualquer um ou cada um daqueles componentes.

[00172] Em uma visão geral breve, quando a máquina local 10 transmite uma requisição 410 para o agente de política 406 para acesso a um aplicativo, o agente de coleta 404 se comunica com a máquina local 10, recuperando uma informação sobre a máquina local 10, e transmite a informação de máquina local 412 para o agente de política 406. O agente de política 406 toma uma decisão de controle de acesso pela aplicação de uma política a partir do banco de dados de política 408 para a informação recebida 412.

[00173] Em maiores detalhes, a máquina local 10 transmite uma requisição 410 para um recurso para o agente de política 406. Em uma modalidade, o agente de política 406 reside em um servidor de aplicativo 30'. Em uma outra modalidade, o agente de política 406 é

uma máquina remota 30. Ainda em uma outra modalidade, um servidor de aplicativo 30' recebe a requisição 410 a partir da máquina local 10 e transmite a requisição 410 para o agente de política 406. Ainda em uma outra modalidade, a máquina local transmite uma requisição 410 para um recurso para uma máquina remota 30'', a qual transmite a requisição 410 para o agente de política 406.

[00174] Mediante o recebimento da requisição, o agente de política 406 inicia uma acumulação de informação pelo agente de coleta 404. O agente de coleta 404 acumula uma informação referente à máquina local 10 e transmite a informação 412 para o agente de política 406.

[00175] Em algumas modalidades, o agente de coleta 404 acumula e transmite a informação 412 por uma conexão de rede. Em algumas modalidades, o agente de coleta 404 compreende um código de byte, tal como um aplicativo escrito na linguagem de programação de código de byte JAVA. Em algumas modalidades, o agente de coleta 404 compreende pelo menos um script. Naquelas modalidades, o agente de coleta 404 acumula uma informação ao rodar pelo menos um script na máquina local 10. Em algumas modalidades, o agente de coleta compreende um controle Active X na máquina local 10. Um controle Active X é um objeto de Modelo de Objeto de Componente (COM) que implementa um conjunto de interfaces que permitem que ele pareça e atue como um controle.

[00176] Em uma modalidade, o agente de política 406 transmite o agente de coleta 404 para a máquina local 10. Em uma outra modalidade, uma aplicação 1250 pode armazenar ou armazenar em cache o agente de coleta. A aplicação 1250 então pode transmitir o agente de coleta para uma máquina local 10. Em outras modalidades, uma aplicação 1250 pode interceptar a transmissão de um agente de coleta 404. Ainda em uma outra modalidade, uma aplicação 1250 pode acelerar a entrega de um agente de coleta. Em uma modalidade,

o agente de política 406 requer uma segunda execução do agente de coleta 404 após o agente de coleta 404 ter transmitido uma informação 412 para o agente de política 406. Nesta modalidade, o agente de política 406 pode ter uma informação insuficiente 412 para determinar se a máquina local 10 satisfaz a uma condição em particular. Em outras modalidades, o agente de política 406 requer uma pluralidade de execuções do agente de coleta 404, em resposta à informação recebida 412.

[00177] Em algumas modalidades, o agente de política 406 transmite instruções para o agente de coleta 404 determinando o tipo de informação que o agente de coleta 404 acumula. Naquelas modalidades, um administrador de sistema pode configurar as instruções transmitidas para o agente de coleta 404 a partir do agente de política 406. Isto provê um controle maior em relação ao tipo de informação coletada. Isto também expande os tipos de decisões de controle de acesso que o agente de política 406 pode tomar, devido ao maior controle sobre o tipo de informação coletada. O agente de coleta 404 acumula uma informação 412 incluindo, sem limitação, uma ID de máquina da máquina local 10, tipo de sistema operacional, existência de um "patch" para um sistema operacional, endereços de MAC de placas de rede instaladas, uma marca d'água digital no dispositivo cliente, uma participação em um Diretório Ativo, a existência de uma varredura de vírus, a existência de um firewall pessoal, um cabeçalho de HTTP, um tipo de navegador, um tipo de dispositivo, uma informação de conexão de rede, tal como um endereço de protocolo de internet ou uma faixa de endereços, uma ID de máquina da máquina remota 30, data ou horário de requisição de acesso incluindo ajustes quanto a fusos-horários variáveis, e credenciais de autorização. Em algumas modalidades, uma agente de coleta acumula uma informação para determinar se um aplicativo pode ser acelerado

no cliente usando um programa de aceleração 6120.

[00178] Em algumas modalidades, o tipo de dispositivo é um assistente digital pessoal. Em outras modalidades, o tipo de dispositivo é um telefone celular. Em outras modalidades, o tipo de dispositivo é um computador laptop. Em outras modalidades, o tipo de dispositivo é um computador de mesa. Em outras modalidades, o tipo de dispositivo é um quiosque de Internet.

[00179] Em algumas modalidades, a marca d'água digital inclui uma inserção de dados. Em algumas modalidades, a marca d'água compreende um padrão de dados inserido em um arquivo para a provisão de uma informação de fonte sobre o arquivo. Em outras modalidades, a marca d'água compreende arquivos de comprovação de dados para a provisão de detecção de violação. Em outras modalidades, a marca d'água provê uma informação de direitos autorais sobre o arquivo.

[00180] Em algumas modalidades, a informação de conexão de rede se refere a capacidades de largura de banda. Em outras modalidades, a informação de conexão de rede se refere a um endereço de Protocolo de Internet. Ainda em outras modalidades, a informação de conexão de rede consiste em um endereço de Protocolo de Internet. Em uma modalidade, a informação de conexão de rede compreende uma zona de rede identificando o agente de logon ao qual a máquina local proveu credenciais de autenticação.

[00181] Em algumas modalidades, as credenciais de autenticação incluem vários tipos de informação de autenticação, incluindo, sem limitação, nomes de usuário, nomes de cliente, endereços de cliente, senhas, PINs, amostras de voz, códigos de senha de uso único, dados biométricos, certificados digitais, tíquetes, etc. e combinações dos mesmos. Após o recebimento da informação acumulada 412, o agente de política 406 toma uma decisão de controle de acesso com base na

informação recebida 412.

[00182] Com referência, agora, à figura 4B, um diagrama de blocos descreve uma modalidade de um agente de política 406, incluindo um primeiro componente 420 compreendendo um banco de dados de condição 422 e um agente de logon 424, e incluindo um segundo componente 430 compreendendo um banco de dados de política 432. O primeiro componente 420 aplica uma condição ao banco de dados de condição 422 para a informação recebida sobre a máquina local 10 e determina se a informação recebida satisfaz à condição.

[00183] Em algumas modalidades, uma condição pode requerer que a máquina local 10 execute um sistema operacional em particular para satisfazer à condição. Em algumas modalidades, uma condição pode requerer que a máquina local 10 execute um "patch" de sistema operacional em particular para satisfazer à condição. Ainda em outras modalidades, uma condição pode requerer que a máquina local 10 proveja um endereço de MAC para cada placa de rede instalada para satisfazer à condição. Em algumas modalidades, uma condição pode requerer que a máquina local 10 indique uma participação em um Diretório Ativo em particular para satisfazer à condição. Em uma outra modalidade, uma condição pode requerer que a máquina local 10 execute uma varredura de vírus para satisfazer à condição. Em outras modalidades, uma condição pode requerer que a máquina local 10 execute um firewall pessoal para satisfazer à condição. Em algumas modalidades, uma condição pode requerer que a máquina local 10 compreenda um tipo de dispositivo em particular para satisfazer à condição. Em outras modalidades, uma condição pode requerer que a máquina local 10 estabeleça um tipo em particular de conexão de rede para satisfazer à condição.

[00184] Se a informação recebida satisfizer a uma condição, o primeiro componente 420 armazenará um identificador para aquela

condição em um conjunto de dados 426. Em uma modalidade, a informação recebida satisfaz a uma condição, se a informação tornar a condição verdadeira. Por exemplo, uma condição pode requerer que um sistema operacional em particular seja instalado. Se a máquina local 10 tiver um sistema operacional, a condição será verdadeira e satisfeita. Em uma outra modalidade, a informação recebida satisfaz a uma condição, se a informação tornar a condição falsa. Por exemplo, uma condição pode se endereçar a se existe um spyware na máquina local 10. Se a máquina local 10 não contiver um spyware, a condição será falsa e satisfeita.

[00185] Em algumas modalidades, o agente de logon 424 reside fora do agente de política 406. Em outras modalidades, o agente de logon 424 reside no agente de política 406. Em uma modalidade, o primeiro componente 420 inclui um agente de logon 424, o qual inicia a acumulação de informação sobre a máquina local 10. Em algumas modalidades, o armazenamento de dados inclui as condições para as quais o agente de coleta pode acumular informação. Este armazenamento de dados é distinto do banco de dados de condição 422.

[00186] Em algumas modalidades, o agente de logon 424 inicia uma acumulação de informação pela execução do agente de coleta 404. Em outras modalidades, o agente de logon 424 inicia uma acumulação de informação pela transmissão do agente de coleta 404 para a máquina local 10 para execução na máquina local 10. Ainda em outras modalidades, o agente de logon 424 inicia uma acumulação de informação adicional após receber uma informação 412. Em uma modalidade, o agente de logon 424 também recebe a informação 412. Nesta modalidade, o agente de logon 424 gera o conjunto de dados 426 com base na informação recebida 412. Em algumas modalidades, o agente de logon 424 gera o conjunto de dados 426 pela aplicação de

uma condição a partir do banco de dados 422 para a informação recebida a partir do agente de coleta 404.

[00187] Em uma outra modalidade, o primeiro componente 420 inclui uma pluralidade de agentes de logon 424. Nesta modalidade, pelo menos um da pluralidade de agentes de logon 424 reside em cada domínio de rede a partir do qual uma máquina local 10 pode transmitir uma requisição de recurso. Nesta modalidade, a máquina local 10 transmite a requisição de recurso para um agente de logon 424 em particular. Em algumas modalidades, o agente de logon 424 transmite para o agente de política 406 o domínio de rede a partir do qual a máquina local 10 acessou o agente de logon 424. Em uma modalidade, o domínio de rede a partir do qual a máquina local 10 acessa um agente de logon 424 é referido como a zona de rede da máquina local 10.

[00188] O banco de dados de condição 422 armazena as condições que o primeiro componente 420 aplica à informação recebida. O banco de dados de política 432 armazena as políticas que o segundo componente 430 aplica ao conjunto de dados recebidos 426. Em algumas modalidades, o banco de dados de condição 422 e o banco de dados de política 432 armazenam dados em um banco de dados em conformidade com ODBC. Por exemplo, o banco de dados de condição 422 e o banco de dados de política 432 podem ser providos como um banco de dados ORACLE, fabricado pela Oracle Corporation de Redwood Shores, Calif. Em outras modalidades, o banco de dados de condição 422 e o banco de dados de política 432 podem ser um banco de dados do Microsoft ACCESS ou um banco de dados de servidor Microsoft SQL, fabricado pela Microsoft Corporation de Redmond, Wash.

[00189] Após o primeiro componente 420 aplicar a informação recebida a cada condição no banco de dados de condição 422, o

primeiro componente transmite o conjunto de dados 426 para o segundo componente 430. Em uma modalidade, o primeiro componente 420 transmite apenas o conjunto de dados 426 para o segundo componente 430. Portanto, nesta modalidade, o segundo componente 430 não recebe uma informação 412, apenas identificadores para condições satisfeitas. O segundo componente 430 recebe o conjunto de dados 426 e toma uma decisão de controle de acesso pela aplicação de uma política a partir do banco de dados de política 432, com base nas condições identificadas no conjunto de dados 426.

[00190] Em uma modalidade, o banco de dados de política 432 armazena as políticas aplicadas à informação recebida 412. Em uma modalidade, as políticas armazenadas no banco de dados de política 432 são especificadas, pelo menos em parte, pelo administrador do sistema. Em uma outra modalidade, um usuário especifica pelo menos algumas das políticas armazenadas no banco de dados de política 432. A política ou as políticas específicas de usuário são armazenadas como preferências. O banco de dados de política 432 pode ser armazenado em uma memória volátil ou não volátil, ou, por exemplo, distribuído através de múltiplos servidores.

[00191] Em uma modalidade, uma política permite acessar um recurso apenas se uma ou mais condições forem satisfeitas. Em uma outra modalidade, uma política permite acesso a um recurso, mas proíbe uma transmissão do recurso para a máquina local 10. Uma outra política poderia tornar uma conexão contingente na máquina local 10 que requisita acesso seja em uma rede segura. Em algumas modalidades, o recurso é um programa aplicativo e a máquina local 10 requisitou a execução do programa aplicativo. Em uma destas modalidades, uma política pode permitir uma execução do programa aplicativo na máquina local 10. Em uma outra destas modalidades,

uma política pode permitir que a máquina local 10 receba uma transmissão de arquivos compreendendo o programa aplicativo. Nesta modalidade, a transmissão de arquivos pode ser armazenada e executada em um ambiente de isolamento. Ainda em uma outra destas modalidades, uma política pode permitir apenas a execução do programa aplicativo em uma máquina remota, tal como um servidor de aplicativo, e requerer que a máquina remota transmita dados de saída de aplicativo para a máquina local 10.

[00192] Com referência, agora, à figura 4C, um fluxograma descreve uma modalidade das etapas seguidas pelo agente de política 406 para tomar uma decisão de controle de acesso, com base em uma informação recebida a partir de uma máquina local 10. Mediante o recebimento da informação acumulada sobre a máquina local 10 (Etapa 450), o agente de política 406 gera um conjunto de dados com base na informação (Etapa 452). O conjunto de dados 426 contém identificadores para cada condição satisfeita pela informação recebida 412. O agente de política 406 aplica uma política a cada condição identificada no conjunto de dados 426. Aquele aplicativo produz uma enumeração de recursos os quais a máquina local 10 pode acessar (Etapa 454). O agente de política 406 então apresenta aquela enumeração para a máquina local 10. Em algumas modalidades, o agente de política 406 cria um documento de Linguagem de Marcação de Hipertexto (HTML) usado para apresentação da enumeração para a máquina local.

[00193] Com referência à figura 4D, uma modalidade de uma rede construída de acordo com a invenção é descrita, a qual inclui uma máquina local 10, um agente de coleta 404, um agente de política 406, um banco de dados de política 408, um banco de dados de condição 410, uma máquina local 20, um servidor de sessão 420, um banco de dados de aplicativo armazenado 422, uma máquina remota 30', um

primeiro banco de dados 428, uma máquina remota 30'' e um segundo banco de dados 432. Em uma visão geral breve, quando a máquina local 10 transmite para o servidor de controle de acesso 406 uma requisição 412 para acesso a um programa aplicativo, o agente de coleta 404 se comunica com a máquina local 10, recupera uma informação sobre a máquina local 10 e transmite a informação de máquina local 414 para o agente de política 406. O agente de política 406 toma uma decisão de controle de acesso, conforme discutido acima na figura 4A e na figura 4B. A máquina local 10 recebe uma enumeração de aplicativos disponíveis associados à máquina local 10.

[00194] Em algumas modalidades, o servidor de sessão 420 estabelece uma conexão entre a máquina local 10 e uma pluralidade de sessões de aplicativo associadas à máquina local 10. Em outras modalidades, o agente de política 406 determina que a máquina local 10 tem autorização para recuperar uma pluralidade de arquivos aplicativos compreendendo a aplicativo e para executar o programa aplicativo localmente. Em uma destas modalidades, a máquina remota 30' armazena dados de sessão de aplicativo e uma pluralidade de arquivos aplicativos compreendendo o programa aplicativo. Em uma outra destas modalidades, a máquina local 10 estabelece uma sessão de transmissão de aplicativo com uma máquina remota 30' armazenando os dados de sessão de aplicativo e a pluralidade de arquivos aplicativos compreendendo o programa aplicativo. Em algumas modalidades, o agente de política 406 determina se é para acelerar a entrega da sessão de transmissão pela transmissão de um programa de aceleração 6120 para a máquina local 10. Em algumas modalidades, o agente de política 406 determina se é para acelerar a entrega de arquivos de dados pela transmissão de um programa de aceleração 6120 para a máquina local 10.

[00195] Com referência, agora, à figura 4E, um fluxograma

descreve uma modalidade das etapas seguidas pelo servidor de sessão 420 para a provisão de acesso para a máquina local 10 a suas sessões de aplicativo associadas. O servidor de sessão 420 recebe uma informação sobre a máquina local 10 a partir do agente de política 406 contendo uma decisão de controle de acesso que o agente de política 406 fez (etapa 480). O servidor de sessão 420 gera uma enumeração de aplicativos associados (etapa 482). O servidor de sessão 420 pode conectar a máquina local 10 para um aplicativo associada (etapa 484). Em uma modalidade, a informação também inclui a informação de máquina local 414. Em uma outra modalidade, a informação inclui uma autorização para executar o programa aplicativo localmente.

[00196] O servidor de sessão 420 gera uma enumeração de aplicativos associados (etapa 482). Em algumas modalidades, o agente de política 406 identifica uma pluralidade de sessões de aplicativo já associadas à máquina local 10. Em outras modalidades, o servidor de sessão 420 identifica as sessões de aplicativo armazenadas associadas à máquina local 10. Em algumas destas modalidades, o servidor de sessão 420 automaticamente identifica as sessões de aplicativo armazenadas, mediante o recebimento da informação a partir do agente de política 406. Em uma modalidade, o banco de dados de aplicativo armazenado 422 reside no servidor de sessão 420. Em uma outra modalidade, o banco de dados de aplicativo armazenado 422 reside no agente de política 406.

[00197] O banco de dados de aplicativo armazenado 422 contém dados associados a uma pluralidade de máquinas remotas no banco 38 executando sessões de aplicativo ou provendo acesso a dados de sessão de aplicativo e arquivos aplicativos compreendendo programas aplicativos. Em algumas modalidades, a identificação das sessões de aplicativo associadas à máquina local 10 requer consultar dados

armazenados associados a uma ou mais máquinas remotas. Em algumas destas modalidades, o armazenamento de sessão 420 consulta os dados armazenados associados a uma ou mais máquinas remotas. Em outras destas modalidades, o agente de política 406 consulta os dados armazenados associados a uma ou mais máquinas remotas. Em algumas modalidades, uma primeira sessão de aplicativo roda em uma máquina remota 30' e uma segunda sessão de aplicativo roda em uma máquina remota 30''. Em outras modalidades, todas as sessões de aplicativo rodam em uma máquina remota única 30 no banco 38.

[00198] O servidor de sessão 420 inclui uma informação relacionada a sessões de aplicativo iniciadas por usuários. O servidor de sessão pode ser armazenado em uma memória volátil ou não volátil ou, por exemplo, distribuído através de múltiplos servidores. A Tabela 1 mostra os dados incluídos em uma porção de um servidor de sessão ilustrativo 420:

Sessão de Aplicativo	Sessão de Aplicativo 1	Sessão de Aplicativo 2	Sessão de Aplicativo 3
ID de usuário	Usuário 1	Usuário 2	Usuário 3
ID de cliente	Primeiro cliente		Primeiro cliente
Endereço de cliente	172.16.0.50		172.16.0.50
Status	Ativo	Desconectado	Ativo
Aplicativos	Processador de texto	Banco de dados	Planilha
Número de processo	1	3	2
Servidor	Servidor A	Servidor B	Servidor C
Endereço de servidor	172.16.2.55	172.16.2.55	172.16.2.56

Tabela 1

[00199] O servidor de sessão ilustrativo 420 na Tabela 1 inclui dados associando cada sessão de aplicativo ao usuário que iniciou a sessão de aplicativo, uma identificação do computador cliente 10 ou 20, se houver, a partir do qual o usuário está atualmente conectado à

máquina remota 30', e o endereço de IP daquele computador cliente 10 ou 20. O servidor de sessão ilustrativo 420 também inclui o status de cada sessão de aplicativo. Um status de sessão de aplicativo pode ser, por exemplo, "ativo" (significando que um usuário está conectado à sessão de aplicativo), ou "desconectado" (significando que um usuário não está conectado à sessão de aplicativo). Em uma modalidade alternativa, um status de sessão de aplicativo pode ser regulado para "em execução – desconectado" (significando que o usuário se desconectou da sessão de aplicativo, mas os aplicativos na sessão de aplicativo ainda estão em execução), ou "parado – desconectado" (significando que o usuário está desconectado e os aplicativos na sessão de aplicativo não estão em execução, mas seu estado operacional imediatamente antes da desconexão foi armazenado). O servidor de sessão 420 ainda armazena uma informação indicando os aplicativos 116 que estão em execução em cada sessão de aplicativo e dados indicando cada processo de aplicativo no servidor. Em modalidades nas quais a máquina remota 30' faz parte do banco 38, o servidor de sessão 420 é pelo menos uma parte do armazenamento dinâmico, e também inclui os dados nas duas últimas linhas da Tabela 1 que indicam em qual máquina remota 30 no banco 38 cada aplicativo está/estava em execução, e o endereço de IP daquela máquina remota 30. Em modalidades alternativas, o servidor de sessão 420 inclui um indicador de status para cada aplicativo em cada sessão de aplicativo.

[00200] Por exemplo, no exemplo da Tabela 1, existem três sessões de aplicativo, a Sessão de Aplicativo 1, a Sessão de Aplicativo 2 e a Sessão de Aplicativo 3. A Sessão de Aplicativo 1 está associada ao Usuário 1, que está atualmente usando o terminal 1. O endereço de IP do terminal um é 152.16.2.50. O status da Sessão de Aplicativo 1 é ativo e na Sessão de Aplicativo 1, um programa de

processamento de texto está sendo executado. O programa de processamento de texto está em execução no Servidor A como o número de processo 1. O endereço de IP do servidor A é 152.16.2.55. A Sessão de Aplicativo 2 na Tabela 1 é um exemplo de uma sessão de aplicativo desconectada 118. A Sessão de Aplicativo 2 está associada ao Usuário 2, mas a Sessão de Aplicativo 2 não está conectada a uma máquina local 10 ou 20. A Sessão de Aplicativo 2 inclui um programa de banco de dados que está em execução no Servidor A, no endereço de IP 152.16.2.5 como o número de processo 3. A Sessão de Aplicativo 3 é um exemplo de como um usuário pode interagir com sessões de aplicativo operando em máquinas remotas 30 diferentes. A Sessão de Aplicativo 3 está associada ao Usuário 1, como a Sessão de Aplicativo 1 está. A Sessão de Aplicativo 3 inclui um programa de planilha que está em execução no Servidor B no endereço de IP 152.16.2.56 como o número de processo 2, ao passo que a sessão de aplicativo incluída na Sessão de Aplicativo 1 está em execução no Servidor A.

[00201] Em um outro exemplo, um usuário pode acessar um primeiro programa aplicativo através de uma sessão de aplicativo em execução em uma máquina remota 30', tal como o Servidor A, enquanto em comunicação através de uma sessão de transmissão de aplicativo com uma segunda máquina remota 30'', tal como o Servidor B, para a recuperação de um segundo programa aplicativo a partir da segunda máquina remota 30'' para execução local. O usuário da máquina local 10 pode ter adquirido uma autorização para execução do segundo programa aplicativo localmente, enquanto falha em satisfazer a pré-requisitos de execução do primeiro programa aplicativo.

[00202] Em uma modalidade, o servidor de sessão 420 é configurado para receber uma requisição de desconexão para desconexão das sessões de aplicativo associadas à máquina local 10

e desconecta as sessões de aplicativo em resposta à requisição. O servidor de sessão 420 continua a executar uma sessão de aplicativo após a desconexão da máquina local 10 da sessão de aplicativo. Nesta modalidade, o servidor de sessão 420 acessa o banco de dados de aplicativo armazenado 422 e atualiza um registro de dados associado a cada sessão de aplicativo desconectada, de modo que o registro indique que a sessão de aplicativo associada à máquina local 10 está desconectada.

[00203] Após o recebimento de uma informação de autenticação associada a uma máquina local se conectando à rede, o servidor de sessão 420 consulta o banco de dados de aplicativos armazenados 422 para a identificação de quaisquer sessões de aplicativo ativas que estejam associadas a um usuário da máquina local, mas que são conectadas a uma máquina local diferente, tal como a máquina local 10, se a informação de autenticação estiver associada à máquina local 20, por exemplo. Em uma modalidade, se o servidor de sessão 420 identificar qualquer uma dessas sessões de aplicativo ativas, o servidor de sessão 420 automaticamente desconectará a(s) sessão (sessões) de aplicativo da máquina local 10 e conectará a(s) sessão (sessões) de aplicativo à máquina local atual 20. Em algumas modalidades, a informação de autenticação recebida restringirá as sessões de aplicativo às quais a máquina local 10 pode se reconectar. Em outras modalidades, a informação de autenticação recebida autoriza uma execução de um programa aplicativo na máquina local 20, onde a autorização pode ter sido negada para a máquina local 10. Em uma destas modalidades, o servidor de sessão 420 pode prover à máquina local uma informação de acesos para recuperação do programa aplicativo para execução local.

[00204] Uma requisição é recebida para execução de um aplicativo enumerado (etapa 206). Em uma modalidade, um usuário da máquina

local 10 seleciona um aplicativo para execução a partir da enumeração recebida de aplicativos disponíveis. Em uma outra modalidade, o usuário seleciona um aplicativo para execução independentemente da enumeração recebida. Em algumas modalidades, o usuário seleciona um aplicativo para execução pela seleção de uma representação gráfica do aplicativo apresentado na máquina local 10 por um agente de cliente. Em outras modalidades, o usuário seleciona um aplicativo para execução pela seleção de uma representação gráfica do aplicativo apresentado ao usuário em um servidor da web ou uma outra máquina remota 30''. Em algumas modalidades, uma aplicação 1250 ou um programa de aceleração 6120 acelera a entrega da representação gráfica. Em algumas modalidades, uma aplicação 1250 armazena em cache ou armazena todos e quaisquer aplicativos associados ou porções dos aplicativos associados.

[00205] Ainda em outras modalidades, o usuário requisita acesso a um arquivo. Em uma destas modalidades, a execução de um aplicativo é requerida para a provisão ao usuário de acesso ao arquivo. Em uma outra destas modalidades, o aplicativo é automaticamente selecionado para execução mediante a seleção do arquivo para acesso. Ainda em uma outra destas modalidades, antes da requisição para acesos ao arquivo, o aplicativo é associado a um tipo de arquivo, permitindo uma seleção automática do aplicativo mediante uma identificação de um tipo de arquivo associado ao arquivo requisitado. Em algumas modalidades, uma aplicação 1250 ou um programa de aceleração 6120 pode ser usado para a aceleração da entrega de um ou mais arquivos. Em algumas modalidades, uma aplicação 1250 pode armazenar em cache ou armazenar parte ou todo um arquivo.

[00206] Em uma modalidade, o aplicativo enumerado compreende uma pluralidade de arquivos aplicativos. Em algumas modalidades, a pluralidade de arquivos aplicativos reside em um servidor de arquivo

em separado ou máquina remota 30”. Ainda em outras modalidades, a pluralidade de arquivos aplicativos pode ser transmitida para uma máquina local 10. Ainda em outras modalidades, um arquivo na pluralidade de arquivos aplicativos pode ser executado antes da transmissão de um segundo arquivo na pluralidade de arquivos aplicativos para a máquina local 10. Em algumas modalidades, uma aplicação 1250 ou um programa de aceleração 6120 pode ser usado para aceleração da entrega de um ou mais arquivos aplicativos.

[00207] Em algumas modalidades, a máquina remota 30 recupera uma informação sobre o aplicativo enumerado a partir de uma máquina remota 30’. Em uma destas modalidades, a máquina remota 30 recebe uma identificação de uma máquina remota 30” hospedando uma pluralidade de arquivos aplicativos. Em uma outra destas modalidades, a máquina remota 30 recebe uma identificação de uma localização de uma pluralidade de arquivos aplicativos, a identificação em conformidade com uma Convenção de Denominação Universal (UNC). Ainda em uma outra destas modalidades, a identificação inclui uma localização de rede e um soquete para um protocolo de transmissão de aplicativo.

[00208] Em uma modalidade, a máquina remota 30 recupera um arquivo contendo uma informação sobre o aplicativo enumerado. O arquivo pode incluir uma identificação de uma localização de um servidor hospedando o aplicativo enumerado. O arquivo pode incluir uma identificação de uma pluralidade de versões do aplicativo enumerado. O arquivo pode incluir uma enumeração de uma pluralidade de arquivos aplicativos compreendendo o aplicativo enumerado. O arquivo pode incluir uma identificação de um arquivo comprimido compreendendo uma pluralidade de arquivos aplicativos compreendendo o aplicativo enumerado. O arquivo pode incluir uma identificação de pré-requisitos a serem satisfeitos por uma máquina

executando o aplicativo enumerado. O arquivo pode incluir uma enumeração de arquivos de dados associados ao aplicativo enumerado. O arquivo pode incluir uma enumeração de scripts a serem executados em uma máquina executando o aplicativo enumerado. O arquivo pode incluir uma enumeração de dados de registro associados ao aplicativo enumerado. O arquivo pode incluir uma enumeração de regras para uso em uma modalidade em que o aplicativo enumerado é executado em um ambiente de isolamento. Em uma modalidade, o arquivo pode ser referido como um arquivo de "manifesto". A informação que o arquivo pode conter é descrita em maiores detalhes em relação à figura 21 abaixo.

[00209] Em algumas modalidades, a máquina remota 30 aplica uma política a uma característica identificada da máquina local 10. Em uma destas modalidades, a máquina remota 30 identifica uma versão do aplicativo enumerado para execução, em resposta à característica identificada. Em uma outra destas modalidades, a máquina remota 30 faz uma determinação para execução de uma versão do aplicativo enumerado compatível com uma característica da máquina local 10. Ainda em uma outra destas modalidades, a máquina remota 30 faz uma determinação para execução de uma versão do aplicativo enumerado compatível com um sistema operacional em execução na máquina local 10. Ainda em uma outra destas modalidades, a máquina remota 30 faz uma determinação para execução de uma versão do aplicativo enumerado compatível com um nível de revisão de um sistema operacional na máquina local 10. Em uma destas modalidades, a máquina remota 30 faz uma determinação para execução de uma versão do aplicativo enumerado compatível com uma linguagem especificada por um sistema operacional na máquina local 10.

[00210] Um de um número predeterminado de métodos para

execução do aplicativo enumerado é selecionado, em resposta a uma política, o número predeterminado de métodos incluindo um método para transmissão de aplicativo do aplicativo enumerado (etapa 208). Em uma modalidade, a seleção é feita em resposta a um aplicativo de uma política para as credenciais recebidas associadas à máquina local 10. Em algumas modalidades, a seleção é feita por um agente de política, tal como o agente de política 406 descrito acima na figura 4A, na figura 4B e na figura 4C. Em outras modalidades, a máquina remota 30 recebendo as credenciais e a requisição para execução do aplicativo enumerado ainda compreende um agente de política 406.

[00211] Em uma modalidade, o número predeterminado de métodos inclui um método para execução do aplicativo enumerado em uma máquina remota 30'. Em uma outra modalidade, o aplicativo enumerado de métodos inclui um método para execução do aplicativo enumerado na máquina local 10. Ainda em uma outra modalidade, o número predeterminado de métodos inclui um método para execução do aplicativo enumerado em uma segunda máquina remota 30'.

[00212] Em algumas modalidades, o número predeterminado de métodos inclui um método para a provisão do aplicativo enumerado para a máquina local 10 através de uma sessão de transmissão de aplicativo. Em uma destas modalidades, a máquina local 10 compreende um agente de serviço de transmissão capaz de iniciar uma conexão com uma máquina remota 30' e receber a partir da máquina remota 30' uma transmissão de pacotes de dados transmitidos.

[00213] A transmissão de pacotes de dados pode incluir arquivos aplicativos compreendendo o aplicativo enumerado. Em algumas modalidades, os arquivos aplicativos incluem arquivos de dados associados a um programa aplicativo. Em outras modalidades, os arquivos aplicativos incluem arquivos executáveis requeridos para a

execução do programa aplicativo. Ainda em outras modalidades, os arquivos aplicativos incluem metadados incluindo uma informação sobre os arquivos, tais como localização, exigências de compatibilidade, dados de configuração, dados de registro, identificação de regras de script de execução para uso em ambientes de isolamento, ou exigências de autorização. Em uma modalidade, a transmissão de pacotes de dados é transmitida através de uma conexão de camada de transporte, tal como uma carga útil de um pacote de TCP/IP.

[00214] Em algumas modalidades, o aplicativo transmitido é executado antes da transmissão de cada arquivo aplicativo em uma pluralidade de arquivos aplicativos compreendendo o aplicativo transmitido. Em uma destas modalidades, a execução do aplicativo transmitido começa mediante o recebimento por uma máquina local 10 de um arquivo aplicativo na pluralidade de aplicativos. Em uma outra destas modalidades, a execução do aplicativo transmitido começa mediante o recebimento por uma máquina local 10 de um arquivo aplicativo executável na pluralidade de arquivos aplicativos. Ainda em uma outra destas modalidades, a máquina local 10 executa um primeiro arquivo aplicativo recebido em uma pluralidade de arquivos aplicativos e o primeiro arquivo aplicativo recebido requisita acesso a um segundo arquivo aplicativo na pluralidade de arquivos aplicativos.

[00215] Em uma modalidade, o aplicativo transmitido é executado na máquina local 10 sem residir permanentemente na máquina local 10. Nesta modalidade, o aplicativo transmitido pode ser executado na máquina local 10 e ser removido da máquina local 10 mediante o término do aplicativo transmitido. Em uma outra modalidade, o aplicativo transmitido é executado na máquina local 10 após uma cópia pré-empregada de cada arquivo aplicativo ser armazenada na máquina local 10. Ainda em uma outra modalidade, o aplicativo

transmitido é executado na máquina local 10 após uma cópia de cada arquivo aplicativo ser armazenada em um ambiente de isolamento na máquina local. Ainda em uma outra modalidade, o aplicativo transmitido é executado na máquina local 10 após uma cópia de cada arquivo aplicativo ser armazenada em um cache na máquina local 10.

[00216] Em uma modalidade, o método para transmissão do aplicativo para a máquina local 10 é selecionado a partir do número predeterminado de métodos em resposta a uma determinação que a máquina local 10 pode receber os arquivos aplicativos transmitidos. Em uma outra modalidade, o método para transmissão do aplicativo para a máquina local 10 é selecionado a partir do número predeterminado de métodos em resposta a uma determinação que a máquina local 10 tem uma autoridade para executar os arquivos de aplicativo transmitido localmente.

[00217] Em outras modalidades, o número predeterminado de métodos inclui um método para a provisão de dados de saída de aplicativo para a máquina local 10, os dados de saída de aplicativo gerados a partir de uma execução do aplicativo enumerado em uma máquina remota 30. Em uma destas modalidades, a máquina remota 30 é uma máquina remota 30 que recebe a requisição para execução do aplicativo enumerado. Em uma outra destas modalidades, a máquina remota 30 é uma segunda máquina remota 30', tal como um servidor de arquivo ou um servidor de aplicativo. Em algumas modalidades, o aplicativo enumerado reside na máquina remota 30' executando o aplicativo enumerado. Em outras modalidades, a máquina remota 30' executando o aplicativo enumerado primeiramente recebe o aplicativo enumerado a partir de uma segunda máquina remota 30' através de uma sessão de transmissão de aplicativo. Em uma destas modalidades, a máquina remota 30' compreende um agente de serviço de transmissão capaz de iniciar uma conexão com

uma segunda máquina remota 30' e receber a partir da segunda máquina remota 30' uma transmissão de dados transmitidos. Em uma outra destas modalidades, a segunda máquina remota 30' pode ser identificada usando-se uma técnica de equilíbrio de carga. Ainda em uma outra destas modalidades, a segunda máquina remota 30' pode ser identificada com base na proximidade com a máquina remota 30'. Estas modalidades serão descritas em maiores detalhes em relação à figura 9 abaixo.

[00218] Em algumas modalidades, a máquina remota 30 seleciona a partir do número predeterminado de métodos para execução do aplicativo enumerado um método para transmissão do aplicativo enumerado para a máquina remota 30, executando o aplicativo enumerado na máquina remota 30 e provendo para a máquina local 10 dados de saída de aplicativo gerados pela execução do aplicativo enumerado. Em uma destas modalidades, a máquina remota 30 seleciona o método em resposta a uma avaliação da máquina local 10. Em uma outra destas modalidades, a determinação é feita em resposta artigo absorvente uma aplicação de uma política para avaliação da máquina local 10. Ainda em uma outra destas modalidades, a determinação é feita em resposta a uma avaliação das credenciais recebidas. Em uma modalidade, a máquina remota 30 recebe uma pluralidade de arquivos aplicativos compreendendo o aplicativo enumerado. Em uma outra modalidade, a máquina remota 30 provê dados de saída de aplicativo através de um protocolo de nível de apresentação, tal como um protocolo de nível de apresentação de ICA ou um protocolo de nível de apresentação de Windows de Área de Trabalho Remota ou um protocolo de nível de apresentação de X-Windows.

[00219] Em algumas modalidades, a máquina remota 30 também provê uma informação de acesso associada ao aplicativo enumerado,

a informação de acesso gerada em resposta ao método selecionado. Em uma destas modalidades, a informação de acesso provê uma indicação para a máquina local 10 do método selecionado para execução do programa aplicativo enumerado. Em uma outra destas modalidades, a informação de acesso inclui uma identificação de uma localização do aplicativo enumerado, a identificação em conformidade com uma Convenção de Denominação Universal (UNC). Ainda em uma outra destas modalidades, a informação de acesso inclui uma identificação de um servidor de gerenciamento de sessão.

[00220] Em algumas modalidades, a informação de acesso inclui um tíquete de abertura compreendendo uma informação de autenticação. Em uma destas modalidades, a máquina local 10 pode usar o tíquete de abertura para autenticação da informação de acesso recebida a partir da máquina remota 30. Em uma outra destas modalidades, a máquina local 10 pode usar o tíquete de abertura para autenticar a si mesma para uma segunda máquina remota 30 hospedando o aplicativo enumerado. Ainda em uma outra destas modalidades, a máquina remota 30 inclui um tíquete de abertura na informação de acesso em resposta a uma requisição a partir da máquina local 10 para o tíquete de abertura.

[00221] Com referência, agora, à figura 5, um diagrama de blocos descreve uma modalidade da presente invenção na qual uma máquina local 10 requisita uma execução de um programa aplicativo e uma máquina remota 30 seleciona um método de execução do programa aplicativo. Em uma modalidade, a máquina remota 30 recebe credenciais da máquina local 10. Em uma outra modalidade, a máquina remota 30 recebe uma requisição para uma enumeração de aplicativos disponíveis a partir da máquina local 10.

[00222] Em algumas modalidades, múltiplas máquinas remotas redundantes 30, 30', 30'', 30''' e 30'''' são providas. Em uma destas

modalidades, pode haver, por exemplo, múltiplos servidores de arquivo, múltiplos servidores de gerenciamento de sessão, múltiplas máquinas de estágio, múltiplas interfaces da web, ou múltiplos consoles de suíte de acesso. Em uma outra destas modalidades, se uma máquina remota falhar, uma máquina remota redundante 30 será selecionada para a provisão da funcionalidade da máquina que falhou. Em outras modalidades, embora as máquinas remotas 30, 30', 30'', 30''' e 30'''' e a interface da web 558 e o console de suíte de acesso 520 sejam descritos como máquinas remotas 30 em separado tendo funcionalidades em separado de um servidor de gerenciamento, um servidor de gerenciamento de sessão, uma máquina de estágio, um servidor de arquivo, um servidor da web e um console de suíte de acesso, uma única máquina remota 30 pode ser provida tendo a funcionalidade de todas estas máquinas. Ainda em outras modalidades, uma máquina remota 30 pode prover a funcionalidade e os serviços de uma ou mais das outras máquinas remotas.

[00223] Com referência, agora, à figura 5 em maiores detalhes, um diagrama de blocos descreve uma modalidade de uma máquina remota 30 provendo acesso a um programa aplicativo. Além das interfaces e dos subsistemas descritos acima em relação à figura 1D, a máquina remota 30 ainda pode incluir um serviço de comunicação de gerenciamento 514, um serviço de XML 516 e um serviço de gerenciamento 504. O serviço de gerenciamento 504 pode compreender um subsistema de gerenciamento de aplicativo 506, um subsistema de gerenciamento de servidor 508, um subsistema de gerenciamento de sessão 510 e um subsistema de gerenciamento de licença 512. A máquina remota 30 pode estar em comunicação com um console de suíte de acesso 520.

[00224] Em uma modalidade, o serviço de gerenciamento 504 ainda compreende um subsistema de chamada de procedimento remoto

especializado, o subsistema de Chamada de Procedimento Remoto de Metaquadro (MRFPC) 522. Em algumas modalidades, o subsistema de MRFPC 522 roteia comunicações entre os subsistemas na máquina remota 30, tal como o serviço de XML 516, e o serviço de gerenciamento 504. Em outras modalidades, o subsistema de MRFPC 522 provê uma interface de chamada de procedimento remoto (RPC) para chamada de funções de gerenciamento, entrega chamadas de RPC para o serviço de gerenciamento 504 e retorna os resultados para o subsistema fazendo a chamada.

[00225] Em algumas modalidades, a máquina remota 30 está em comunicação com um agente de protocolo, tal como o agente de política 406 descrito acima na figura 4B. Em uma destas modalidades, a máquina remota 30 está em comunicação com um agente de política 406 residente em uma máquina remota 30'. Em outras modalidades, a máquina remota 30 ainda compreende um agente de política 406.

[00226] A máquina remota 30 pode estar em comunicação com um console de suíte de acesso 520. O console de suíte de acesso 520 pode hospedar ferramentas de gerenciamento para um administrador de uma máquina remota 30 ou de um banco 38. Em algumas modalidades, a máquina remota 30 se comunica com o console de suíte de acesso 520 usando XML. Em outras modalidades, a máquina remota 30 se comunica com o console de suíte de acesso 520 usando um Protocolo de Acesso de Objeto Simples (SOAP).

[00227] Para modalidades tais como aquelas descritas na figura 1D e na figura 5, em que a máquina remota 30 compreende um subconjunto de subsistemas, o serviço de gerenciamento 504 pode compreender uma pluralidade de subsistemas. Em uma modalidade, cada subsistema é um subsistema de linha única ou de linha múltipla. Uma linha é uma transmissão independente de execução rodando em um ambiente de multitarefa. Um subsistema de linha única é capaz de

executar apenas uma linha de uma vez. Um subsistema de linha múltipla pode suportar múltiplas linhas em execução concorrentemente, isto é, um subsistema de linha múltipla pode realizar múltiplas tarefas simultaneamente.

[00228] O subsistema de gerenciamento de aplicativo 506 gerencia uma informação associada a uma pluralidade de aplicativos capazes de serem transmitidos. Em uma modalidade, o subsistema de gerenciamento de aplicativo 506 lida com requisições de outros componentes, tais como requisições para armazenamento, apagamento, atualização, enumeração ou resolução de aplicativos. Em uma outra modalidade, o subsistema de gerenciamento de aplicativo 506 lida com requisições enviadas por componentes relacionados a um aplicativo capaz de ser transmitido. Estes eventos podem ser classificados em três tipos de eventos: publicação de aplicativo, enumeração de aplicativo e abertura de aplicativo, cada um dos quais sendo descrito em maiores detalhes abaixo. Em outras modalidades, o subsistema de gerenciamento de aplicativo 506 ainda compreende um suporte para resolução de aplicativo, publicação de aplicativo e editoração de aplicativo. Em outras modalidades, o subsistema de gerenciamento de aplicativo 506 usa um armazenamento de dados para o armazenamento de propriedades de aplicativo e políticas.

[00229] O subsistema de gerenciamento de servidor 508 lida com configurações específicas para a transmissão de aplicativo em configurações de banco de servidores. Em algumas modalidades, o subsistema de gerenciamento de servidor 508 também lida com eventos que requerem uma recuperação de informação associada a uma configuração de um banco 38. Em outras modalidades, o subsistema de gerenciamento de servidor 508 lida com eventos enviados por outros componentes relacionados a máquinas remotas

provendo acesso a aplicativos através de transmissões contínuas e propriedades daquelas máquinas remotas. Em uma modalidade, o subsistema de gerenciamento de servidor 508 armazena propriedades de máquina remota e propriedades de banco.

[00230] Em algumas modalidades, a máquina remota 30 ainda compreende um ou mais subsistemas de aplicativo comum 524 provendo acesso para um ou mais subsistemas de aplicativo especializado. Estas máquinas remotas 30 também podem ter um ou mais subsistemas de máquina remota comum provendo serviços para um ou mais subsistemas de máquina remota especializada. Em outras modalidades, nenhum subsistema de aplicativo comum 524 é provido, e cada subsistema de aplicativo e máquina remota especializado implementa toda a funcionalidade requerida.

[00231] Em uma modalidade na qual a máquina remota 30 compreende um subsistema de aplicativo comum 524, o subsistema de aplicativo comum 524 gerencia propriedades comuns para aplicativos publicados. Em algumas modalidades, o subsistema de aplicativo comum 524 lida com eventos que requerem uma recuperação de informação associada a aplicativos publicados ou a propriedades comuns. Em outras modalidades, o subsistema de aplicativo comum 524 lida com todos os eventos enviados por outros componentes relacionados a aplicativos comuns e suas propriedades.

[00232] Um subsistema de aplicativo comum 524 pode "publicar" aplicativos para o banco 38, o qual torna cada aplicativo disponível para enumeração e abertura por uma máquina local 10. Geralmente, um aplicativo é instalado em cada máquina remota 30 na qual uma disponibilidade daquele aplicativo é desejada. Em uma modalidade, para publicar um aplicativo, um administrador roda uma ferramenta de administração especificando uma informação, tal como as máquinas remotas 30 hospedando o aplicativo, o nome do arquivo executável

em cada máquina remota, as capacidades requeridas de uma máquina local para execução do aplicativo (por exemplo, áudio, vídeo, encriptação, etc.) e uma lista de usuários que podem usar o aplicativo. Esta informação especificada é categorizada em informação específica de aplicativo e informação comum. Os exemplos de informação específica de aplicativo são: o nome de caminho para acesso ao aplicativo e o nome do arquivo executável para rodar o aplicativo. A informação comum (isto é, dados de aplicativo comum) inclui, por exemplo, o nome amigável para usuário (por exemplo, "Microsoft WORD 2000"), uma identificação única do aplicativo e os usuários do aplicativo.

[00233] A informação específica de aplicativo e a informação comum podem ser enviadas para um subsistema de aplicativo especializado controlando o aplicativo em cada máquina remota 30 hospedando o aplicativo. O subsistema de aplicativo especializado pode escrever a informação específica de aplicativo e a informação comum para um armazenamento persistente 240.

[00234] Quando provido, um subsistema de aplicativo comum 524 também provê uma instalação para gerenciamento dos aplicativos publicados no banco 38. Através de um subsistema de aplicativo comum 524, um administrador pode gerenciar os aplicativos do banco 38 usando uma ferramenta de administração, tal como o console de suíte de acesso 520, para a configuração de grupos de aplicativo e produção de uma hierarquia de árvore de aplicativo daqueles grupos de aplicativo. Cada grupo de aplicativo pode ser representado como uma pasta de arquivos na hierarquia de árvore de aplicativo. Cada pasta de aplicativo na hierarquia de árvore de aplicativo pode incluir uma ou mais outras pastas de aplicativo e instâncias específicas de máquinas remotas. O subsistema de aplicativo comum 524 provê funções para se criarem, moverem, renomearem, apagarem e enumerarem pastas de

aplicativo.

[00235] Em uma modalidade, o subsistema de aplicativo comum 524 suporta o subsistema de gerenciamento de aplicativo 506 na manipulação de enumeração de aplicativo e requisições de resolução de aplicativo. Em algumas modalidades, o subsistema de aplicativo comum 524 provê uma funcionalidade para a identificação de um aplicativo para execução em resposta a um mapeamento entre um tipo de arquivo de dados e um aplicativo para processamento do tipo de arquivo de dados. Em outras modalidades, um segundo subsistema de aplicativo provê a funcionalidade para a associação de tipo de arquivo.

[00236] Em algumas modalidades, a máquina remota 30 ainda pode compreender um subsistema de política. Um subsistema de política inclui uma regra de política para se determinar se um aplicativo pode ser transmitido para uma máquina local 10 mediante uma requisição pela máquina local 10 para execução do aplicativo. Em algumas modalidades, o subsistema de política identifica uma opção de acesso de servidor associada a um aplicativo transmitido publicado no console de suíte de acesso 520. Em uma destas modalidades, o subsistema de política usa a opção de acesso de servidor como uma política no lugar da regra de política.

[00237] O subsistema de monitoração de sessão 510 mantém e atualiza um status de sessão de uma sessão de transmissão de aplicativo associada a uma máquina local 10 e faz cumprir exigências de licença para sessões de transmissão de aplicativo. Em uma modalidade, o subsistema de gerenciamento de sessão 510 monitora sessões e registra eventos, tal como a abertura de um aplicativo ou o término de uma sessão de transmissão de aplicativo. Em uma outra modalidade, o subsistema de monitoração de sessão 510 recebe comunicações, tais como mensagens de batimento cardíaco, transmitidas a partir da máquina local 10 para a máquina remota 30.

Ainda em uma outra modalidade, o subsistema de gerenciamento de sessão 510 responde a consultas sobre sessões a partir de ferramentas de gerenciamento, tais como ferramentas no console de suíte de acesso 520. Em algumas modalidades, o serviço de gerenciamento 504 ainda compreende um subsistema de gerenciamento de licença em comunicação com o subsistema de gerenciamento de sessão para a provisão e a manutenção de licenças para máquinas locais para execução de aplicativos.

[00238] Em uma modalidade, o serviço de gerenciamento 504 provê uma funcionalidade para enumeração de aplicativo e resolução de aplicativo. Em algumas modalidades, o serviço de gerenciamento 504 também provê uma funcionalidade para abertura de aplicativo, monitoração e acompanhamento de sessão, publicação de aplicativo e cumprimento de licença.

[00239] Com referência, agora, à figura 6, um diagrama de blocos descreve uma modalidade de uma máquina remota 30 compreendendo um serviço de gerenciamento provendo uma enumeração de aplicativo. O serviço de gerenciamento 504 pode prover uma enumeração de aplicativo através do uso de uma interface da web interagindo com um serviço de XML 516. Em uma modalidade, o serviço de XML 516 enumera aplicativos para um usuário de uma máquina local 10. Em uma outra modalidade, o serviço de XML 516 implementa a funcionalidade do subsistema de navegador de ICA e o subsistema de vizinhança de programa descritos acima. O serviço de XML 516 pode interagir com um serviço de comunicação de gerenciamento 514. Em uma modalidade, o serviço de XML 516 gera uma requisição de enumeração de aplicativo usando o serviço de comunicação de gerenciamento 514. A requisição de enumeração de aplicativo pode incluir um tipo de cliente indicando um método de execução a ser usado quando da execução do agente de pesquisa. A

requisição de enumeração de aplicativo é enviada para um subsistema de aplicativo comum 524. Em uma modalidade, o subsistema de aplicativo comum 524 retorna uma enumeração de aplicativos associados ao tipicamente pode cliente da requisição de enumeração de aplicativo. Em uma outra modalidade, o subsistema de aplicativo comum 524 retorna uma enumeração de aplicativos disponíveis para o usuário da máquina local 10, a enumeração selecionada em resposta a uma aplicação de uma política a uma credencial associada à máquina local 10. Nesta modalidade, um agente de política 406 pode aplicar a política a credenciais acumuladas por um agente de coleta 404, conforme descrito em relação à figura 4B acima. Ainda em uma outra modalidade, a enumeração de aplicativos é retornada e uma aplicação de uma política à máquina local 10 é adiada até uma execução de um aplicativo enumerado ser requisitada.

[00240] O serviço de gerenciamento 504 pode prover um serviço de resolução de aplicativo para a identificação de uma segunda máquina remota 30' hospedando um aplicativo. Em uma modalidade, a segunda máquina remota 30' é um servidor de arquivo ou um servidor de aplicativo. Em algumas modalidades, o serviço de gerenciamento 504 consulta um arquivo incluindo identificadores para uma pluralidade de máquinas remotas 30 hospedando aplicativos. Em uma modalidade, o serviço de gerenciamento 504 provê o serviço de resolução de aplicativo em resposta a uma requisição de uma máquina local 10 para execução de um aplicativo. Em uma outra modalidade, o serviço de gerenciamento 504 identifica uma segunda máquina remota 30' capaz de implementar um método diferente de execução do aplicativo do que uma primeira máquina remota 30. Em algumas modalidades, o serviço de gerenciamento 504 identifica uma primeira máquina remota 30' capaz de transmitir um programa aplicativo para uma máquina local 10 e uma segunda máquina remota 30' capaz de executar o programa

aplicativo e prover dados de saída de aplicativo gerados em resposta à execução do programa aplicativo para a máquina local 10.

[00241] Em uma modalidade, uma interface da web transmite uma requisição de resolução de aplicativo para o serviço de XML 516. Em uma outra modalidade, o serviço de XML 516 recebe uma requisição de resolução de aplicativo e transmite a requisição para o subsistema de MRFPC 522.

[00242] Em uma modalidade, o subsistema de MRFPC 522 identifica um tipo de cliente incluído com uma requisição de resolução de aplicativo recebida. Em uma outra modalidade, o subsistema de MRFPC aplica uma política ao tipo de cliente e determina a "transmissão" do aplicativo para a máquina local 10. Nesta modalidade, o subsistema de MRFPC 522 pode encaminhar a requisição de resolução de aplicativo para um subsistema de gerenciamento de aplicativo 506. Em uma modalidade, mediante o recebimento da requisição de resolução de aplicativo a partir do subsistema de MRFPC 522, o subsistema de gerenciamento de aplicativo 506 pode identificar uma máquina remota 30''' funcionando como um servidor de gerenciamento de sessão 562 para a máquina local 10. Em algumas modalidades, a máquina local transmite uma mensagem de batimento cardíaco para o servidor de gerenciamento de sessão 562. Em uma outra modalidade, o subsistema de gerenciamento de aplicativo 506 pode identificar uma máquina remota 30' hospedando uma pluralidade de arquivos aplicativos compreendendo o aplicativo a ser transmitido para a máquina local 10.

[00243] Em algumas modalidades, o subsistema de gerenciamento de aplicativo 506 usa um arquivo enumerando uma pluralidade de máquinas remotas hospedando a pluralidade de arquivos aplicativos para a identificação da máquina remota 30'. Em outras modalidades, o subsistema de gerenciamento de aplicativo 506 identifica uma

máquina remota 30' tendo um endereço de IP similar a um endereço de IP da máquina local 10. Ainda em outras modalidades, o subsistema de gerenciamento de aplicativo 506 identifica uma máquina remota 30' tendo um endereço de IP em uma faixa de endereços de IP acessível para a máquina local 10.

[00244] Ainda em uma outra modalidade, o subsistema de MRFPC 522 aplica uma política ao tipo de cliente e determina que o aplicativo pode ser executado em uma máquina remota 30', a máquina remota 30' transmitindo dados de saída de aplicativo gerados por uma execução do aplicativo para a máquina local 10. Nesta modalidade, o subsistema de MRFPC 522 pode encaminhar a requisição de resolução de aplicativo para um subsistema de aplicativo comum 524 para a recuperação de um identificador de um endereço de hospedagem para uma máquina remota 30'. Em uma modalidade, a máquina remota identificada 30' pode transmitir os dados de saída de aplicativo para a máquina local usando um protocolo de nível de apresentação, tal como ICA ou RDP ou X Windows. Em algumas modalidades, a máquina remota 30' recebe o aplicativo a partir de uma segunda máquina remota 30' através de uma sessão de transmissão de aplicativo.

[00245] Em uma modalidade, mediante a conclusão de enumeração de aplicativo enumerado e resolução de aplicativo, uma informação de acesso é transmitida para a máquina local 10 que inclui uma identificação de um método de execução para um aplicativo enumerado e um identificador de uma máquina remota 30' hospedando o aplicativo enumerado. Em uma modalidade em que o serviço de gerenciamento 504 determina que o aplicativo enumerado será executado na máquina local 10, uma interface da web cria e transmite para a máquina local 10 um arquivo contendo uma informação de nome e um arquivo contendo uma informação de nome

resolvido sobre o aplicativo enumerado. Em algumas modalidades, o arquivo pode ser identificado usando-se uma extensão ".rad". A máquina local 10 pode executar o aplicativo enumerado em resposta ao conteúdo do arquivo recebido. A Tabela 2 descreve uma modalidade de informação contida no arquivo:

Campo	Descrição	Fonte
Caminho de UNC	Pontos para um arquivo de manifesto mestre de recipiente no servidor de arquivo.	Serviço de XML
Programa inicial	Programa para abertura a partir do recipiente.	Serviço de XML
Linha de comando	Para abertura de documentos usando FTA.	Serviço de XML
URL de servidor da web		Para mensagens de cliente de RADE para WI. WI config
ID de banco	O banco a que o aplicativo pertence – necessário para mensagens de batimento cardíaco. WI config	
Tíquete de abertura	O cliente de transmissão de aplicativo usa o tíquete de abertura para adquirir uma execução de autorização de licença do programa.	XML/IMA
Informação de abertura de	ICA inserido para "fallback", se for	Arquivo de
"fallback" de ICA	para ser permitido um "fallback".	Serviço de XML

Tabela 2

[00246] O arquivo também pode conter um tíquete de abertura para

uso pela máquina local na execução do aplicativo, conforme mostrado na Tabela 2. Em algumas modalidades, o tíquete de abertura expira após um período de tempo predeterminado. Em uma modalidade, a máquina local provê o tíquete de abertura para uma máquina remota hospedando o aplicativo enumerado a ser executado. O uso do tíquete de abertura para autorização de acesso ao aplicativo enumerado por um usuário da máquina local ajuda a evitar que o usuário re-use o arquivo ou gere uma versão não autorizada do arquivo para acessar de forma inapropriada aplicativos. Em uma modalidade, o tíquete de abertura compreende um número grande gerado randomicamente.

[00247] Conforme descrito acima em relação à figura 2, um método para seleção de um método de execução de um programa aplicativo começa quando credenciais associadas à máquina local 10 ou a um usuário da máquina local 10 são recebidas (etapa 202) e uma enumeração de uma pluralidade de programas aplicativos disponível para a máquina local é provida, em resposta às credenciais recebidas (etapa 204). Uma requisição é recebida para execução do aplicativo enumerado (etapa 206) e um de um número predeterminado de métodos para execução do aplicativo enumerado é selecionado, em resposta a uma política, o número predeterminado de métodos incluindo um método para transmissão de aplicativo do aplicativo enumerado (etapa 208).

[00248] Com referência, agora, à figura 7, um fluxograma descreve uma modalidade das etapas seguidas para se acessar uma pluralidade de arquivos compreendendo um programa aplicativo. Uma máquina local realiza uma análise de pré-abertura da máquina local (etapa 210). Em uma modalidade, a máquina local 10 realiza uma análise de pré-abertura da máquina local (etapa 210). Em uma modalidade, a máquina local 10 realiza uma análise de pré-abertura antes de recuperar e executar uma pluralidade de arquivos aplicativos

compreendendo um programa aplicativo. Em uma outra modalidade, a máquina local 10 realiza uma análise de pré-abertura em resposta a uma indicação recebida que a análise de pré-abertura é uma exigência para autorização para acessar a pluralidade de arquivos aplicativos compreendendo um programa aplicativo.

[00249] Em algumas modalidades, a máquina local 10 recebe, a partir de uma máquina remota 30, uma informação de acesso associada à pluralidade de arquivos aplicativos. Em uma destas modalidades, a informação de acesso inclui uma identificação de uma localização de uma máquina remota 30' hospedando a pluralidade de arquivos aplicativos. Em uma outra destas modalidades, a máquina local 10 recebe uma identificação de uma pluralidade de aplicativos compreendendo uma ou mais versões do programa aplicativo. Ainda em uma outra destas modalidades, a máquina local 10 recebe uma identificação de uma pluralidade de arquivos aplicativos compreendendo um ou mais programas aplicativos. Em outras modalidades, a máquina local 10 recebe uma enumeração de programas aplicativos disponíveis para a máquina local 10 para recuperação e execução. Em uma destas modalidades, a enumeração resulta de uma avaliação da máquina local 10. Ainda em outras modalidades, a máquina local 10 recupera pelo menos uma característica em resposta à identificação recuperada da pluralidade de arquivos aplicativos compreendendo um programa aplicativo.

[00250] Em algumas modalidades, a informação de acesso inclui um tíquete de abertura capaz de autorizar a máquina local a acessar a pluralidade de arquivos aplicativos. Em uma destas modalidades, o tíquete de abertura é provido para a máquina local 10 em resposta a uma avaliação da máquina local 10. Em uma outra destas modalidades, o tíquete de abertura é provido para a máquina local 10 subsequente a uma análise de pré-abertura da máquina local 10

pela máquina local 10.

[00251] Em outras modalidades, a máquina local 10 recupera pelo menos uma característica requerida para execução da pluralidade de arquivos aplicativos. Em uma destas modalidades, a informação de acesso inclui pelo menos uma característica. Em uma outra destas modalidades, a informação de acesso inclui uma localização de um arquivo para recuperação pela máquina local 10, o arquivo enumerando pelo menos uma característica. Ainda em uma outra destas modalidades, o arquivo enumerando pelo menos uma característica ainda compreende uma enumeração da pluralidade de arquivos aplicativos e a identificação de uma máquina remota 30 hospedando a pluralidade de arquivos aplicativos.

[00252] A máquina local 10 determina a existência de pelo menos uma característica na máquina local. Em uma modalidade, a máquina local 10 faz esta determinação como parte da análise de pré-abertura. Em uma outra modalidade, a máquina local 10 determina se a máquina local 10 tem pelo menos uma característica.

[00253] Em uma modalidade, a determinação da existência de pelo menos uma característica na máquina local 10 inclui determinar se um driver de dispositivo está instalado na máquina local. Em uma outra modalidade, a determinação da existência de pelo menos uma característica na máquina local 10 inclui determinar se um sistema operacional está instalado na máquina local 10. Ainda em uma outra modalidade, a determinação da existência de pelo menos uma característica na máquina local 10 inclui determinar se um nível de revisão em particular de um sistema operacional está instalado na máquina local 10.

[00254] Em algumas modalidades, a determinação da existência de pelo menos uma característica na máquina local 10 inclui determinar se a máquina local 10 adquiriu uma autorização para executar um

aplicativo enumerado. Em uma destas modalidades, uma determinação é feita pela máquina local 10 quanto a se a máquina local 10 recebeu uma licença para execução do aplicativo enumerado. Em uma outra destas modalidades, uma determinação é feita pela máquina local 10 quanto a se a máquina local 10 recebeu uma licença para receber através de uma sessão de transmissão de aplicativo uma pluralidade de arquivos aplicativos compreendendo o aplicativo enumerado. Em outras modalidades, a determinação da existência de pelo menos uma característica na máquina local 10 inclui determinar se a máquina local 10 tem largura de banda suficiente disponível para recuperar e executar um aplicativo enumerado.

[00255] Em algumas modalidades, a determinação da existência de pelo menos uma característica na máquina local 10 inclui a execução de um script na máquina local 10. Em outras modalidades, a determinação da existência de pelo menos uma característica na máquina local 10 inclui a instalação de um software na máquina local 10. Ainda em outras modalidades, a determinação da existência de pelo menos uma característica na máquina local 10 inclui a modificação de um registro na máquina local 10. Ainda em outras modalidades, a determinação da existência de pelo menos uma característica na máquina local 10 inclui a transmissão de um agente de coleta 404 para a máquina local 10 para execução na máquina local 10 para acumular credenciais associadas à máquina local 10.

[00256] A máquina local 10 requisita a partir de uma máquina remota 30 uma autorização para execução da pluralidade de arquivos aplicativos, a requisição incluindo um tíquete de abertura (etapa 212). Em algumas modalidades, a máquina local 10 faz a requisição em resposta a uma determinação que pelo menos uma característica existe na máquina local 10. Em uma outra destas modalidades, a máquina local 10 determina que a pluralidade de características existe na

máquina local 10, a pluralidade de características associada a um aplicativo enumerado e uma resposta recebida a uma requisição para execução do aplicativo enumerado. Em uma outra destas modalidades, se a máquina local 10 recebe uma indicação que a autorização para execução dos arquivos aplicativos enumerados depende da existência de pelo menos uma característica na máquina local 10. Em uma modalidade, a máquina local 10 recebeu uma enumeração de programas aplicativos, requisitou uma execução de um aplicativo enumerado e recebeu uma informação de acesso incluindo pelo menos uma característica e um tíquete de abertura autorizando a execução do aplicativo enumerado quando da determinação da existência de pelo menos uma característica na máquina local 10.

[00257] Em uma modalidade, a máquina local 10 recebe a partir da máquina remota 30 uma licença autorizando uma execução da pluralidade de arquivos aplicativos. Em algumas modalidades, a licença autoriza uma execução por um período de tempo especificado. Em uma destas modalidades, a licença requer uma transmissão de uma mensagem de batimento cardíaco para manutenção da autorização para execução da pluralidade de arquivos aplicativos.

[00258] Em uma outra modalidade, a máquina local 10 recebe a partir da máquina remota 30 a licença e um identificador associado a uma máquina remota 30 monitorando a execução da pluralidade de arquivos aplicativos. Em algumas modalidades, a máquina remota é um servidor de gerenciamento de sessão 562, conforme descrito na figura 5. Em uma destas modalidades, o servidor de gerenciamento de sessão 562 inclui um subsistema de gerenciamento de sessão 510 que monitora a sessão associada à máquina local 10. Em outras modalidades, uma máquina remota 30'''' em separado é o servidor de gerenciamento de sessão 562.

[00259] A máquina local 10 recebe e executa a pluralidade de

arquivos aplicativos (etapa 214). Em uma modalidade, a máquina local 10 recebe a pluralidade de arquivos aplicativos através de uma sessão de transmissão de aplicativo. Em uma outra modalidade, a máquina local 10 armazena a pluralidade de arquivos aplicativos em um ambiente de isolamento na máquina local 10. Ainda em uma outra modalidade, a máquina local 10 executa um da pluralidade de arquivos aplicativos antes de receber um segundo da pluralidade de arquivos aplicativos. Em algumas modalidades, uma máquina remota transmite a pluralidade de arquivos aplicativos para uma pluralidade de máquinas locais, cada máquina local na pluralidade tendo estabelecido uma sessão de transmissão de aplicativo em separado com a máquina remota.

[00260] Em algumas modalidades, a máquina local 10 armazena a pluralidade de arquivos aplicativos em um cache e atrasa a execução dos arquivos aplicativos. Em uma destas modalidades, a máquina local 10 recebe uma autorização para execução dos arquivos aplicativos durante um período de tempo predefinido. Em uma outra destas modalidades, a máquina local 10 recebe uma autorização para execução dos arquivos aplicativos durante o período de tempo predefinido, quando a máquina local 10 carecer de acesso a uma rede. Em outras modalidades, a máquina local armazena a pluralidade de arquivos aplicativos em um cache. Em uma destas modalidades, o cliente de transmissão de aplicativo 552 estabelece uma sessão de transmissão de aplicativo interna para recuperação da pluralidade de arquivos aplicativos a partir do cache. Em uma outra destas modalidades, a máquina local 10 recebe uma autorização para execução dos arquivos aplicativos durante um período de tempo predefinido quando a máquina local 10 carecer de acesso a uma rede.

[00261] A máquina local 10 transmite pelo menos uma mensagem de batimento cardíaco para uma máquina remota (etapa 216). Em

algumas modalidades, a máquina local 10 transmite pelo menos uma mensagem de batimento cardíaco para reter uma autorização para execução da pluralidade de arquivos aplicativos compreendendo o aplicativo enumerado. Em outras modalidades, a máquina local 10 transmite pelo menos uma mensagem de batimento cardíaco para reter uma autorização para recuperação de um arquivo aplicativo na pluralidade de arquivos aplicativos. Ainda em outras modalidades, a máquina local 10 recebe uma licença autorizando uma execução da pluralidade de arquivos aplicativos durante o período de tempo predeterminado.

[00262] Em algumas modalidades, a máquina local 10 transmite a mensagem de batimento cardíaco para uma segunda máquina remota 30"". Em uma destas modalidades, a segunda máquina remota 30"" pode compreender um servidor de gerenciamento de sessão 562 monitorando a recuperação e a execução da pluralidade de arquivos aplicativos. Em uma outra destas modalidades, a segunda máquina remota 30"" pode renovar uma licença autorizando uma execução da pluralidade de arquivos aplicativos, em resposta à mensagem de batimento cardíaco transmitida. Ainda em uma outra destas modalidades, a segunda máquina remota 30"" pode transmitir para a máquina local 10 um comando em resposta à mensagem de batimento cardíaco transmitida.

[00263] Com referência de volta à figura 5, a máquina local 10 pode incluir um cliente de transmissão de aplicativo 552, um serviço de transmissão 554 e um ambiente de isolamento 556. O cliente de transmissão de aplicativo 552 pode ser um programa executável. Em algumas modalidades, o cliente de transmissão de aplicativo 552 pode ser capaz de abrir um outro programa executável. Em outras modalidades, o cliente de transmissão de aplicativo 552 pode iniciar o serviço de transmissão 554. Em uma destas modalidades, o cliente de

transmissão de aplicativo 552 pode prover o serviço de transmissão 554 com um parâmetro associado à execução de um programa aplicativo. Em uma outra destas modalidades, o cliente de transmissão de aplicativo 552 pode iniciar o serviço de transmissão 554 usando uma chamada de procedimento remoto.

[00264] Em uma outra modalidade, a máquina local 10 requisita uma execução de um programa aplicativo e recebe uma informação de acesso a partir de uma máquina remota 30 com referência a uma execução. Em uma outra modalidade, o cliente de transmissão de aplicativo 552 recebe a informação de acesso. Ainda em uma outra modalidade, o cliente de transmissão de aplicativo 552 provê uma informação de acesso ao serviço de transmissão 554. Ainda em uma outra modalidade, a informação de acesso inclui uma identificação de uma localização de um arquivo associado a uma pluralidade de arquivos aplicativos compreendendo o programa aplicativo.

[00265] Em uma modalidade, o serviço de transmissão 554 recupera um arquivo associado a uma pluralidade de arquivos aplicativos. Em algumas modalidades, o arquivo recuperado inclui uma identificação de uma localização da pluralidade de arquivos aplicativos. Em uma destas modalidades, o serviço de transmissão 554 recupera a pluralidade de arquivos aplicativos. Em uma outra destas modalidades, o serviço de transmissão 554 executa a pluralidade recuperada de arquivos aplicativos na máquina local 10. Em outras modalidades, o serviço de transmissão 554 transmite as mensagens de batimento cardíaco para uma máquina remota para manutenção da autorização para recuperação e execução da pluralidade de arquivos aplicativos.

[00266] Em algumas modalidades, o arquivo recuperado inclui uma identificação de uma localização de mais de uma pluralidade de arquivos aplicativos, cada pluralidade de arquivos aplicativos

compreendendo um programa aplicativo diferente. Em uma destas modalidades, o serviço de transmissão 554 recupera a pluralidade de arquivos aplicativos compreendendo o programa aplicativo compatível com a máquina local 10. Em uma outra destas modalidades, o serviço de transmissão 554 recebe uma autorização para recuperação de uma pluralidade em particular de arquivos aplicativos, em resposta a uma avaliação da máquina local 10.

[00267] Em algumas modalidades, a pluralidade de arquivos aplicativos é comprimida e armazenada em um servidor de arquivo em um arquivo de armazenamento de backup, tal como CAB, ZIP, SIT, TAR, JAR ou outro arquivo de armazenamento de backup. Em uma modalidade, uma pluralidade de arquivos aplicativos armazenada em um arquivo de armazenamento de backup compreende um programa aplicativo. Em uma outra modalidade, múltiplas pluralidades de arquivos aplicativos armazenadas em um arquivo de armazenamento de backup compreendem, cada uma, diferentes versões de um programa aplicativo. Ainda em uma outra modalidade, múltiplas pluralidades de arquivos aplicativos armazenadas em um arquivo de armazenamento de backup compreendem, cada uma, programas aplicativos diferentes. Em algumas modalidades, um arquivo de armazenamento de backup inclui metadados associados a cada arquivo na pluralidade de arquivos aplicativos. Em uma destas modalidades, o serviço de transmissão 554 gera uma estrutura de diretório em resposta aos metadados incluídos. Conforme será descrito em maiores detalhes em relação à figura 12 abaixo, os metadados podem ser usados para satisfazerem às requisições por programas aplicativos para enumeração de diretório.

[00268] Em uma modalidade, o serviço de transmissão 554 descomprime um arquivo de armazenamento de backup para a aquisição da pluralidade de arquivos aplicativos. Em uma outra

modalidade, o serviço de transmissão 554 determina se uma cópia local de um arquivo na pluralidade de arquivos aplicativos existe em um cache na máquina local 10, antes da recuperação do arquivo a partir da pluralidade de arquivos aplicativos. Ainda em uma outra modalidade, o driver de filtro de sistema de arquivo 564 determina se a cópia local existe no cache. Em algumas modalidades, o serviço de transmissão 554 modifica uma entrada de registro antes da recuperação de um arquivo na pluralidade de arquivos aplicativos.

[00269] Em algumas modalidades, o serviço de transmissão 554 armazena uma pluralidade de arquivos aplicativos em um cache na máquina local 10. Em uma destas modalidades, o serviço de transmissão 554 pode prover uma funcionalidade para armazenamento em cache de uma pluralidade de arquivos aplicativos, mediante o recebimento de uma requisição para armazenar em cache a pluralidade de arquivos aplicativos. Em uma outra destas modalidades, o serviço de transmissão 554 pode prover uma funcionalidade para assegurar um cache na máquina local 10. Em uma outra destas modalidades, o serviço de transmissão 554 pode usar uma longitudinal para ajuste de um tamanho e uma localização do cache.

[00270] Em algumas modalidades, o serviço de transmissão 554 cria um ambiente de isolamento 556 na máquina local 10. Em uma destas modalidades, o serviço de transmissão 554 usa uma interface de programação de aplicativo de ambiente de isolamento para a criação do ambiente de isolamento 556. Em uma outra destas modalidades, o serviço de transmissão 554 armazena a pluralidade de arquivos aplicativos no ambiente de isolamento 556. Em ainda uma outra destas modalidades, o serviço de transmissão 554 executa um arquivo na pluralidade de arquivos aplicativos no ambiente de isolamento. Em ainda uma outra destas modalidades, o serviço de

transmissão 554 executa o programa aplicativo no ambiente de isolamento.

[00271] Para modalidades nas quais a autorização é recebida para execução de um aplicativo na máquina local 10, a execução do aplicativo pode ocorrer em um ambiente de isolamento 556. Em algumas modalidades, uma pluralidade de arquivos aplicativos compreendendo o aplicativo é armazenada na máquina local 10, antes da execução do aplicativo. Em outras modalidades, um subconjunto da pluralidade de arquivos aplicativos é armazenado na máquina local 10, antes da execução do aplicativo. Ainda em outras modalidades, a pluralidade de arquivos aplicativos não reside no ambiente de isolamento 556. Ainda em outras modalidades, um subconjunto da pluralidade de arquivos aplicativos não reside na máquina local 10. Independentemente de um subconjunto da pluralidade de arquivos aplicativos ou cada arquivo aplicativo na pluralidade de arquivos aplicativos residir na máquina local 10 ou em um ambiente de isolamento 556, em algumas modalidades, um arquivo aplicativo na pluralidade de arquivos aplicativos pode ser executado em um ambiente de isolamento 556.

[00272] O ambiente de isolamento 556 pode consistir em um sistema de núcleo capaz de prover uma Virtualização de Sistema de Arquivo, uma Virtualização de Sistema de Registro e uma Virtualização de Objeto Denominado para redução de questões de compatibilidade de aplicativo, sem requerer qualquer mudança no código fonte do aplicativo. O ambiente de isolamento 556 pode redirecionar as requisições de recurso de aplicativo usando-se enganchamento no modo de usuário para registro e em virtualização de objeto denominado, e no kernel usando um driver de filtro de sistema de arquivo para virtualização de sistema de arquivo. O que vem a seguir é uma descrição de algumas modalidades de um

ambiente de isolamento 556.

[00273] Com referência, agora, à figura 8A, uma modalidade de um computador rodando sob o controle de um sistema operacional 100 que tem uma compatibilidade de aplicativo reduzida e problemas de sociabilidade de aplicativo é mostrada. O sistema operacional 100 torna disponíveis vários recursos nativos para os programas aplicativos 112, 114 através de sua camada de sistema 108. A vista de recursos concretizados pela camada de sistema 108 será denominada o "escopo do sistema". De modo a se evitar um acesso conflitante para recursos nativos 102, 104, 106, 107 pelos programas aplicativos 112, 114, um ambiente de isolamento 200 é provido. Conforme mostrado na figura 8A, o ambiente de isolamento 200 inclui uma camada de isolamento de aplicativo 220 e uma camada de isolamento de usuário 240. Conceitualmente, o ambiente de isolamento 200 provê, através da camada de isolamento de aplicativo 220, um programa aplicativo 112, 114 com uma vista única de recursos nativos, tal como o sistema de arquivo 102, o registro 104, objetos 106 e nomes de janela 107. Cada camada de isolamento modifica a visão de recursos nativos providos para um aplicativo. A visão modificada de recursos nativos provida por uma camada será referida como o "escopo de isolamento" de camada. Conforme mostrado na figura 8A, a camada de isolamento de aplicativo inclui dois escopos de isolamento de aplicativo 222, 224. O escopo 222 representa a visão de recursos nativos providos para o aplicativo 112 e o escopo 224 representa a visão de recursos nativos providos para o aplicativo 114. Assim, na modalidade mostrada na figura 8A, o APP1 112 é provido com uma visão específica do sistema de arquivo 102', enquanto o APP2 114 é provido com uma outra visão do sistema de arquivo 102'', a qual é específica para ele. Em algumas modalidades, a camada de isolamento de aplicativo 220 provê uma visão específica de recursos nativos 102, 104, 106, 107 para cada

programa aplicativo individual em execução no topo do sistema operacional 100. Em outras modalidades, os programas aplicativos 112, 114 podem ser agrupados em conjuntos e, nestas modalidades, a camada de isolamento de aplicativo 220 provê uma visão específica de recursos nativos para cada conjunto de programas aplicativos. Os programas aplicativos em conflito podem ser colocados em grupos separados para melhoria da compatibilidade e sociabilidade de aplicativos. Ainda em outras modalidades, os aplicativos pertencentes a um conjunto podem ser configurados por um administrador. Em algumas modalidades, um escopo de inserção de "passagem" pode ser definido, o qual corresponde exatamente ao escopo de sistema. Em outras palavras, os aplicativos em execução e um escopo de isolamento de passagem operam diretamente no escopo do sistema.

[00274] Em algumas modalidades, o escopo de isolamento de aplicativo é adicionalmente dividido em subescopos em camadas. O subescopo principal contém o escopo de isolamento de abertura de base e os subescopos adicionais contêm várias modificações para este escopo que podem ser visíveis para múltiplas instâncias em execução do aplicativo. Por exemplo, um subescopo pode conter modificações para o escopo que concretizam uma mudança no nível de "patch" do parâmetro ou a instalação ou a remoção de recursos adicionais. Em algumas modalidades, o conjunto de subescopos adicionais que é tornado visível para uma instância do aplicativo em execução é configurável. Em algumas modalidades, aquele conjunto de subescopos visíveis é o mesmo para todas as instâncias do aplicativo em execução, independentemente de o usuário em nome do qual o aplicativo está em execução. Em outras, o conjunto de subescopos visíveis pode variar para usuários diferentes executando o aplicativo. Ainda em outras modalidades, vários conjuntos de subescopos podem ser definidos e o usuário pode ter uma escolha

quanto a qual conjunto usar. Em algumas modalidades, os subescopos podem ser descartados quando não mais necessários. Em algumas modalidades, as modificações contidas em um conjunto de subescopos podem ser fundidas em conjunto para a formação de um subescopo único.

[00275] Com referência, agora, à figura 8B, um computador de usuário múltiplo, tendo uma compatibilidade de aplicativo reduzida e problemas de sociabilidade de aplicativo, é descrito. O computador de usuário múltiplo inclui os recursos nativos 102, 104, 106, 107 na camada de sistema 108, bem como o ambiente de isolamento 200 discutido imediatamente acima. A camada de isolamento de aplicativo 220 funciona conforme discutido acima, provendo um aplicativo ou um grupo de aplicativos com uma visão modificada de recursos nativos. A camada de isolamento de usuário 240 conceitualmente provê um programa aplicativo 112, 114 com uma visão de recursos nativos que é adicionalmente alterada com base em uma identidade de usuário do usuário em cujo nome o aplicativo é executado. Conforme mostrado na figura 8B, a camada de isolamento de usuário 240 pode ser considerada para compreender vários escopos de isolamento 242', 242'', 242''', 242'''', 242''''' (geralmente 242). Um escopo de isolamento de usuário 242 provê uma vista específica de usuário de vistas específicas de aplicativo de recursos nativos. Por exemplo, o APP1 112 em execução em uma sessão de usuário 110 em nome do usuário "a" é provido com uma vista de sistema de arquivo 102'(a) que é alterada ou modificada pelo escopo de isolamento de usuário 242' e o escopo de isolamento de aplicativo 222.

[00276] Dito de uma outra forma, a camada de isolamento de usuário 240 altera a vista de recursos nativos para cada usuário individual pela "colocação em camadas" de uma modificação de vista específica de usuário por um escopo de isolamento de usuário 242'

"no topo de" uma modificação de vista específica de aplicativo provida por um escopo de isolamento de aplicativo 222, o qual por sua vez é "colocado em camadas no topo da" vista pansistêmica de recursos nativos providos pela camada de sistema. Por exemplo, quando a primeira instância de APP1 112 acessa uma entrada no banco de dados de registro 104, a visão do banco de dados de registro específica para a primeira sessão de usuário e o aplicativo 104'(a) é consultada. Se uma chave de registro requisitada for encontrada na visão específica de usuário do registro 104'(a), aquela chave de registro será retornada para o APP1 112. Caso não, a visão do banco de dados de registro específica para o aplicativo 104' é consultada. Se a chave de registro requisitada for encontrada na visão específica de aplicativo do registro 104', aquela chave de registro será retornada para o APP1 112. Caso não, então, a chave de registro armazenada no banco de dados de registro 104 na camada de sistema 108 (isto é, a chave de registro nativa) é retornada para o APP1 112.

[00277] Em algumas modalidades, a camada de isolamento de usuário 240 provê um escopo de isolamento para cada usuário individual. Em outras modalidades, a camada de isolamento de usuário 240 provê um escopo de isolamento para um grupo de usuários, os quais podem ser definidos por papéis na organização ou podem ser predeterminados por um administrador. Ainda em outras modalidades, nenhuma camada de isolamento de usuário 240 é provida. Nestas modalidades, a visão de recursos nativos vista por um programa aplicativo é aquela provida pela camada de isolamento de aplicativo 220. O ambiente de isolamento 200, embora descrito em relação a computadores de usuário múltiplo suportando uma execução concorrente de programas aplicativos por vários usuários, também pode ser usado em computadores de usuário único para se dirigir a problemas de compatibilidade de aplicativo e sociabilidade resultantes

de uma execução seqüencial de programas aplicativos no mesmo sistema de computador por usuários diferentes, e aqueles problemas resultantes de instalação e execução de programas incompatíveis pelo mesmo usuário.

[00278] Em algumas modalidades, o escopo de isolamento de usuário é adicionalmente dividido em subescopos. As modificações pelo escopo de isolamento de usuário para a visão apresentada para um aplicativo em execução naquele escopo é o agregado das modificações contidas em cada subescopo no escopo. Os subescopos são colocados em camadas no topo uns dos outros, e na visão agregada as modificações em um recurso em um subescopo mais alto suprimem as modificações no mesmo recurso em camadas mais baixas.

[00279] Em algumas destas modalidades, um ou mais destes subescopos podem conter modificações para a vista que sejam específicas para o usuário em algumas destas modalidades, um ou mais subescopos podem conter modificações para a visão que são específicas para conjuntos de usuários, os quais podem ser definidos pelos administradores de sistema ou definidos como um grupo de usuário no sistema operacional. Em algumas destas modalidades, um destes subescopos pode conter modificações para a visão que são específicas para a sessão de login em particular e, daí, que são descartadas quando a sessão termina. Em algumas destas modalidades, as mudanças em recursos nativos por instâncias de aplicativo associadas ao escopo de isolamento de usuário sempre afetam um destes subescopos, e nas outras mudanças aquelas modalidades podem afetar subescopos diferentes, dependendo do recurso em particular mudado.

[00280] A arquitetura conceitual descrita acima permite que um aplicativo em execução em nome de um usuário seja apresentado com uma visão agregada ou unificada virtualizada de recursos nativos,

específicos para aquela combinação de aplicativo e usuário. Esta visão agregada pode ser referida como "escopo virtual". A instância de aplicativo em execução em nome de um usuário é apresentada com uma vista simples de recursos nativos refletindo todas as instâncias virtualizadas operativas dos recursos nativos. Conceitualmente, esta vista agregada consiste primeiramente no conjunto de recursos nativos providos pelo sistema operacional no escopo do sistema, sobreposta com as modificações concretizadas no escopo de isolamento de aplicativo aplicável ao parâmetro em execução, ainda sobreposta com as modificações concretizadas no escopo de isolamento de usuário aplicável ao aplicativo em execução em nome do usuário. Os recursos nativos no escopo do sistema são caracterizados por serem comuns a todos os usuários e aplicativos no sistema, exceto onde as permissões de sistema operacional negarem acesso a usuários ou aplicativos específicos. As modificações na vista de recurso concretizadas em um escopo de isolamento de aplicativo são caracterizadas como sendo comuns a todas as instâncias de aplicativos associadas àquele escopo de isolamento de aplicativo. As modificações na vista de recurso concretizada no escopo de isolamento de usuário são caracterizadas como sendo comuns a todos os aplicativos associados ao escopo de isolamento de aplicativo aplicável que estão em execução em nome do usuário associado ao escopo de isolamento de usuário.

[00281] Este conceito pode ser estendido para subescopos; as modificações na vista de recurso concretizada em um subescopo de usuário são comuns a todos os aplicativos associados ao subescopo de isolamento aplicável em execução em nome de um usuário, ou um grupo de usuários, associado a um subescopo de isolamento de usuário. Por toda esta descrição, deve ser entendido que sempre que uma referência geral for feita ao "escopo", pretende-se fazer referência também a subescopos, quando aqueles existirem.

[00282] Quando um aplicativo requisita uma enumeração de um recurso nativo, tal como uma porção do sistema de arquivo ou banco de dados de registro, uma enumeração virtualizada é construída primeiramente pela enumeração da instância de "escopo de sistema" do recurso nativo, isto é, a instância encontrada na camada de sistema, se houver. Em seguida, a instância de "escopo de aplicativo" do recurso requisitado, isto é, a instância encontrada no escopo de isolamento de aplicativo apropriado, se houver, é enumerada. Quaisquer recursos enumerados encontrados no escopo de isolamento de aplicativo são adicionados à vista. Se o recurso enumerado já existir na vista (porque ele foi apresentado no escopo do sistema também), ele será substituído pela instância do recurso encontrada no escopo de isolamento de aplicativo. De modo similar, a instância de "escopo de usuário" do recurso requisitado, isto é, a instância encontrada no escopo de isolamento de usuário apropriado, se houver, é enumerada. De novo, quaisquer recursos enumerados encontrados no escopo de isolamento de usuário são adicionados à vista. Se o recurso nativo já existir na vista (porque foi apresentado no escopo do sistema ou no escopo de isolamento de aplicativo apropriado), ele será substituído pela instância do recurso encontrada no escopo de isolamento de usuário. Desta maneira, qualquer enumeração de recursos nativos apropriadamente refletirá uma visualização dos recursos nativos enumerados. Conceitualmente, a mesma abordagem se aplica a uma enumeração de escopo de isolamento que compreenda múltiplos subescopos. Os subescopos individuais são enumerados, com recursos de subescopos mais altos substituindo instâncias de combinação dos subescopos mais baixos na vista agregada.

[00283] Em outras modalidades, uma enumeração pode ser realizada a partir da camada de escopo de isolamento de usuário

descendo para a camada de sistema, ao invés de o inverso. Nestas modalidades, o escopo de isolamento de usuário é enumerado. Então, o escopo de isolamento de aplicativo é enumerado em quaisquer instâncias de recurso aparecendo no escopo de isolamento de aplicativo que não foram enumeradas no escopo de isolamento de usuário são adicionadas à vista agregada que estiver sob construção. Um processo similar pode ser repetido para recursos aparecendo apenas no escopo de sistema.

[00284] Ainda em outras modalidades, todos os escopos de isolamento podem ser simultaneamente enumerados e as respectivas enumerações combinadas.

[00285] Se um aplicativo tentar abrir uma instância existente de um recurso nativo com nenhuma intenção de modificar aquele recurso, a instância específica que é retornada para o aplicativo é uma que é encontrada no escopo virtual, ou, de forma equivalente, a instância que apareceria na enumeração virtualizada da origem do recurso requisitado. Do ponto de vista do ambiente de isolamento, o aplicativo é dito como estando requisitando a abertura de um "recurso virtual", e a instância em particular de recurso nativo usada para satisfazer àquela requisição é dita como sendo o "recurso literal" correspondente ao recurso requisitado.

[00286] Se um aplicativo em execução em nome de um usuário tentar abrir um recurso e indicar que ele está fazendo isso com a intenção de modificar aquele recurso, àquela instância de aplicativo normalmente é dada uma cópia privada daquele recurso para modificação, já que os recursos no escopo de isolamento de aplicativo e no escopo do sistema são comuns a aplicativos em execução em nome de outros usuários. Tipicamente, uma cópia de escopo de usuário do recurso é feita, a menos que a instância de escopo de usuário já exista. A definição da vista agregada provida por um escopo

virtual significa que o ato de copiar um recurso de escopo de aplicativo ou de escopo de sistema para um escopo de isolamento de usuário não muda a vista agregada provida pelo escopo virtual para o usuário e o aplicativo em questão, nem para qualquer outro usuário, nem para qualquer outra instância de aplicativo. Modificações subsequentes para o recurso copiado pela instância de aplicativo em execução em nome do usuário não afetam a vista agregada de qualquer instância de aplicativo que não compartilhe o mesmo escopo de isolamento de usuário. Em outras palavras, aquelas modificações não mudam a vista agregada de recursos nativos para outros usuários, ou para instâncias de aplicativo não associadas ao mesmo escopo de isolamento de aplicativo.

[00287] Os aplicativos podem ser instalados em um escopo de isolamento em particular (descrito abaixo em maiores detalhes). Os aplicativos que são instalados em um escopo de isolamento são sempre associados àquele escopo. Alternativamente, os aplicativos podem ser abertos em um escopo de isolamento em particular ou em vários escopos de isolamento. Com efeito, um aplicativo é aberto e associado a um ou mais escopos de isolamento. O escopo de isolamento associado ou os escopos provêem acesso com uma vista em particular de recursos nativos. Os aplicativos também podem ser abertos no escopo do sistema, isto é, eles podem ser associados a nenhum escopo de isolamento. Isto permite a execução seletiva de aplicativos de sistema operacional, tal como o Internet Explorer, bem como aplicativos de terceiros, em um ambiente de isolamento.

[00288] Esta capacidade de abrir aplicativos em um escopo de isolamento independentemente de onde o aplicativo está instalado mitiga questões de compatibilidade e sociabilidade de aplicativo, sem requerer uma instalação em separado do parâmetro no escopo de isolamento. A capacidade de seletivamente abrir aplicativos instalados

em diferentes escopos de isolamento provê a capacidade de ter aplicativos os quais precisam de aplicativos de ajuda (tais como Word, Notepad, etc.) tendo aqueles aplicativos de ajuda abertos com os mesmos conjuntos de regra.

[00289] Ainda, a capacidade de abrir um aplicativo em ambientes isolados múltiplos permite uma melhor integração entre aplicativos isolados e aplicativos comuns.

[00290] Com referência, agora, à figura 8C, e em uma visão geral breve, um método para associação de um processo a um escopo de isolamento inclui as etapas de abertura do processo em um estado suspenso (etapa 882). As regras associadas ao escopo de isolamento desejado são recuperadas (etapa 884) e um identificador para o processo e as regras recuperadas são armazenados em um elemento de memória (etapa 886) e o processo suspenso é retomado (etapa 888). As chamadas subseqüentes para acesso aos recursos nativos feitas pelo processo são interceptadas ou enganchadas (etapa 890) e as regras associadas ao identificador de processo, se houver, são usadas para virtualização do acesso ao recurso requisitado (etapa 892).

[00291] Ainda com referência à figura 8C, e em maiores detalhes, um processo é aberto em um estado suspenso (etapa 882). Em algumas modalidades, um programa de abertura personalizado é usado para a realização desta tarefa. Em algumas destas modalidades, o programa para abertura é especificamente projetado para abrir um processo em um escopo de isolamento selecionado. Em outras modalidades, o programa para abertura aceita como uma entrada uma especificação do escopo de isolamento desejado, por exemplo, por uma opção de linha de comando.

[00292] As regras associadas ao escopo de isolamento desejado são recuperadas (etapa 884). Em algumas modalidades, as regras são

recuperadas a partir de um elemento de armazenamento persistente, tal como uma unidade de disco rígido ou um outro elemento de memória de estado sólido. As regras podem ser armazenadas como um banco de dados relacional, um banco de dados de arquivo simples, um banco de dados de árvore estruturada, uma estrutura de árvore binária ou outra estrutura de dados persistente. Em outras modalidades, as regras podem ser armazenadas em uma estrutura de dados especificamente configurada para o armazenamento delas.

[00293] Um identificador para o processo, tal como uma ID de processo (PID) e as regras recuperadas são armazenadas em um elemento de memória (etapa 886). Em algumas modalidades, um driver de modo de kernel é provido, que recebe mensagens de sistema operacional concernindo à nova criação de processo. Nestas modalidades, a PID e as regras recuperadas podem ser armazenadas no contexto do driver. Em outras modalidades, um driver de filtro de sistema de arquivo ou um minifiltro é provido, que intercepta as requisições de recurso nativo. Nestas modalidades, a PID e as regras recuperadas podem ser armazenadas no filtro. Em outras modalidades ainda, toda a interceptação é realizada pelo enganchamento de modo de usuário e nenhuma PID é armazenada de forma alguma. As regras são carregadas pelo aparelho de enganchamento de modo de usuário durante a inicialização de processo, e nenhum outro componente precisa saber as regras que se aplicam à PID, porque uma associação de regra é realizada inteiramente no processo.

[00294] O processo suspenso é retomado (etapa 888) e chamadas subseqüentes para acesso aos recursos nativos feitas pelo processo são interceptadas ou enganchadas (etapa 890) e as regras associadas ao identificador de processo, se houver, são usadas para a virtualização do acesso ao recurso requisitado (etapa 892). Em algumas modalidades, um driver de filtro de sistema de arquivo ou um

minifiltro ou driver de sistema de arquivo intercepta requisições para acesso aos recursos nativos e determina se o identificador de processo associado à requisição interceptada foi associado a um conjunto de regras. Se assim for, as regras associadas ao identificador de processo armazenado são usadas para a virtualização da requisição para acesso a recursos nativos. Caso não, a requisição para acesso a recursos nativos é passada adiante não modificada. Em outras modalidades, uma biblioteca ligada dinamicamente é carregada no processo recém criado e a biblioteca carrega as regras de isolamento. Ainda em outras modalidades, técnicas de modo de kernel (enganchamento, driver de filtro, minifiltro) e técnicas de modo de usuário são usadas para a interceptação de chamadas para acesso a recursos nativos. Para modalidades nas quais um driver de filtro de sistema de arquivo armazena as regras, a biblioteca pode carregar as regras a partir do driver de filtro de sistema de arquivo.

[00295] Os processos que são "filhos" de processos associados a escopos de isolamento são associados aos escopos de isolamento de seu processo "pai". Em algumas modalidades, isto é realizado por um driver de modo de kernel notificando o driver de filtro de sistema de arquivo que um processo filho é criado. Nestas modalidades, o driver de filtro de sistema de arquivo determina se o identificador de processo do processo pai está associado a um escopo de isolamento. Se assim for, o driver de filtro de sistema de arquivo armazena uma associação entre o identificador de processo para o processo filho recém criado e o escopo de isolamento do processo pai. Em outras modalidades, o driver de filtro de sistema de arquivo pode ser chamado diretamente a partir do sistema, sem o uso de um driver de modo de kernel. Em outras modalidades, nos processos que estão associados a escopos de isolamento, as funções de sistema operacional que criam novos processos são enganchadas ou

interceptadas. Quando uma requisição para criação de um novo processo é recebida a partir de um processo como esse, a associação entre o novo processo filho e o espumas sintáticas do pai é armazenada.

[00296] Em algumas modalidades, um escopo ou subescopo pode ser associado a uma linha individual, ao invés de um processo inteiro, permitindo que um isolamento seja realizado em uma base por linha. Em outras modalidades, um isolamento por linha pode ser usado para Serviços e servidores COM+.

[00297] Em algumas modalidades, os ambientes de isolamento são usados para a provisão de funcionalidade adicional ao cliente de transmissão de aplicativo 552. Em uma destas modalidades, um programa aplicativo é executado em um ambiente de isolamento. Em uma outra destas modalidades, uma pluralidade recuperada de arquivos aplicativos reside no ambiente de isolamento. Ainda em uma outra destas modalidades, mudanças em um registro na máquina local 10 são feitas dentro do ambiente de isolamento.

[00298] Em uma modalidade, o cliente de transmissão de aplicativo 552 inclui um ambiente de isolamento 556. Em algumas modalidades, o cliente de transmissão de aplicativo 552 inclui um driver de filtro de sistema de arquivo 564 que intercepta requisições de aplicativo para arquivos. Em uma destas modalidades, o driver de filtro de sistema de arquivo 564 intercepta uma requisição de aplicativo para abertura de um arquivo existente e determina que o arquivo não reside no ambiente de isolamento 556. Em uma outra destas modalidades, o driver de filtro de sistema de arquivo 564 redireciona a reação química para o serviço de transmissão 554, em resposta a uma determinação que o arquivo não reside no ambiente de isolamento 556. O serviço de transmissão 554 pode extrair o arquivo a partir da pluralidade de arquivos aplicativos e armazenar o arquivo no ambiente de isolamento

556. O driver de filtro de sistema de arquivo 564 pode então responder à requisição para o credencial com a cópia armazenada do credencial. Em algumas modalidades, o driver de filtro de sistema de arquivo 564 pode redirecionar a requisição para o arquivo para um servidor de arquivo 540, em resposta a uma indicação que o serviço de transmissão 554 não recuperou o arquivo ou a pluralidade de arquivos aplicativos e uma determinação que o arquivo não reside no ambiente de isolamento 556. Em algumas modalidades, o serviço de transmissão 554 pode incluir/compreender um programa de aceleração 6120 para a realização de algumas ou de todas as técnicas de aceleração discutidas abaixo para a aceleração do armazenamento ou da entrega de arquivos e aplicativos.

[00299] Em algumas modalidades, o driver de filtro de sistema de arquivo 564 usa uma regra de isolamento estrita para evitar que um conflito ou dados inconsistentes apareçam no ambiente de isolamento 556. Em uma destas modalidades, o driver de filtro de sistema de arquivo 564 interceptando uma requisição por um recurso em um ambiente de isolamento pode redirecionar a requisição para um ambiente de isolamento de aplicativo. Em uma outra destas modalidades, o driver de filtro de sistema de arquivo 564 não redireciona a requisição para um escopo do sistema.

[00300] Em uma modalidade, o serviço de transmissão 554 usa comandos de IOCTL para comunicação com o driver de filtro. Em uma outra modalidade, comunicações para o servidor de arquivo 540 são recebidas pelo protocolo de transmissão Microsoft SMB.

[00301] Em algumas modalidades, o mecanismo de pacote 530 armazena em um arquivo de manifesto uma lista de tipos de arquivo publicados como aplicativos disponíveis e torna esta informação disponível para um software de publicação de aplicativo. Em uma destas modalidades, o mecanismo de pacote 530 recebe esta

informação a partir da monitoração de uma instalação de um programa aplicativo no ambiente de isolamento na máquina de estágio. Em uma outra destas modalidades, um usuário do mecanismo de pacote 530 provê esta informação para o mecanismo de pacote 530. Em outras modalidades, o software de publicação de aplicativo no console de suíte de acesso 520 consulta o arquivo de manifesto para apresentar a um usuário do console de suíte de acesso 520 os tipos possíveis de arquivo que podem estar associados ao aplicativo requisitado sendo publicado. O usuário seleciona um tipo de arquivo para associar a um aplicativo publicado em particular. O tipo de arquivo é apresentado para a máquina local 10 no momento da enumeração de aplicativo.

[00302] A máquina local 10 pode incluir um agente de cliente 560. O agente de cliente 560 provê uma funcionalidade para associação de um tipo de arquivo com um programa aplicativo e seleção de um método de execução do programa aplicativo em resposta à associação. Em uma modalidade, o agente de cliente 560 é um aplicativo de vizinhança de programa.

[00303] Quando um programa aplicativo é selecionado para execução, a máquina local 10 faz uma determinação quanto a um método de execução associado a um tipo de arquivo do programa aplicativo. Em uma modalidade, a máquina local 10 determina que o tipo de arquivo está associado a um método de execução requerendo uma sessão de transmissão de aplicativo para recuperação dos arquivos aplicativos e execução em um ambiente de isolamento. Neste ambiente, a máquina local 10 pode redirecionar a requisição para o cliente de transmissão de aplicativo 552 ao invés de abertura de uma versão local do programa aplicativo. Em uma outra modalidade, o agente de cliente 560 faz a determinação. Ainda em uma outra modalidade, o agente de cliente 560 redireciona a requisição para o cliente de transmissão de aplicativo 552.

[00304] Em uma modalidade, o cliente de transmissão de aplicativo 552 requisita uma informação de acesso associada ao programa aplicativo a partir da máquina remota 30. Em algumas modalidades, o cliente de transmissão de aplicativo 552 recebe um programa executável contendo a informação de acesso. Em uma destas modalidades, o cliente de transmissão de aplicativo 552 recebe um programa executável capaz de exibir na máquina local 10 dados de saída de aplicativo gerados a partir de uma execução do programa aplicativo em uma máquina remota. Em uma outra destas modalidades, o cliente de transmissão de aplicativo 552 recebe um programa executável capaz de recuperação do programa aplicativo através de uma sessão de transmissão de aplicativo e executar o programa aplicativo em um ambiente de isolamento na máquina local 10. Nesta modalidade, o cliente de transmissão de aplicativo 552 pode executar o programa executável recebido. Ainda em uma outra destas modalidades, a máquina remota 30 seleciona um programa executável para a provisão para a máquina local 10 em resposta à realização de uma resolução de aplicativo, conforme descrito acima.

[00305] Com referência, agora, à figura 9, um fluxograma descreve uma modalidade de etapas seguidas em um método para execução de um aplicativo. Conforme descrito acima na figura 7, com referência à etapa 214, uma máquina local 10 recebe e executa a pluralidade de arquivos aplicativos. Em uma visão geral breve, a máquina local 10 recebe um arquivo que inclui uma informação de acesso para acesso a uma pluralidade de arquivos aplicativos e para execução de um primeiro cliente capaz de receber uma transmissão de aplicativo (etapa 902). A máquina local 10 recupera uma identificação da pluralidade de arquivos aplicativos, em resposta ao arquivo (etapa 904). A máquina local 10 recupera pelo menos uma característica requerida para execução da pluralidade de arquivos aplicativos, em

resposta ao arquivo (etapa 906). A máquina local 10 determina se a máquina local 10 inclui pelo menos uma característica (etapa 908). A máquina local 10 executa um segundo cliente, o segundo cliente requisitando uma execução da pluralidade de arquivos aplicativos em uma máquina remota, em resposta a uma determinação que a máquina local 10 carece de pelo menos uma característica (etapa 910).

[00306] Com referência à figura 9, e em maiores detalhes, a máquina local 10 recebe um arquivo que inclui uma informação de acesso para acesso a uma pluralidade de arquivos aplicativos e para execução de um primeiro cliente capaz de receber uma transmissão de aplicativo (etapa 902). Em uma modalidade, a máquina local 10 recebe uma informação de acesso que inclui uma identificação de uma localização de uma pluralidade de arquivos aplicativos compreendendo um programa aplicativo. Em uma outra modalidade, a máquina local 10 recebe o arquivo em resposta a uma requisição de execução do programa aplicativo. Ainda em uma outra modalidade, a informação de acesso inclui uma indicação que a pluralidade de arquivos aplicativos reside em uma máquina remota 30', tal como um servidor de aplicativo ou um servidor de arquivo. Ainda em uma outra modalidade, a informação de acesso indica que a máquina local 10 pode recuperar a pluralidade de arquivos aplicativos a partir da máquina remota 30 por uma sessão de transmissão de aplicativo.

[00307] A máquina local 10 recupera uma identificação da pluralidade de arquivos aplicativos, em resposta ao arquivo (etapa 904). Em uma modalidade, a máquina local 10 identifica uma máquina remota na qual a pluralidade de arquivos aplicativos reside, em resposta ao arquivo incluindo uma informação de acesso. Em uma outra modalidade, a máquina local 10 recupera a partir da máquina remota 30 um arquivo identificando a pluralidade de arquivos

aplicativos. Em algumas modalidades, a pluralidade de arquivos aplicativos compreende um programa aplicativo. Em outras modalidades, a pluralidade de arquivos aplicativos compreende múltiplos programas aplicativos. Ainda em outras modalidades, a pluralidade de arquivos aplicativos compreende múltiplas versões de um único programa aplicativo.

[00308] Com referência à frente à figura 10, um fluxograma descreve uma modalidade de uma pluralidade de arquivos aplicativos residentes em uma máquina remota 30', tal como o servidor de arquivo 540. Na figura 10, uma pluralidade de arquivos aplicativos, referidos como um pacote, inclui arquivos aplicativos compreendendo três versões diferentes de um ou mais programas aplicativos.

[00309] Em uma modalidade, cada subconjunto de arquivos aplicativos compreendendo uma versão de um ou mais programas aplicativos e armazenados no pacote é referido como um alvo. O Alvo 1, por exemplo, inclui uma versão de um programa aplicativo de processamento de texto e de um programa de planilha, a versão compatível com a versão em língua inglesa do sistema operacional Microsoft Windows 2000. O Alvo 2 inclui uma versão de um programa aplicativo de processamento de texto e de um programa de planilha, a versão compatível com a versão em língua inglesa do sistema operacional Microsoft XP. O Alvo 3 é uma versão de um programa aplicativo de processamento de texto e de um programa de planilha, a versão compatível com a versão em língua japonesa do sistema operacional Microsoft Windows 2000 com Service Pack 3.

[00310] Retornando, agora, à figura 9, em algumas modalidades, o arquivo recuperado a partir da máquina remota 30 hospedando a pluralidade de arquivos aplicativos inclui uma descrição do pacote e os alvos incluídos na pluralidade de arquivos aplicativos. Em outras modalidades, o arquivo recuperado a partir da máquina remota 30

identifica a pluralidade de arquivos aplicativos compreendendo um programa aplicativo requisitado para execução pela máquina local 10.

[00311] A máquina local 10 recupera pelo menos uma característica requerida para execução da pluralidade de arquivos aplicativos, em resposta ao arquivo (etapa 906). Em algumas modalidades, a máquina local 10 pode não executar um programa aplicativo a menos que a máquina local inclua certas características. Em uma destas modalidades, programas aplicativos diferentes requerem que as máquinas locais 10 incluam características diferentes das características requeridas por outros programas aplicativos. Em uma outra destas modalidades, a máquina local 10 recebe uma identificação de pelo menos uma característica requerida para execução da pluralidade de arquivos aplicativos compreendendo o programa aplicativo requisitado pela máquina local 10.

[00312] A máquina local determina se a máquina local 10 inclui pelo menos uma característica (etapa 908). Em uma modalidade, a máquina local 10 avalia um sistema operacional da máquina local 10 para determinar se a máquina local 10 inclui pelo menos uma característica. Em uma outra modalidade, a máquina local 10 identifica uma linguagem usada por um sistema operacional na máquina local 10 para determinar se a máquina local 10 inclui pelo menos uma característica. Ainda em uma outra modalidade, a máquina local 10 identifica um nível de revisão de um sistema operacional na máquina local 10 para determinar se a máquina local 10 inclui pelo menos uma característica. Em ainda uma outra modalidade, a máquina local 10 identifica uma versão de aplicativo de um programa aplicativo residente na máquina local 10, para determinar se a máquina local 10 inclui pelo menos uma característica. Em algumas modalidades, a máquina local 10 determina se a máquina local 10 inclui um driver de dispositivo para determinar se a máquina local 10 inclui pelo menos

uma característica. Em outras modalidades, a máquina local 10 determina se a máquina local 10 inclui um sistema operacional para determinar se a máquina local 10 inclui pelo menos uma característica. Ainda em outras modalidades, a máquina local 10 determina se a máquina local 10 inclui uma licença para execução da pluralidade de arquivos aplicativos para determinar se a máquina local 10 inclui pelo menos uma característica.

[00313] A máquina local 10 executa um segundo cliente, o segundo cliente requisitando a execução da pluralidade de arquivos aplicativos em uma máquina remota 30, em resposta a uma determinação que a máquina local 10 carece de pelo menos uma característica (etapa 910). Em uma modalidade, quando a máquina local 10 determina que a máquina local 10 carece de pelo menos uma característica, a máquina local 10 não executa o primeiro cliente capaz de receber uma transmissão de aplicativo. Em uma outra modalidade, uma política proíbe a máquina local 10 de receber a pluralidade de arquivos aplicativos por uma transmissão de aplicativo, quando a máquina local 10 carecer de pelo menos uma característica. Em algumas modalidades, a máquina local 10 determina que a máquina local 10 não inclui pelo menos uma característica. Em uma destas modalidades, a máquina local 10 executa o primeiro cliente, o primeiro cliente recebendo uma transmissão de aplicativo compreendendo a pluralidade de arquivos aplicativos a partir de uma máquina remota 30 para execução na máquina local.

[00314] Em algumas modalidades, a máquina local 10 executa o segundo cliente requisitando uma execução da pluralidade de arquivos aplicativos em uma máquina remota quando determina que a máquina local 10 carece da pelo menos uma característica. Em uma destas modalidades, o segundo cliente transmite a requisição para uma máquina remota 30 hospedando a pluralidade de arquivos aplicativos.

Em uma outra destas modalidades, a máquina remota 30 executa a pluralidade de arquivos aplicativos compreendendo o programa aplicativo e gera dados de saída de aplicativo. Ainda em uma outra destas modalidades, o segundo cliente recebe dados de saída de aplicativo gerados pela execução da pluralidade de arquivos aplicativos na máquina remota. Em algumas modalidades, o segundo cliente recebe os dados de saída de aplicativo através de um protocolo de nível de apresentação de Arquitetura de Computação Independente ou um protocolo de nível de apresentação de Windows de Área de Trabalho Remota ou um protocolo de nível de apresentação de X-Windows. Ainda em uma outra destas modalidades, o segundo cliente exibe a saída de aplicativo na máquina local 10.

[00315] Em algumas modalidades, o segundo cliente transmite a requisição para uma máquina remota 30 que não hospeda a pluralidade de arquivos aplicativos. Em uma destas modalidades, a máquina remota 30 pode requisitar a pluralidade de arquivos aplicativos a partir de uma segunda máquina remota 30 hospedando a pluralidade de arquivos aplicativos. Em uma outra destas modalidades, a máquina remota 30 pode receber a pluralidade de arquivos aplicativos a partir da segunda máquina remota 30 através de uma sessão de transmissão de aplicativo. Ainda em uma outra destas modalidades, a máquina remota 30 armazena a pluralidade recebida de arquivos aplicativos em um ambiente de isolamento e executa o programa aplicativo no ambiente de isolamento. Ainda em uma outra destas modalidades, a máquina remota transmite os dados de saída de aplicativo gerados para o segundo cliente na máquina local.

[00316] Com referência de volta à figura 5, em uma modalidade, o primeiro cliente, capaz de receber a transmissão de aplicativo, é um cliente de transmissão de aplicativo 552. O cliente de transmissão de aplicativo 552 recebe o arquivo, recupera uma identificação de uma

pluralidade de arquivos aplicativos e pelo menos uma característica requerida para execução da pluralidade de arquivos aplicativos, em resposta ao arquivo e determina se a máquina local 10 inclui pelo menos uma característica. Em uma outra modalidade, o segundo cliente é um agente de cliente 560. Em algumas modalidades, o agente de cliente 560 recebe o arquivo a partir do cliente de transmissão de aplicativo 552 em resposta a uma determinação, pelo cliente de transmissão de aplicativo 552 que a máquina local 10 carece de pelo menos uma característica.

[00317] Em algumas modalidades, o aplicativo 566 em execução na máquina local 10 enumera os arquivos associados ao aplicativo 566 usando as chamadas Win32 FindFirstFile() e FindNextFile() API. Em uma destas modalidades, uma pluralidade de arquivos aplicativos compreende o aplicativo 566. Em uma outra destas modalidades, nem todos os arquivos na pluralidade de arquivos aplicativos residem na máquina local 10. Ainda em uma outra destas modalidades, o serviço de transmissão 554 recuperou a pluralidade de arquivos aplicativos em um arquivo de armazenamento de backup, mas extraiu apenas um subconjunto da pluralidade de arquivos aplicativos. Ainda em uma outra destas modalidades, o serviço de transmissão 554 e o driver de filtro de sistema de arquivo 564 provêem uma funcionalidade para satisfação da requisição enumerada, mesmo quando o arquivo requisitado não residir na máquina local 10.

[00318] Em uma modalidade, a funcionalidade é provida pela interceptação das requisições de enumeração e provisão dos dados como se todos os arquivos na pluralidade de arquivos aplicativos residissem na máquina local 10. Em uma outra modalidade, a funcionalidade é provida pela interceptação, pelo driver de filtro de sistema de arquivo 564, de uma requisição de enumeração transmitida como um comando de IOCTL, tal como

IRP_MJ_DIRECTORY_CONTROL IOCTL. Quando o 564 Quando o driver de filtro de sistema de arquivo 564 intercepta a chamada, o driver de filtro de sistema de arquivo 564 redireciona a requisição para o serviço de transmissão 554. Em uma modalidade, o driver de filtro de sistema de arquivo 564 determina que a enumeração requisitada reside em um ambiente de isolamento na máquina local 10, antes do redirecionamento da requisição para o serviço de transmissão 554. Em uma outra modalidade, o serviço de transmissão 554 cumpre a requisição usando um arquivo na pluralidade de arquivos aplicativos, o arquivo incluindo uma enumeração de uma estrutura de diretório associada à pluralidade de arquivos aplicativos. Ainda em uma outra modalidade, o serviço de transmissão 554 provê a resposta para a requisição para o driver de filtro de sistema de arquivo 564 para satisfação da requisição de enumeração.

[00319] Com referência, agora, à figura 11, um fluxograma descreve uma modalidade das etapas seguidas em um método para se responder localmente a requisições por metadados de arquivo associados a arquivos armazenados remotamente. Em uma visão geral breve, (i) uma estrutura de diretório representando um programa aplicativo armazenado pela máquina remota e (ii) metadados associados a cada arquivo compreendendo o programa aplicativo armazenado são recebidos a partir de uma máquina remota (etapa 1102). A estrutura de diretório e os metadados são armazenados (etapa 1104). Pelo menos uma requisição para acesso a metadados associados a um arquivo específico na estrutura de diretório é recebida (etapa 1106). Pelo menos uma requisição é respondida usando-se os metadados armazenados (etapa 1108).

[00320] Com referência à figura 11 em maiores detalhes, uma estrutura de diretório representando um programa aplicativo armazenado pela máquina remota e metadados associados a cada

arquivo compreendendo o programa aplicativo armazenado são recebidos a partir de uma máquina remota (etapa 1102). Em uma modalidade, o serviço de transmissão 554 recebe a estrutura de diretório e os metadados. Em uma outra modalidade, o serviço de transmissão 554 recebe a estrutura de diretório e os metadados quando o serviço de transmissão 554 recupera uma pluralidade de arquivos aplicativos compreendendo o programa aplicativo armazenado. Ainda em uma outra modalidade, a estrutura de diretório e os metadados são armazenados em um arquivo na pluralidade de arquivos aplicativos.

[00321] Em uma modalidade, os metadados associados a cada arquivo compreendem um nome alternativo para pelo menos um arquivo. Em uma outra modalidade, os metadados associados a cada arquivo incluem um nome resumido para pelo menos um arquivo, o nome tendo um comprimento de oito caracteres, um ponto e uma extensão de três caracteres. Ainda em uma outra modalidade, os metadados associados a cada arquivo incluem um mapeamento entre o nome alternativo para pelo menos um arquivo e o nome resumido para pelo menos um arquivo. Em algumas modalidades, um arquivo na pluralidade de arquivos aplicativos tem um nome de arquivo alternativo. Em uma destas modalidades, quando o arquivo é recuperado por um serviço de transmissão 554 para uma máquina local, o arquivo é associado a um nome resumido, em resposta ao mapeamento entre o nome alternativo para o arquivo e o nome resumido para pelo menos um arquivo.

[00322] A estrutura de diretório e os metadados são armazenados (etapa 1104). Em uma modalidade, a estrutura de diretório e os metadados são armazenados em um ambiente de isolamento 556. Em uma outra modalidade, a estrutura de diretório e os metadados são armazenados em um elemento de memória cache. Ainda em uma

outra modalidade, a estrutura de diretório representando um programa aplicativo armazenado pela máquina remota é usada para a geração de uma enumeração de uma estrutura de diretório representando um programa aplicativo em execução na máquina local.

[00323] Pelo menos uma requisição para acesso a metadados associados a um arquivo específico na estrutura de diretório é recebida (etapa 1106). Em uma modalidade, a requisição é uma requisição para enumeração do arquivo. Em uma outra modalidade, a requisição é uma requisição para se_determinar se uma cópia do arquivo compreendendo o programa aplicativo armazenado reside localmente.

[00324] Em uma modalidade, a requisição é feita por um aplicativo 566 em execução em um ambiente de isolamento em uma máquina local. Em uma outra modalidade, a requisição é feita pelo cliente de transmissão de aplicativo 552. Ainda em uma outra modalidade, a requisição é feita em nome do aplicativo 566.

[00325] Em uma modalidade, a requisição é interceptada por um driver de filtro de sistema de arquivo 564. Em uma outra modalidade, a requisição é encaminhada para o cliente de transmissão de aplicativo 552 pelo driver de filtro de sistema de arquivo 564. Ainda em uma outra modalidade, a requisição é encaminhada para o serviço de transmissão 554 pelo driver de filtro de sistema de arquivo 564.

[00326] Em algumas modalidades, a requisição é enganchada por uma função que substitui a função ou as funções de sistema operacional para enumeração de um diretório. Em uma outra modalidade, uma biblioteca ligada dinamicamente de enganchamento é usada para a interceptação da requisição. A função de enganchamento pode ser executada em um modo de usuário ou em um modo de kernel. Para modalidades nas quais a função de enganchamento se executa no modo de usuário, a função de

enganchamento pode ser carregada no espaço de endereço de um processo quando o processo for criado. Para modalidades nas quais a função de enganchamento se executa no modo de kernel, a função de enganchamento pode ser associada a um recurso de sistema operacional que é usado no despacho de requisições para operações de arquivo. Para modalidades nas quais uma função em separado de sistema operacional é provida para cada tipo de operação de arquivo, cada função pode ser enganchada separadamente. De foram alternativa, uma função de enganchamento única pode ser provida, a qual intercepta chamadas de criação ou abertura para vários tipos de operações de arquivo.

[00327] Pelo menos uma requisição é respondida usando-se os metadados armazenados (etapa 1108). Em uma modalidade, o driver de filtro de sistema de arquivo 564 responde à requisição. Em uma outra modalidade, o cliente de transmissão de aplicativo 552 responde à requisição. Ainda em uma outra modalidade, o serviço de transmissão 554 responde à requisição. Em uma modalidade, os metadados armazenados são acessados para responderem a pelo menos uma requisição. Em uma outra modalidade, a requisição é respondida com uma indicação falsa que uma cópia remota do arquivo reside localmente.

[00328] Em uma modalidade, uma operação de FindFirst de Sistema Operacional Windows é satisfeita em resposta aos metadados recebidos. Em uma outra modalidade, uma operação FindNext de Sistema Operacional Windows é satisfeita em resposta aos metadados recebidos. Ainda em uma outra modalidade, uma operação para identificação de um nó de raiz em uma estrutura de diretório é satisfeita em resposta aos metadados recebidos. Em algumas modalidades, uma API de camada de aplicativo, tal como WIN32_FIND_DATA API é usada para responder à operação. Em

outras modalidades, uma API de camada de kernel tal como FILE_BOTH_DIR_INFORMATION é usada para responder à operação.

[00329] Em uma modalidade, os metadados satisfazem a uma operação para a identificação de um horário de acesso associado a um nó em uma estrutura de diretório. Em uma outra modalidade, os metadados satisfazem a uma operação para a identificação de um horário de modificação associado a um nó em uma estrutura de diretório. Ainda em uma outra modalidade, os metadados satisfazem a uma operação para a identificação de um nó modificado em uma estrutura de diretório.

[00330] Com referência, agora, à figura 12, um diagrama de blocos descreve uma modalidade de um sistema para se responder localmente a requisições para metadados de arquivo associados a arquivos armazenados remotamente, incluindo um serviço de transmissão 554, um driver de filtro de sistema de arquivo 564, uma estrutura de diretório 570, uma pluralidade de arquivos aplicativos 572, metadados 574 e um elemento de memória cache 576. Em uma breve visão geral, a estrutura de diretório 570 identifica uma pluralidade de arquivos associados a pelo menos um programa aplicativo. Os metadados 574 são associados a pelo menos um da pluralidade de arquivos, pelo menos um da pluralidade de arquivos residindo em uma máquina remota. Em uma modalidade, a estrutura de diretório 570 inclui os metadados 574. O elemento de memória cache 576 armazena a estrutura de diretório 570. O driver de filtro de sistema de arquivo 564 intercepta uma requisição para acesso a metadados associados a pelo menos um arquivo armazenado remotamente, acessa o elemento de memória cache e responde a pelo menos uma requisição usando a estrutura de diretório armazenada.

[00331] Em algumas modalidades, o serviço de transmissão 554

recebe a estrutura de diretório 570 e os metadados 574. Em uma destas modalidades, a estrutura de diretório 570 representa uma pluralidade de arquivos aplicativos 572 associada a um programa aplicativo, a pluralidade de arquivos aplicativos 572 residindo em uma máquina remota, tal como a máquina remota 30. Em uma outra destas modalidades, os metadados 574 compreendem uma informação para se responder a uma requisição de FindFirst de Sistema Operacional Windows. Ainda em uma outra destas modalidades, os metadados 574 compreendem uma informação para se responder a uma requisição de FindNext de Sistema Operacional Windows. Ainda em uma outra destas modalidades, os metadados 574 compreendem uma informação para se responder a uma requisição para a identificação de um nó de raiz em uma estrutura de diretório. Em uma outra destas modalidades, os metadados 574 compreendem uma informação para se responder a uma requisição para a identificação de um nó em uma estrutura de diretório. Em algumas modalidades, uma API de camada de aplicativo, tal como WIN32_FIND_DATA API é usada para responder à operação. Em outras modalidades, uma API de camada de kernel tal como FILE_BOTH_DIR_INFORMATION é usada para se responder à operação.

[00332] Em algumas modalidades, pequenas quantidades de metadados 574 sobre um arquivo podem ser armazenadas diretamente no nome de arquivo literal, tal como pela sufixação do nome virtual com um indicador de metadados, onde um indicador de metadados é uma string unicamente associada a um estado de metadados em particular. O indicador de metadados pode indicar ou codificar um ou vários bits de metadados. As requisições para acesso ao arquivo por uma verificação de nome de arquivo virtual quanto a possíveis variações do nome de arquivo literal devido à presença de um indicador de metadados, e requisições para recuperação do nome

de arquivo em si são enganchadas ou interceptadas de modo a se responder com o nome literal. Em outras modalidades, um ou mais nomes alternativos para o arquivo podem ser formados a partir do nome de arquivo virtual e de um indicador de metadados, e podem ser criados usando-se instalações de enlace rígido ou de enlace flexível providas pelo sistema de arquivo. A existência destes enlaces pode ser escondida dos aplicativos pelo ambiente de isolamento pela indicação que o arquivo não é encontrado, se uma requisição for dada para acessar um arquivo usando o nome de um enlace. Uma presença ou ausência de um enlace em particular pode indicar um bit de metadados para cada indicador de metadados, ou pode haver um enlace com um indicador de metadados que pode assumir múltiplos estados para indicar vários bits de metadados. Ainda em outras modalidades, quando o sistema de arquivo suporta transmissões contínuas de arquivo alternativas, uma transmissão de arquivo alternativa pode ser criada para a concretização de metadados, com o tamanho da transmissão indicando vários bits de metadados. Ainda em outras modalidades, um sistema de arquivo pode prover diretamente a capacidade de armazenar alguns metadados de terceiros para cada arquivo no sistema de arquivo. Ainda em uma outra modalidade, um subescopo em separado pode ser usado para a gravação de arquivos apagados, e a existência de um arquivo (não marcado com um símbolo marcador de posição) naquele subescopo é tomada como significando que o arquivo é apagado.

[00333] Em uma modalidade, os dados em um ambiente de isolamento de usuário, um ambiente de isolamento de aplicativo e um escopo do sistema são combinados para a formação de uma enumeração local de uma estrutura de diretório representando um aplicativo. Em uma outra modalidade, o serviço de transmissão 554 acessa metadados 574 e a estrutura de diretório 570 para preencher o

ambiente de isolamento de aplicativo. Ainda em uma outra modalidade, o driver de filtro de sistema de arquivo 564 gera a enumeração local da estrutura de diretório. Em ainda uma outra modalidade, a enumeração local da estrutura de diretório identifica pelo menos um arquivo na pluralidade de arquivos aplicativos 572, pelo menos um arquivo residindo em uma máquina remota e não na máquina local. Em algumas modalidades, a enumeração local da estrutura de diretório é armazenada no elemento de memória cache 576. Em outras modalidades, o serviço de transmissão 554 gera o ambiente de isolamento de aplicativo e a enumeração local da estrutura de diretório.

[00334] Em uma modalidade, o driver de filtro de sistema de arquivo 564 intercepta uma requisição transmitida para um escopo do sistema para acesso à enumeração local da estrutura de diretório. Em uma outra modalidade, o driver de filtro de sistema de arquivo 564 gera a enumeração local após a interceptação da requisição. Ainda em uma outra modalidade, o driver de filtro de sistema de arquivo 564 redireciona a requisição para a enumeração local para o ambiente de isolamento de usuário. Ainda em uma outra modalidade, o driver de filtro de sistema de arquivo 564 redireciona a requisição para a enumeração local para o ambiente de isolamento de aplicativo.

[00335] Em algumas modalidades, o driver de filtro de sistema de arquivo 564 intercepta uma requisição para acesso a um arquivo, identifica a enumeração local no diretório, o arquivo residente em uma máquina remota. Em uma destas modalidades, o driver de filtro de sistema de arquivo 564 requisita uma recuperação do arquivo pelo serviço de transmissão 554, conforme descrito em maiores detalhes em relação à figura 13 abaixo.

[00336] Conforme os aplicativos rodando em um ambiente de isolamento fazem requisições para arquivos, um driver de filtro

intercepta estas requisições. Se a requisição for para abertura de um arquivo, o driver de filtro primeiramente redirecionará a requisição para um ambiente de isolamento, para determinar se a requisição pode ser satisfeita pelo ambiente de isolamento. Se a chamada for bem-sucedida, o driver de filtro responderá à requisição com a instância do arquivo localizado no ambiente de isolamento.

[00337] Contudo, se o arquivo requisitado não residir no ambiente de isolamento, o driver de filtro enviará uma requisição para o serviço de transmissão 554 para recuperação do arquivo a partir da pluralidade de arquivos aplicativos, bloqueará até a requisição estar completa e, então, tentará de novo a abertura original. Em algumas modalidades, a funcionalidade do serviço de transmissão 554 para recuperação de arquivos a partir da pluralidade de arquivos aplicativos mediante o recebimento de uma requisição do driver de filtro é referida como "armazenamento em cache sob demanda".

[00338] Com referência, agora, à figura 13, um fluxograma descreve uma modalidade das etapas seguidas em um método para acesso a um arquivo remoto em uma estrutura de diretório associada a um programa aplicativo em execução localmente. Em uma visão geral breve, uma requisição por um aplicativo para acesso a um arquivo é interceptada (etapa 1302). A requisição é redirecionada para um primeiro ambiente de isolamento (etapa 1304). Uma determinação é feita que o arquivo requisitado não existe no primeiro ambiente de isolamento (etapa 1306). A requisição é redirecionada para um segundo ambiente de isolamento, em resposta a uma determinação que o arquivo está identificado em uma enumeração de uma estrutura de diretório associada a uma pluralidade de arquivos aplicativos residentes em uma máquina remota (etapa 1308). O arquivo requisitado é recuperado a partir da máquina remota, em resposta a uma determinação que o segundo ambiente de isolamento não contém

o arquivo e que o arquivo está identificado na enumeração (etapa 1310).

[00339] Com referência à figura 13, e em maiores detalhes, uma requisição por um aplicativo para acesso a um arquivo é interceptada (etapa 1302). Em uma modalidade, a requisição é interceptada por um driver de filtro de sistema de arquivo. Em uma outra modalidade, o driver de filtro de sistema de arquivo intercepta todas as requisições para acesso a arquivos. Ainda em uma outra modalidade, um cliente de transmissão de aplicativo 552 intercepta a requisição. Em algumas modalidades, uma requisição por um aplicativo para acesso a um arquivo executável é interceptada. Em outras modalidades, uma requisição por um aplicativo para acesso a um arquivo, uma porção do aplicativo em execução em uma máquina local 10, é interceptada.

[00340] A requisição é redirecionada para um primeiro ambiente de isolamento (etapa 1304). Em uma modalidade, o aplicativo é executado no primeiro ambiente de isolamento. Em uma modalidade, o aplicativo é um programa aplicativo, tal como um programa de processamento de texto ou um programa de planilha. Em uma outra modalidade, o aplicativo é o cliente de transmissão de aplicativo 552. Ainda em uma outra modalidade, o aplicativo é um componente dentro do cliente de transmissão de aplicativo 552 tentando abrir um programa aplicativo em nome de um usuário da máquina local 10. Em uma outra modalidade, o driver de filtro de sistema de arquivo redireciona a requisição para o primeiro ambiente de isolamento.

[00341] Uma determinação é feita que o arquivo requisitado não existe no primeiro ambiente de isolamento (etapa 1306). Em uma modalidade, o driver de filtro de sistema de arquivo recebe uma indicação que o arquivo requisitado não existe no primeiro ambiente de isolamento.

[00342] A requisição é redirecionada para um segundo ambiente de

isolamento em resposta a uma determinação que o arquivo está identificado em uma enumeração de uma estrutura de diretório associada a uma pluralidade de arquivos aplicativos residindo em uma máquina remota (etapa 1308). Em uma modalidade, a enumeração da estrutura de diretório é recebida com uma informação de acesso referente a uma execução do primeiro aplicativo. Em uma outra modalidade, a enumeração identifica uma pluralidade de arquivos aplicativos compreendendo um segundo aplicativo. Nesta modalidade, o primeiro aplicativo é uma cópia local do segundo aplicativo.

[00343] O arquivo requisitado é recuperado a partir da máquina remota, em resposta a uma determinação que o segundo ambiente de isolamento não contém o arquivo e que o arquivo está identificado na enumeração (etapa 1310). Em uma modalidade, o arquivo requisitado é recuperado a partir de uma segunda máquina remota. Em uma outra modalidade, o arquivo requisitado é recuperado a partir de um servidor de arquivo. Em algumas modalidades, a enumeração da estrutura de diretório identifica uma pluralidade de arquivos aplicativos residindo na máquina local. Em outras modalidades, a enumeração da estrutura de diretório indica que a pluralidade de arquivos aplicativos reside na máquina local. Em uma destas modalidades, quando o aplicativo requisita acesso ao arquivo na pluralidade de arquivos aplicativos, a qual a enumeração da estrutura de diretório indicou que reside na máquina local, o arquivo é adquirido a partir do servidor de arquivo mediante uma interceptação da requisição de acesso. Em uma outra destas modalidades, o servidor de arquivo transmite o arquivo requisitado para a máquina local. Ainda em uma outra destas modalidades, quando o aplicativo requisita acesso ao arquivo na pluralidade de arquivos aplicativos a qual a enumeração da estrutura de diretório indicou residir na máquina local, uma cópia do arquivo é provida para o aplicativo a partir de um cache local.

[00344] Em algumas modalidades, o arquivo requisitado é encriptado. Em outras modalidades, o arquivo requisitado é armazenado em uma forma encriptada. Ainda em outras modalidades, o aplicativo requisitando o arquivo pode ser impedido de descriptar o arquivo requisitado, se o aplicativo carecer de autorização para acessar o arquivo requisitado.

[00345] Em uma modalidade, uma determinação é feita que a enumeração da estrutura de diretório não identifica o arquivo. Nesta modalidade, a requisição para acessar o arquivo pode ser redirecionada para um ambiente fora do primeiro ambiente de isolamento e fora do segundo ambiente de isolamento.

[00346] Em algumas modalidades, uma segunda requisição para acesso ao arquivo é interceptada. Em uma destas modalidades, a requisição para acesso ao arquivo é feita por um segundo aplicativo. Em uma outra destas modalidades, o segundo aplicativo se executa em um terceiro ambiente de isolamento. Ainda em uma outra destas modalidades, a requisição é redirecionada para o segundo ambiente de isolamento, em resposta a uma determinação que o arquivo está enumerado na enumeração e que o segundo ambiente de isolamento realmente contém o arquivo. A determinação é feita que a máquina local armazenou o arquivo no segundo ambiente de isolamento mediante o recebimento do arquivo a partir do servidor de arquivo. Ainda em uma outra modalidade, o arquivo é armazenado no terceiro ambiente de isolamento.

[00347] Com referência, agora, à figura 14, um diagrama de blocos descreve uma modalidade de um sistema para acesso a um arquivo em uma estrutura de diretório associada a um aplicativo. Em uma visão geral breve, uma máquina local 10 inclui um cliente de transmissão de aplicativo 552, um serviço de transmissão 554, um ambiente de isolamento 556, um driver de filtro de sistema de arquivo

564, e um primeiro aplicativo 566. A máquina local 10 pode interagir com um servidor de arquivo 540, uma máquina remota 30, uma interface da web 558 e um segundo aplicativo 566'.

[00348] A máquina local 10 inicializa o cliente de transmissão de aplicativo 552 para execução do primeiro aplicativo 566. Em uma modalidade, o cliente de transmissão de aplicativo 552 inicializa um serviço de transmissão 554 para recuperação e execução do primeiro aplicativo 566. Em algumas modalidades, uma pluralidade de arquivos aplicativos compreende o primeiro aplicativo 566. Em uma destas modalidades, o serviço de transmissão 554 recupera a pluralidade de arquivos aplicativos e os armazena no ambiente de isolamento 566. Em uma outra destas modalidades, o serviço de transmissão 554 identifica uma localização de uma máquina remota na qual a pluralidade de arquivos aplicativos reside, mas não recupera a pluralidade de arquivos aplicativos. Ainda em uma outra destas modalidades, o serviço de transmissão 554 recupera um subconjunto dos arquivos na pluralidade de arquivos aplicativos. Ainda em uma outra destas modalidades, o serviço de transmissão 554 recupera um arquivo de armazenamento de backup contendo a pluralidade de arquivos aplicativos.

[00349] Em uma modalidade, o primeiro aplicativo 566 compreende uma cópia local de um segundo aplicativo 566' residente em uma máquina remota 30. Em uma outra modalidade, a pluralidade de arquivos aplicativos reside na máquina remota 30 e compreende o segundo aplicativo 566' residindo em uma máquina remota 30. Ainda em uma outra modalidade, para a execução do segundo aplicativo 566', a máquina local 10 recupera a pluralidade de arquivos aplicativos, criando o primeiro aplicativo 566 na máquina local, e executa o primeiro aplicativo 566. Em algumas modalidades, os aplicativos 566 e 566' são aplicativos de usuário, tais como aplicativos

de processamento de texto ou aplicativos de planilha ou aplicativos de apresentação.

[00350] Em algumas modalidades, a pluralidade de arquivos aplicativos inclui um arquivo identificando uma estrutura de diretório associada à pluralidade de arquivos aplicativos na máquina remota 30. Em uma destas modalidades, o arquivo inclui metadados sobre cada arquivo aplicativo na pluralidade de arquivos aplicativos. Em uma outra destas modalidades, o serviço de transmissão 554 recupera os metadados a partir do arquivo para a geração de uma enumeração da estrutura de diretório associada à pluralidade de arquivos aplicativos, conforme descrito em relação à figura 12 acima. Ainda em uma outra destas modalidades, o serviço de transmissão 554 armazena a enumeração da estrutura de diretório associada à pluralidade de arquivos aplicativos compreendendo o segundo aplicativo 566'. Em algumas modalidades, o serviço de transmissão 554 armazena a enumeração em um segundo ambiente de isolamento.

[00351] Em uma modalidade, o serviço de transmissão 554 recupera um arquivo executável inicial associado ao primeiro aplicativo 566. Em uma outra modalidade, o serviço de transmissão 554 executa o primeiro aplicativo 566 na máquina local 10 mediante a recuperação do arquivo executável inicial. Ainda em uma outra modalidade, o primeiro aplicativo 566 requisita acesso a outros arquivos na pluralidade de arquivos aplicativos, já que os arquivos são necessários para uma execução continuada do primeiro aplicativo 566. Em algumas modalidades, o primeiro aplicativo 566 se executa no ambiente de isolamento 556.

[00352] O driver de filtro de sistema de arquivo 564 intercepta requisições pelo primeiro aplicativo 566 em execução no ambiente de isolamento 556 para acesso a um arquivo na pluralidade de arquivos aplicativos. O driver de filtro de sistema de arquivo 564 redireciona a

requisição para o ambiente de isolamento 556. Se o arquivo requisitado residir no ambiente de isolamento 556, o acesso ao arquivo requisitado será provido para o primeiro aplicativo 566.

[00353] Se o arquivo requisitado não residir no ambiente de isolamento 556, o driver de filtro de sistema de arquivo 564 redireciona a requisição para um segundo ambiente de isolamento. Em uma modalidade, o segundo ambiente de isolamento inclui a enumeração da estrutura de diretório gerada pelo serviço de transmissão 554 e associada à pluralidade de arquivos aplicativos compreendendo o segundo aplicativo 566'. Em uma outra modalidade, uma determinação é feita que o arquivo requisitado é identificado na enumeração da estrutura de diretório.

[00354] Em algumas modalidades, o serviço de transmissão 554 provê um semáforo para o ambiente de isolamento 556. Em uma destas modalidades, o driver de filtro de sistema de arquivo 564 usando o semáforo indica para o serviço de transmissão 554 que um acesso a um arquivo na pluralidade de arquivos aplicativos é requerido. Em outras modalidades, o driver de filtro de sistema de arquivo 564 usa uma linha para indicar para o serviço de transmissão 554 que o acesso ao arquivo é requerido.

[00355] Mediante o recebimento da notificação do driver de filtro de sistema de arquivo 564, o serviço de transmissão 554 recupera o arquivo requisitado a partir da pluralidade de arquivos aplicativos. Ainda em uma outra destas modalidades, o serviço de transmissão 554 armazena o arquivo requisitado no segundo ambiente de isolamento de aplicativo. Em uma modalidade, a requisição para acesso ao arquivo é satisfeita com a instância do arquivo recuperado a partir da pluralidade de arquivos aplicativos e armazenado no segundo ambiente de isolamento. Em uma outra modalidade, o arquivo requisitado também é armazenado no primeiro ambiente de

isolamento.

[00356] Em algumas modalidades, uma determinação é feita que o segundo ambiente de isolamento não contém o arquivo e que o arquivo está identificado na enumeração. Em uma destas modalidades, o arquivo é identificado na enumeração na estrutura de diretório associada à pluralidade de arquivos aplicativos compreendendo o segundo aplicativo 566' e o arquivo é um arquivo na pluralidade de arquivos aplicativos. Em uma outra destas modalidades, o serviço de transmissão 554 não recuperou o arquivo a partir da máquina remota. Ainda em uma outra destas modalidades, o serviço de transmissão 554 não recuperou uma pluralidade de arquivos aplicativos incluindo o arquivo requisitado. Ainda em uma outra destas modalidades, o serviço de transmissão 554 recuperou a pluralidade de arquivos aplicativos em um arquivo de armazenamento de backup, mas não recuperou o arquivo requisitado a partir do arquivo de armazenamento de backup.

[00357] Em uma modalidade, o serviço de transmissão 554 inclui um transceptor, em comunicação com o driver de filtro de sistema de arquivo. Em uma outra modalidade, o transceptor recebe a requisição redirecionada a partir do driver de filtro de sistema de arquivo. Ainda em uma outra modalidade, o transceptor encaminha a requisição para o arquivo para uma máquina remota hospedando o arquivo requisitado. Em uma modalidade, a máquina remota é um servidor de arquivo 540. Em uma outra modalidade, a requisição é encaminhada para uma máquina remota 30, a qual roteia a requisição para um servidor de arquivo 540. Em algumas modalidades, o servidor de arquivo 540 transmite o arquivo requisitado para o transceptor na máquina local 10. Em outras modalidades, a máquina remota 30 transmite o arquivo requisitado para o transceptor na máquina local 10. Ainda em outras modalidades, mediante o recebimento do arquivo

requisitado a partir do servidor de arquivo 540, o transceptor armazena o arquivo recebido no segundo ambiente de isolamento.

[00358] Em uma modalidade, o driver de filtro de sistema de arquivo 564 intercepta uma segunda requisição para acesso ao arquivo feita por um terceiro aplicativo 566” em execução na máquina local 10, em um terceiro ambiente de isolamento. Em uma outra modalidade, o driver de filtro de sistema de arquivo 564 redireciona a requisição para acesso ao arquivo para o segundo ambiente de isolamento. Ainda em uma outra modalidade, o driver de filtro de sistema de arquivo 564 determina que o serviço de transmissão 554 armazenou o arquivo recebido no segundo ambiente de isolamento, antes da interceptação da requisição para acesso pelo terceiro aplicativo 566”.

[00359] Em algumas modalidades, mediante uma inicialização, o serviço de transmissão 554 pode preencher um cache em um ambiente de isolamento, antes da execução de um programa aplicativo. Em uma destas modalidades, o serviço de transmissão 554 instala um arquivo de registro no ambiente de isolamento. Em uma outra destas modalidades, o serviço de transmissão 554 armazena um mapeamento entre um nome longo de um arquivo e um nome de arquivo curto.

[00360] Em uma modalidade, para se poupar espaço na máquina local, o tamanho do cache pode ser limitado. Em algumas modalidades, quando o cache se aproxima de seu tamanho limite, os arquivos mais antigos no cache automaticamente serão purgados para se criar espaço para novos arquivos. Em uma destas modalidades, a idade de um arquivo é determinada por uma estampa de tempo mantida pelo sistema operacional indicando um tempo de estampa de tempo de ‘último acesso’. Além da idade de um arquivo, o tipo de arquivo pode ser levado em consideração – arquivos executáveis binários (.EXE, .DLL, etc.) podem ser mantidos por mais tempo do que

artigos de idade similar de outros tipos.

[00361] Mediante uma inicialização, o serviço de transmissão 554 pode enumerar arquivos atualmente em um cache, e determinar o tamanho total do cache. Após um arquivo ser adicionado ao cache, por um ambiente de isolamento 556 ou pelo serviço de transmissão 554, o serviço de transmissão 554 chama uma função para informar o sistema de cache quanto ao novo arquivo, sua localização e seu tamanho. O tamanho de cada arquivo recém armazenado em cache é adicionado ao total rodando do tamanho atual de cache. Este novo total então é comparado em relação ao limite de tamanho de cache e, se o limite tiver sido excedido, o código dispara uma linha para envelhecer o cache. Pode haver apenas uma instância desta linha rodando em qualquer dado tempo.

[00362] A linha gera uma lista de todos os arquivos atualmente no cache, classifica esta lista por estampa de tempo de último acesso e, então, começa a caminhar lista abaixo apagando arquivos até termos liberado espaço em disco suficiente para satisfazer ao critério de saída para a linha. O critério de saída é baseado na diminuição do tamanho de cache para um nível abaixo do limite que é determinado como uma percentagem do limite (o valor padrão é 10%). Apagar mais do que é necessário para se evitar exceder ao limite impede o cache de sobrecarregar e ficar tentando realocar recursos, a cada vez que um novo arquivo for adicionado.

[00363] Em algumas modalidades, o serviço de transmissão 554 provê a capacidade de copiar todo arquivo em uma pluralidade de arquivos aplicativos compreendendo um programa aplicativo, em um formato de arquivo comprimido, para a máquina local 10. Esta capacidade pode ser referida como "pré-cache". Em uma destas modalidades, quando o programa aplicativo é subsequente executado, todas as requisições de pacote vão para a cópia local, ao invés de

atravessarem a rede. Estas modalidades permitem que um usuário da máquina local 10 execute o programa aplicativo em um momento quando o usuário não tiver acesso à rede.

[00364] Uma máquina remota 30 inclui uma funcionalidade para monitoração de uso de aplicativo por uma máquina local 10. A máquina remota 30 pode monitorar o status de cada aplicativo usado pela máquina local 10, por exemplo, quando da execução ou do término de um aplicativo. Em uma modalidade, a máquina remota 30 requer que a máquina local 10 transmita mensagens sobre o status de um aplicativo executado pela máquina local 10. Em uma outra modalidade, quando uma máquina local 10 se conecta a uma rede na qual a máquina remota 30 reside, a máquina local 10 transmite uma mensagem indicando que a máquina local 10 se conectou à rede.

[00365] Em uma modalidade, a máquina local 10 é dita como tendo uma sessão quando a máquina local 10 interage com a máquina remota 30 e executa um ou mais aplicativos. Em uma outra modalidade, a máquina remota 30 requer que a máquina local mantenha, pela duração de uma sessão, uma licença de autorização de execução dos aplicativos recebidos a partir de uma máquina remota. Ainda em uma outra modalidade, as sessões têm identificadores de sessão únicos atribuídos pela máquina remota.

[00366] Em uma modalidade, a máquina local 10 transmite mensagens diretamente para o servidor de gerenciamento de sessão 562. Em uma outra modalidade, a máquina local 10 transmite mensagens para uma máquina remota 30, a máquina remota 30 encaminhando as mensagens para o servidor de gerenciamento de sessão 562 com uma identificação da máquina local 10.

[00367] Em algumas modalidades, o servidor de gerenciamento de sessão 562 é uma máquina remota 30 provendo serviços de gerenciamento de licença e monitoração de sessão. Em uma destas

modalidades, o servidor de gerenciamento de sessão 562 inclui um subsistema de gerenciamento de servidor 508 provendo estes serviços.

[00368] Em uma modalidade, a máquina local 10 transmite mensagens diretamente para o servidor de gerenciamento de sessão 562. Em uma outra modalidade, a cliente 10 transmite mensagens para uma máquina remota 30, a máquina remota 30 encaminhando as mensagens para o servidor de gerenciamento de sessão 562 com uma identificação da máquina local 10.

[00369] Uma máquina local 10 pode transmitir uma mensagem de batimento cardíaco para a máquina remota 30. Em uma modalidade, a mensagem de batimento cardíaco inclui uma requisição para uma licença. Nesta modalidade, a máquina local 10 pode transmitir a mensagem de batimento cardíaco após o recebimento de uma informação associada a um programa aplicativo para o qual a máquina local 10 requisitou uma autorização para execução. A máquina local 10 pode transmitir a mensagem de batimento cardíaco antes da execução do aplicativo. Em uma modalidade, a máquina local 10 inclui com a mensagem de batimento cardíaco um tíquete de abertura recebido com a informação de acesso. Nesta modalidade, a máquina remota 30 pode conceder à máquina local 552 uma licença mediante uma verificação bem-sucedida do tíquete de abertura.

[00370] Em uma outra modalidade, a mensagem de batimento cardíaco inclui uma indicação que a máquina local iniciou uma execução de um aplicativo. Ainda em uma outra modalidade, a mensagem de batimento cardíaco inclui uma indicação que a máquina local terminou a execução de um aplicativo. Ainda em uma outra modalidade, a mensagem de batimento cardíaco inclui uma indicação de uma falha na execução de um aplicativo.

[00371] Em uma modalidade, a mensagem de batimento cardíaco

inclui uma requisição para uma identificação de um segundo servidor de gerenciamento de sessão, tal como um servidor de gerenciamento de sessão 562. Em uma outra modalidade, a mensagem de batimento cardíaco inclui uma indicação que a máquina local 10 se conectou a uma rede na qual a máquina remota 30 reside.

[00372] Em algumas modalidades, a mensagem de batimento cardíaco inclui uma requisição para reinicialização de uma sessão de transmissão de aplicativo. Em uma destas modalidades, a máquina local 10 transmite esta mensagem de batimento cardíaco quando um erro tiver ocorrido e uma conexão é terminada entre uma rede na qual a máquina remota 30 reside e a máquina local 10. Em uma outra destas modalidades, a máquina local 10 transmite com a mensagem de batimento cardíaco uma informação associada à sessão. Ainda em uma outra destas modalidades, a máquina remota 30 pode transmitir para a máquina local 10 dados relacionados à sessão, se a sessão não tiver expirado.

[00373] Em uma outra destas modalidades, se uma máquina remota 30 se desconectar de uma rede na qual ela responde, a máquina local 10 poderá não receber uma resposta para uma mensagem de batimento cardíaco transmitida para a máquina remota 30. Em uma modalidade, a máquina local 10 pode restabelecer uma sessão pela transmissão de uma mensagem requisitando uma reinicialização de sessão para a máquina remota 30. Em uma outra modalidade, a máquina local 10 pode restabelecer uma sessão pela transmissão de uma mensagem requisitando uma reinicialização de sessão para uma segunda máquina remota 30. Em algumas modalidades, quando a máquina remota 30 se reconectar à rede, ela criará uma nova sessão para cada requisição de reinicialização de sessão recebida enquanto a máquina remota 30 estava desconectada. Em uma destas modalidades, a nova sessão será associada ao estado

reconectado e não licenciado. Em uma outra destas modalidades, nenhuma nova licença será adquirida para a nova sessão. Ainda em uma outra destas modalidades, quando a máquina local 10 executa um aplicativo, uma nova licença será adquirida e todas as sessões associadas à máquina local 10 estarão associadas a um estado ativo e licenciado.

[00374] Em algumas modalidades, um cliente de transmissão de aplicativo 552 na máquina local 10 gera a mensagem de batimento cardíaco. Em uma destas modalidades, o cliente de transmissão de aplicativo 552 encaminha a mensagem de batimento cardíaco para uma interface da web 558 para transmissão para a máquina local 10 para transmissão para a máquina remota 30. Em outras modalidades, uma máquina remota 30 compreende um ponto coletor 240 (descrito acima em relação à figura 1D) recebe e armazena as mensagens de batimento cardíaco.

[00375] Em algumas modalidades, o cliente de transmissão de aplicativo 552 requer uma licença a partir da máquina remota 30. Em uma destas modalidades, a licença autoriza uma execução de um programa aplicativo na máquina local 552. Em uma outra destas modalidades, a máquina remota 30 pode acessar uma segunda máquina remota para a provisão da licença. Ainda em uma outra destas modalidades, a máquina remota 30 pode prover a licença para a máquina local. Ainda em uma outra destas modalidades, a máquina remota 30 pode prover uma licença aceitável para fins de autorização para uma segunda máquina remota. Em algumas modalidades, a licença é revogada mediante a terminação de execução de um programa aplicativo.

[00376] Em algumas modalidades, uma máquina remota 30 no banco 38 inclui um subsistema de gerenciamento de licença para configuração e manutenção de licenças para aqueles subsistemas que

requerem uma licença para operação e para controle do número de conexões para esses subsistemas. Em outras modalidades, a máquina remota 30 incorpora uma funcionalidade de um subsistema de gerenciamento de licença em outros subsistemas, tais como o subsistema de gerenciamento de aplicativo e o subsistema de gerenciamento de sessão. Em uma modalidade, cada máquina remota 30 inclui um subsistema de gerenciamento de licença ou a funcionalidade associada a um subsistema de gerenciamento de licença. O subsistema de gerenciamento de licença gerencia dois tipos de licenças: (1) licenças de recurso e (2) licenças de conexão. Em uma visão geral breve, o subsistema de gerenciamento de licença usa as licenças de recurso para controle do acesso a "recursos" de produtos de software licenciados. Um recurso pode ser algum aspecto ou uma funcionalidade em particular do produto de software, ou o recurso pode ser o produto inteiro que não trabalhará sem uma licença de recurso.

[00377] A figura 15 mostra uma modalidade da máquina remota 30 no banco 38 na qual a máquina remota 30 inclui um subsistema de gerenciamento de licença 1510, um subsistema de grupo 1520, um módulo de serviço de sistema de armazenamento persistente 1570, um módulo de serviço de sistema de armazenamento dinâmico 1580, um subsistema de relação 1530, um subsistema de máquina remota especializado 1540 e um subsistema de ponto de acesso comum 524 em comunicação com um barramento de evento 1570. Aqueles subsistemas mostrados na figura 15 são para fins de descrição do comportamento do subsistema de gerenciamento de licença 1510. A máquina remota 30 pode incluir outros tipos de subsistemas.

[00378] O subsistema de gerenciamento de licença 1510 se comunica com o subsistema de grupo 1520 por um barramento de evento para a formação e a manutenção de um agrupamento lógico de

licenças (a partir deste ponto, "grupos de licença") para facilitação dos grupos de licença, atribuições e grupos. Um grupo de licença inclui uma coleção de strings de licença, descrito abaixo, e/ou outros grupos de licença. Os grupos de licença coletam licenças de recursos similares e, conseqüentemente, permitem um agrupamento de licenças. Uma licença agrupada é uma licença que está disponível para uso por qualquer máquina remota 30 no banco 38. Cada grupo de licença mantém as capacidades coletivas das licenças no grupo de licença e outros subgrupos de licença (isto é, outros grupos de licença em um grupo de licença). Uma informação relativa a agrupamentos de licença é mantida, em uma modalidade, no armazenamento dinâmico 240. Nesta modalidade, cada subsistema de gerenciamento de licença 1610 armazena localmente o número total de licenças e o número de licença atribuído a uma máquina remota 30 no banco 38. Mediante a concessão de uma licença agrupada, o subsistema de gerenciamento de licença 1510 faz uma entrada no armazenamento dinâmico 240 indicando que uma licença agrupada está "em uso". Todo outro subsistema de gerenciamento de licença 1510 reconhece que essa licença agrupada está indisponível para concessão. Em uma modalidade em particular, o armazenamento dinâmico 240 armazena pares de ID de máquina remota/ID de cliente associados a cada grupo de licença para a identificação das licenças agrupadas que estiverem em uso.

[00379] O subsistema de relação 1530 mantém associações entre licenças e máquinas remotas 30 e entre grupos de licença e máquinas remotas 30. As associações definem o número de licenças para cada licença e grupo de licença que apenas a máquina remota associada 30 pode obter (isto é, "licenças locais"). Uma licença local é uma licença que é atribuída a uma máquina remota no banco 38 e não é compartilhada por outras máquinas remotas 38. O subsistema de

gerenciamento de licença 1510 se comunica com o subsistema de relação 1530 para criar, apagar, consultar e atualizar tais associações. O subsistema de ponto de acesso comum 524 provê chamadas de procedimento remotas (RPCs) para uso por produtos de software residentes na máquina remota 30. Estas interfaces de RPC permitem que esses produtos de software se comuniquem através do subsistema de acesso comum 524 para acessarem uma informação de licença.

[00380] Ainda com referência à figura 15, o subsistema de máquina remota especializado 1540 se comunica com o subsistema de gerenciamento de licença 1510 para a obtenção de uma licença de recurso para cada capacidade do subsistema de máquina remota especializado 1540 para o qual uma licença seja requerida. Isto ocorre na inicialização de subsistema de máquina remota especializado 1540 e após qualquer evento de licença. Caso incapaz de obter a licença de recurso, o subsistema de máquina remota especializado 1540 restringe a funcionalidade que aquele subsistema proverá com uma licença. Também, o subsistema de máquina remota especializado 1540 usa o subsistema de gerenciamento de licença 1510 para a obtenção de licenças de conexão de cliente sempre que uma sessão de cliente for iniciada com a máquina remota 30.

[00381] O subsistema de gerenciamento de licença 1510 se comunica com o módulo de serviço de armazenamento persistente 352 para o armazenamento de licenças de recurso e conexão em um depósito de licença 1550 como strings de licença formadas de acordo com uma convenção de denominação. O depósito de licença 1550 reside no armazenamento persistente 230. Verificações cíclicas de redundância (CRC) impedem uma violação das licenças, enquanto essas licenças estiverem armazenadas no depósito de licença 1550. O subsistema de gerenciamento de licença 1510 também armazena uma

informação relacionada a strings de licença no depósito de licença 1550. Por exemplo, a informação pode indicar quais licenças estão atribuídas a quais máquinas remotas 30 do banco 38 e, em algumas modalidades, o status de ativação de cada licença. Em uma modalidade, uma tabela de licença de conexão 1560 armazena identificadores daquelas máquinas locais que obtiveram uma licença de conexão.

[00382] Em uma modalidade, o subsistema de gerenciamento de licença 1510 suporta eventos para subsistemas requisitando o uso de uma capacidade licenciada, tal como uma requisição para uma licença agrupada disponível. O evento inclui a UID do subsistema requisitando a lâmina cirúrgica e a UID da máquina remota 30 na qual aquele subsistema reside. O evento também contém o tipo de licença requisitada (isto é, uma licença de recurso ou de conexão) na forma de uma ID de grupo de licença. A ID de grupo de licença real armazenada no armazenamento persistente 230 é arbitrária, mas uma aderência à convenção de denominação provê flexibilidade para a adição futura de novos produtos de software (isto é, subsistemas) à máquina remota 30.

[00383] O evento enviado por um subsistema requisitante buscando uma licença inclui (1) uma indicação do tipo de grupo de licença, a identidade da máquina local e a máquina remota requisitando a licença, e um indicador (flag) de "forçar aquisição". Uma indicação de tipo de grupo de licença pode incluir uma identificação de uma licença de recurso, tal como um gerenciamento de carga, ou uma licença de tipo de conexão, tal como um programa aplicativo de software. O campo identificando a máquina local e a máquina remota buscando a licença pode incluir o identificador único associado à máquina remota e à máquina local. O indicador de forçar aquisição pode ser usado, por exemplo, para a reaquisição de licenças de conexão após um evento

de mudança de licença. Um evento de mudança de licença indica que uma informação de licença do armazenamento persistente 230 mudou; por exemplo, uma licença foi apagada, adicionada ou atribuída. Mediante um evento de mudança de licença, cada máquina remota 30 tenta readquirir todas as licenças de conexão que ela possuía antes do evento de mudança de licença, porque a causa em particular do evento de mudança de licença é desconhecida para aquela máquina remota. Este indicador, se regulado, indica que uma licença conexão deve ser adquirida, mesmo caso fazê-lo aumente o número de conexões para a máquina remota 30 além do número máximo predeterminado de conexões admissíveis. Nenhuma licença de conexão nova é subsequente concedida até o número de licenças de conexão em uso cair abaixo deste número máximo predeterminado. Desta maneira, uma conexão de máquina local não será terminada na metade de uma sessão devido a um evento de mudança de licença.

[00384] Com referência, agora, à figura 16, um diagrama de blocos descreve uma modalidade dos componentes envolvidos no cumprimento de licenciamento. Uma máquina remota 30 inclui um subsistema de gerenciamento de servidor 508 e um subsistema de gerenciamento de licença 512. Em algumas modalidades, o subsistema de gerenciamento de servidor 508 e o subsistema de gerenciamento de licença 512 provêm a funcionalidade do subsistema de gerenciamento de licença 1510 descrito acima. Em outras modalidades, um subsistema de gerenciamento de aplicativo 506 e um subsistema de gerenciamento de sessão 510 provêm a funcionalidade do subsistema de gerenciamento de licença 1510 descrito acima. Ainda em outras modalidades, outros subsistemas provêm a funcionalidade do subsistema de gerenciamento de licença 1510 descrito acima.

[00385] Em uma modalidade, o subsistema de gerenciamento de servidor 508 pode incluir um componente de licenciamento usado para a requisição de emissão e revogação de licenças. Em uma outra modalidade, o subsistema de gerenciamento de licença 512 pode aplicar uma política a uma requisição para emissão ou revogação de uma licença recebida a partir do subsistema de gerenciamento de servidor 508. Ainda em uma outra modalidade, o subsistema de gerenciamento de licença 512 pode transmitir a requisição para uma máquina remota 30 provendo uma funcionalidade de cumprimento de licença. Em algumas modalidades, o serviço de gerenciamento 504 pode manter uma conexão com uma máquina remota 30 provendo uma funcionalidade de cumprimento de licença. Em outras modalidades, a máquina remota 30 provê a funcionalidade de cumprimento de licença.

[00386] Em algumas modalidades, uma licença expira e cessa de ser válida mediante uma falha da máquina local 10 em transmitir um número predeterminado de mensagens de batimento cardíaco para a máquina remota. Em uma destas modalidades, uma expiração da licença revoga uma autorização para execução de um programa aplicativo pela máquina local 10.

[00387] Em outras modalidades, uma sessão se esgota após a expiração de um período de tempo predeterminado. Em uma modalidade, o serviço de gerenciamento 504 mantém dados relacionados a uma sessão após a expiração de uma licença até uma expiração de uma sessão. Em algumas modalidades, os dados relacionados à sessão podem incluir uma informação tal como nome de sessão, id de sessão, id de cliente, horário de começo de sessão, nome de servidor (Caminho UNC de Servidor de Arquivo), nome de aplicativo (nome único gerado por máquina local, com base em um nome de navegador), pseudônimo, estado de sessão (ativo/licenciado,

ativo/não licenciado, reconectado/não licenciado). Em uma outra modalidade, a máquina local 10 cessa uma transmissão de mensagens de batimento cardíaco e recomeça uma transmissão de mensagens de batimento cardíaco em um ponto posterior no tempo. Ainda em uma outra modalidade, o serviço de gerenciamento 504 pode reemitir uma licença e tornar os dados relacionados à sessão mantidos disponíveis para a máquina local 10, se a 10 recomeçar uma transmissão de mensagens de batimento cardíaco antes da expiração da sessão.

[00388] Com referência, agora, à figura 17, um fluxograma descreve uma modalidade das etapas seguidas para a requisição e a manutenção de uma licença a partir de uma máquina remota 30 pela duração de uma sessão em uma máquina local 10. Em uma visão geral breve, um cliente de transmissão de aplicativo requisita uma licença (etapa 1702). Uma máquina remota 30 recebe a requisição para a licença, verifica um tíquete associado à requisição, e gera uma licença (etapa 1704). A máquina remota 30 provê a licença e uma informação associada à licença para a máquina local 10 (etapa 1706). A máquina local 10 executa o aplicativo, conforme descrito acima, em relação à etapa 214 na figura 7. A máquina local transmite uma mensagem de batimento cardíaco indicando que a máquina local executou um aplicativo (etapa 1708). A máquina remota 30 recebe a mensagem de batimento cardíaco e verifica uma informação de identificação transmitida com a mensagem de batimento cardíaco (etapa 1708). A máquina remota 30 cria uma sessão associada ao aplicativo executado e à máquina local 10 (etapa 1710). Um resultado da criação da sessão é transmitido para a máquina local 10 (etapa 1712). A máquina local transmite as mensagens de batimento cardíaco por toda a execução do aplicativo, conforme descrito acima em relação à etapa 216 da figura 7. A máquina local recebe uma resposta a uma

mensagem de batimento cardíaco transmitida (etapa 1714). A máquina local transmite uma mensagem de batimento cardíaco indicando uma terminação de uma execução do aplicativo (etapa 1716). A máquina remota 30 recebe a mensagem de batimento cardíaco e determina se é para remover os dados relacionados à sessão e se é para liberar a licença associada à máquina local 10 e ao aplicativo terminado (etapa 1718). Um resultado da determinação feita pela máquina remota 30 é transmitido para a máquina local 10 (etapa 1720).

[00389] Com referência, agora, à figura 17, e em maiores detalhes, um cliente de transmissão de aplicativo em uma máquina local 10 requisita uma licença (etapa 1702). Em algumas modalidades, a máquina local 10 requisita a licença ao receber uma informação de acesso associada a um programa aplicativo. Em uma destas modalidades, a máquina local requisita uma licença da máquina remota 30 concedendo uma autorização para execução do programa aplicativo pela máquina local 10. Em algumas modalidades, a requisição para a licença inclui um tíquete de abertura recebido a partir da máquina remota 30 com a informação de acesso. Em outras modalidades, um cliente de transmissão de aplicativo 552 na máquina local 10 transmite a requisição para uma interface da web 558 e a interface da web 558 transmite a requisição para a máquina remota 30. Ainda em outras modalidades, um subsistema de gerenciamento de sessão 510 na máquina remota recebe e processa a requisição para a licença.

[00390] Uma máquina remota 30 recebe a requisição para a licença, verifica um tíquete associado à requisição, e gera uma licença (etapa 1704). Em uma modalidade, a máquina remota 30 verifica que a máquina local 10 está autorizada a executar o aplicativo. Em uma outra modalidade, a máquina remota 30 determina se a máquina local 10 já está associada a uma licença existente. Ainda em uma outra

modalidade, a máquina remota 30 determina que a máquina local 10 está associada a uma licença existente e provê à máquina local 10 um identificador para um servidor de gerenciamento de sessão 562 gerenciando a licença existente. Ainda em uma outra modalidade, a máquina remota 30 gera e provê à máquina local 10 uma nova licença, um identificador de sessão e uma identificação de um servidor de gerenciamento de sessão 562 gerenciando a nova licença.

[00391] Em algumas modalidades, a máquina remota 30 usa um subsistema de gerenciamento de licença 1510 para responder a uma requisição de licença em uma modalidade na qual o subsistema de gerenciamento de licença 1510 recebe uma requisição de licença. A requisição pode ser para uma licença de recurso ou para uma licença de conexão. O subsistema de gerenciamento de licença 1510 determina se a licença já foi concedida, isto é, se o recurso já foi começado ou uma conexão para uma máquina local já existe. Se a licença já for concedida, o subsistema de gerenciamento de licença 1510 enviará um evento "concedido" para o requisitante de licença. Se a licença não tiver sido previamente concedida, o subsistema de gerenciamento de licença 1510 determinará se uma licença local, isto é, uma licença que foi permanentemente atribuída à máquina remota 30 está disponível. Em algumas modalidades, o subsistema de gerenciamento de licença 1510 realiza esta determinação ao verificar uma memória local. Se uma licença local estiver disponível, isto é, a máquina remota 30 tiver mais licenças permanentemente atribuídas do que concedidas atualmente, o subsistema de gerenciamento de licença 1510 enviará um evento "concedido" para o requisitante de licença.

[00392] A máquina remota 30 provê a licença e a informação associada à licença para a máquina local 10 (etapa 1706). Em uma modalidade, mediante o recebimento da licença, do identificador de

sessão e da identificação do servidor de gerenciamento de sessão 562 a partir da máquina remota 30, a máquina local 10 executa o aplicativo. A máquina local 10 pode executar o aplicativo, conforme descrito acima em relação à etapa 214 na figura 7. A máquina local transmite uma mensagem de batimento cardíaco indicando que a máquina local executou um aplicativo (etapa 1708). Em uma modalidade, a máquina local transmite a mensagem de batimento cardíaco para a máquina remota 30 para transmissão da mensagem de batimento cardíaco para um servidor de gerenciamento de sessão 562. Em uma outra modalidade, a máquina local 10 transmite uma mensagem de batimento cardíaco diretamente para um servidor de gerenciamento de sessão 562, em resposta a um identificador do servidor de gerenciamento de sessão 562 recebido a partir da máquina remota 30.

[00393] A máquina remota 30 recebe a mensagem de batimento cardíaco e verifica uma informação de identificação transmitida com a mensagem de batimento cardíaco (etapa 1708). Em uma modalidade, uma máquina remota 30' é o servidor de gerenciamento de sessão 562. Em uma outra modalidade, o servidor de gerenciamento de sessão 562 verifica um identificador de servidor provido com uma mensagem de batimento cardíaco pela máquina local 10. Em ainda uma outra modalidade, o identificador de servidor é o identificador provido para a máquina local 10 por uma máquina remota 30.

[00394] A máquina remota 30 cria uma sessão associada ao aplicativo executado e à máquina local 10 (etapa 1710). Em uma modalidade, o servidor de gerenciamento de sessão 562 cria uma nova sessão associada ao agente de pesquisa em execução mediante o recebimento da mensagem de batimento cardíaco. Em uma outra modalidade, uma terceira máquina remota 30 cria a nova sessão. Em algumas modalidades, o servidor de gerenciamento de sessão 562

armazena uma informação relacionada à sessão mediante a criação da nova sessão.

[00395] Um resultado de criação da sessão é transmitido para a máquina local 10 (etapa 1712). Em algumas modalidades, o resultado confirma a criação da sessão. Em outras modalidades, o resultado identifica o aplicativo ou os aplicativos associados à sessão. A máquina local transmite mensagens de batimento cardíaco por toda a execução do aplicativo, conforme descrito acima em relação à etapa 216 da figura 7. Em uma modalidade, a máquina local 10 continua a transmitir as mensagens de batimento cardíaco em intervalos regulares por toda a execução do programa aplicativo. A máquina local recebe uma resposta para uma mensagem de batimento cardíaco transmitida (etapa 1714). Em uma modalidade, a máquina local 10 recebe uma confirmação de recebimento de mensagens de batimento cardíaco a partir do servidor de gerenciamento de sessão 562. Em uma outra modalidade, a máquina local 10 recebe um comando para execução do servidor de gerenciamento de sessão 562, em resposta ao recebimento de uma mensagem de batimento cardíaco pelo servidor de gerenciamento de sessão 562.

[00396] A máquina local transmite uma mensagem de batimento cardíaco indicando uma terminação de uma execução do aplicativo (etapa 1716). A máquina remota 30 recebe a mensagem de batimento cardíaco e determina se é para remover dados relacionados à sessão e se é para liberar a licença associada à máquina local 10 e ao aplicativo terminado (etapa 1718). Um resultado da determinação feito pela máquina remota 30 é transmitido para a máquina local 10 (etapa 1720).

[00397] Com referência, agora, à figura 18, um diagrama de blocos descreve uma modalidade de estados que podem estar associados a uma sessão monitorada por um serviço de gerenciamento 504. Em

uma modalidade, um subsistema de gerenciamento de sessão 510 no serviço de gerenciamento 504 monitora uma sessão de uma máquina local 10 e atribui um estado à sessão. Em uma outra modalidade, o subsistema de gerenciamento de sessão 510 mantém uma lista de dados relacionados à licença, os quais podem incluir um identificador associado à máquina local, um identificador associado à sessão, um estado de sessão e uma estampa de tempo indicando o último momento em que a máquina remota 30 recebeu uma mensagem da máquina local 10. Em algumas modalidades, o subsistema de gerenciamento de sessão 510 inclui uma linha de monitoração de sessão. Em uma destas modalidades, a linha de monitoração de sessão é ativada em um intervalo de expiração de licença periódico para varrer a lista de dados relacionados à licença e atualizar o status de sessão de uma sessão.

[00398] Um primeiro estado em que uma sessão pode estar é um estado ativo e licenciado. Em uma modalidade, quando neste estado, a máquina local 10 manteve uma licença válida autorizando uma execução de um aplicativo. Em uma outra modalidade, um servidor de gerenciamento de sessão 562 mantém dados relacionados à sessão. Em algumas modalidades, o servidor de gerenciamento de sessão 562 armazena os dados relacionados à sessão em uma segunda máquina remota. Em uma modalidade, quando uma máquina local 10 inicialmente executa um aplicativo, a sessão para a máquina local está no estado ativo e licenciado.

[00399] Um segundo estado em que uma sessão pode estar é um estado ativo e não licenciado. Em uma modalidade, uma sessão está neste estado quando a máquina local 10 falha em transmitir mensagens de batimento cardíaco e uma licença para a máquina local 10 expirou. Em uma outra modalidade, se uma sessão estiver neste estado, então, enquanto a licença tiver expirado, um tempo insuficiente

decorreu para a sessão expirar, e a sessão será considerada ativa. Em algumas modalidades, enquanto a sessão está neste estado, uma máquina remota 30 ou um servidor de gerenciamento de sessão 562 pode armazenar dados relacionados à sessão em nome da máquina local 10. Em outras modalidades, se uma máquina local 10 transmitir uma mensagem de batimento cardíaco antes da expiração da sessão, os dados relacionados à sessão serão transmitidos para a máquina local 10 com uma nova licença e a sessão retornará para o estado ativo e licenciado. Em uma modalidade, uma máquina remota 30 usa identificadores de sessão e identificadores associados à máquina local para verificar que a sessão não expirou e prover à máquina local dados relacionados à sessão apropriados.

[00400] Um terceiro estado em que uma sessão pode estar é um estado desconectado e não existente. Quando uma sessão expira, os dados relacionados à sessão são apagados.

[00401] Um quarto estado em que uma sessão pode estar é no estado reconectado e licenciado. Em uma modalidade, quando uma sessão em uma máquina local 10 expira, os dados relacionados à sessão são apagados. Em uma outra modalidade, quando a máquina local 10 transmite uma nova mensagem de batimento cardíaco, um novo identificador de sessão e um identificador de máquina local são gerados para a máquina local 10. Em algumas modalidades, a máquina local 10 autentica a máquina remota 30, recebe uma nova licença e entra no estado ativo e licenciado.

[00402] A Tabela 3 resume os estados que podem ser associados a uma sessão.

Status de Sessão	Descrição
Ativo/Licenciado	Modo normal de operação.
Ativo/Não Licenciado	Duração de batimentos cardíacos faltando > Expiração de licença

	E
	Duração de batimentos cardíacos faltando < Expiração de sessão
Reconectado/Não Licenciado	Duração de batimentos cardíacos faltando > Expiração de sessão OU CPS/RADE hospedando a sessão está inativo e volta a estar on-line

Tabela 3

[00403] Em algumas modalidades, um mecanismo de formação de pacote permite a criação de uma pluralidade de arquivos aplicativos associados a um programa aplicativo. Em uma destas modalidades, o mecanismo de formação de pacote permite a identificação de uma pluralidade de arquivos aplicativos. Em uma outra destas modalidades, o mecanismo de formação de pacote permite o agrupamento de arquivos aplicativos individuais em uma pluralidade de arquivos aplicativos. Ainda em uma outra destas modalidades, o mecanismo de formação de pacote permite a hospedagem de uma pluralidade de arquivos aplicativos em uma máquina remota, tal como um servidor de arquivo ou um servidor de aplicativo.

[00404] Em uma modalidade, o mecanismo de formação de pacote é executado em uma máquina remota descrita como uma "máquina de estágio". Em uma outra modalidade, o mecanismo de formação de pacote é executada em uma "máquina limpa". Uma máquina limpa pode ser uma máquina remota tendo apenas um sistema operacional instalado nela, sem um software adicional, drivers, entradas de registro ou outros arquivos. Ainda em uma outra modalidade, a máquina de formação de pacote é executada em uma máquina remota, a máquina remota se parecendo com uma máquina local na qual um programa aplicativo pode ser executado. Em algumas modalidades, a máquina

remota na qual o mecanismo de formação de pacote é executado inclui um ambiente de isolamento provendo um ambiente de máquina limpa no qual um aplicativo pode ser instalado, mesmo quando a máquina remota não for em si uma máquina limpa.

[00405] Em uma modalidade, a pluralidade de arquivos aplicativos é referida como um "pacote". Em uma outra modalidade, o pacote pode ser um arquivo de armazenamento de backup armazenando a pluralidade de arquivos aplicativos. Ainda em uma outra modalidade, o pacote pode ser um arquivo de armazenamento de backup armazenado a pluralidade de arquivos aplicativos e um arquivo incluindo metadados associados a pelo menos um arquivo na pluralidade de arquivos aplicativos. Em algumas modalidades, um pacote pode incluir uma pluralidade de arquivos aplicativos compreendendo um programa aplicativo. Em outras modalidades, um pacote inclui uma pluralidade de arquivos aplicativos compreendendo uma suíte de programas aplicativos. Ainda em outras modalidades, um pacote inclui uma pluralidade de arquivos aplicativos compreendendo um programa aplicativo e um pré-requisito requerido para execução do programa aplicativo.

[00406] Em uma modalidade, o mecanismo de formação de pacote inicia a execução de um programa de instalação em um ambiente de isolamento. Em uma outra modalidade, o mecanismo de formação de pacote monitora uma mudança no ambiente de isolamento gerada pelo programa de instalação. Ainda em uma outra modalidade, o mecanismo de formação de pacote monitora uma criação pelo programa de instalação de um arquivo no ambiente de isolamento. Ainda em uma outra modalidade, o mecanismo de formação de pacote monitora uma modificação pelo programa de instalação de um arquivo no ambiente de isolamento. Em algumas modalidades, a pluralidade de arquivos aplicativos inclui um arquivo criado ou modificado pelo

programa de instalação. Em outras modalidades, o mecanismo de formação de pacote implementa um driver de filtro de sistema de arquivo 564 para a monitoração do ambiente de isolamento.

[00407] Em algumas modalidades, um mecanismo de formação de pacote pode gerar múltiplas pluralidades de arquivos aplicativos, cada uma compreendendo uma versão diferente de um programa aplicativo configurado para execução em um ambiente alvo diferente. Em uma destas modalidades, uma pluralidade de arquivos aplicativos é configurada para execução em uma máquina local tendo um sistema operacional em particular, um nível de revisão, configurações de linguagem e uma unidade mestra (por exemplo, uma pluralidade de arquivos aplicativos pode ser configurada para execução em uma máquina local tendo o sistema operacional Windows XP Professional com nível de revisão SP2 e acima, usando Inglês e tendo como unidade mestra C:\). Em uma outra destas modalidades, mais de uma pluralidade de arquivos aplicativos pode ser combinada em um arquivo de armazenamento de backup único. Ainda em uma outra destas modalidades, cada pluralidade de arquivos aplicativos pode ser referida como um "alvo". Ainda em uma outra destas modalidades, um arquivo de armazenamento de backup contendo uma ou mais pluralidades de arquivos aplicativos pode ser referido como um "pacote".

[00408] Com referência, agora, à figura 19, um diagrama de blocos descreve um pacote que inclui dois alvos, cada alvo compreendendo uma pluralidade de arquivos aplicativos compreendendo um aplicativo. Na figura 19, o programa aplicativo 'Foo' é empacotado em dois alvos. A diferença entre os dois alvos é a 'Linguagem Alvo'. Especificamente, o alvo 1 suporta 'Inglês' e o alvo 2 suporta 'Alemão'. Em uma modalidade, uma enumeração de programas aplicativos disponíveis pode listar o programa aplicativo 'Foo'. Em uma outra modalidade, a

pluralidade apropriada de arquivos é transmitida para uma máquina local requisitando acesso ao programa aplicativo. Ainda em uma outra modalidade, uma determinação é feita para a transmissão de um alvo em particular para uma máquina local, em resposta a uma avaliação da máquina local. Ainda em uma outra modalidade, um arquivo associado ao pacote identifica pelo menos uma característica associada a um alvo no pacote e requerida para execução em uma máquina local.

[00409] Em algumas modalidades, o mecanismo de pacote 530 prepara um programa aplicativo para transmissão pela execução de um programa de instalação associado ao programa aplicativo. Em uma destas modalidades, o mecanismo de formação de pacote gera um ambiente de isolamento na máquina remota 30 no qual o mecanismo de formação de pacote está em execução. Em uma outra destas modalidades, o mecanismo de formação de pacote executa o programa aplicativo no ambiente de isolamento. Ainda em uma outra destas modalidades, o mecanismo de formação de pacote identifica uma pluralidade de arquivos aplicativos gerados ou modificados pelo programa de instalação. Ainda em uma outra destas modalidades, o mecanismo de formação de pacote cria um arquivo de armazenamento de backup incluindo a pluralidade de arquivos aplicativos. Em uma destas modalidades, o mecanismo de formação de pacote cria um arquivo .CAB incluindo a pluralidade de arquivos aplicativos. Em uma outra destas modalidades, o mecanismo de formação de pacote cria um diretório e armazena a pluralidade de arquivos aplicativos no diretório. Em algumas modalidades, o mecanismo de formação de pacote armazena a pluralidade de arquivos aplicativos em um servidor de arquivo ou em uma outra máquina remota 30. Em outras modalidades, o mecanismo de formação de pacote armazena a pluralidade de arquivos aplicativos

em múltiplas máquinas remotas.

[00410] Com referência, agora, à figura 20, um fluxograma descreve uma modalidade das etapas seguidas em um método baseado em política para se instalar efetivamente um programa aplicativo, sem rebutar um sistema operacional. Em uma breve visão geral, um mecanismo de formação de pacote executa um programa instalador em um ambiente de isolamento, o programa instalador instalando pelo menos um arquivo aplicativo associado a um segundo aplicativo no ambiente de isolamento (etapa 2002). Uma chamada pelo programa instalador para pelo menos uma interface de programação de aplicativo (API) é interceptada, a chamada requerendo a execução de uma ação após um reboot de um sistema operacional (etapa 2004). A ação de pelo menos uma chamada interceptada é executada sem reboot do sistema operacional (etapa 2006). Uma identificação de um tipo de arquivo de pelo menos um arquivo aplicativo é recebida (etapa 2008). Pelo menos um método de execução está associado a pelo menos um arquivo aplicativo instalado, em resposta ao tipo de arquivo identificado (etapa 2010). Pelo menos um arquivo aplicativo instalado é armazenado em pelo menos um servidor (etapa 2012). Uma enumeração é gerada do segundo aplicativo, pelo menos um arquivo aplicativo instalado, uma localização de pelo menos um servidor e pelo menos um método de execução (etapa 2014).

[00411] Com referência, agora, à figura 20, e em maiores detalhes, um mecanismo de formação de pacote executa um programa instalador em um ambiente de isolamento, o programa instalador instalando pelo menos um arquivo aplicativo associado a um segundo aplicativo no ambiente de isolamento (etapa 2002). Em uma modalidade, a execução do programa instalador no ambiente de isolamento permite que o mecanismo de formação de pacote isole

mudanças feitas pelo programa instalador para um arquivo ou registro na máquina local. Em uma outra modalidade, o mecanismo de formação de pacote intercepta uma mudança requisitada pelo programa instalador e redireciona a mudança para o ambiente de isolamento para se evitar que a mudança ocorra na máquina local. Ainda em outras modalidades, o mecanismo de formação de pacote executa um segundo programa instalador no ambiente de isolamento, o segundo aplicativo instalando pelo menos um arquivo aplicativo associado a um terceiro aplicativo no ambiente de isolamento.

[00412] Em algumas modalidades, o mecanismo de formação de pacote executa o programa de instalação no ambiente de isolamento, o programa de instalação executando pelo menos um aplicativo executável associado a um aplicativo dentro do ambiente de isolamento. Em uma modalidade na qual o instalador executa um aplicativo, a execução do aplicativo permite a instalação de um segundo aplicativo.

[00413] Em uma outra destas modalidades, a instalação de um aplicativo requer a execução de pelo menos um aplicativo executável, além da execução do programa instalador. Em uma outra destas modalidades, a instalação de um aplicativo requer a execução de um aplicativo navegador da Internet, além da execução do programa instalador. Em algumas modalidades, um programa instalador é executado para instalação de um programa e a execução do programa instalador inclui a execução de um segundo programa requerido para instalação do programa. Em uma destas modalidades, o programa é um plug-in. Em uma outra destas modalidades, o programa é um componente Active X. Ainda em uma outra destas modalidades, o programa é um componente Flash. Ainda em uma outra destas modalidades, o programa é uma barra de ferramentas personalizada, tal como uma barra de ferramentas do Yahoo! ou do Google. Em

outras modalidades, o programa é um componente instalado no segundo programa e não executável independentemente do segundo programa.

[00414] Uma chamada pelo programa instalador para pelo menos uma interface de programação de aplicativo (API) é interceptada, a chamada requerendo a execução de uma ação após um reboot de um sistema operacional (etapa 2004). A ação de pelo menos uma chamada interceptada é executada sem reboot do sistema operacional (etapa 2006). Em algumas modalidades, a execução da ação compreende a execução de uma ação de uma entrada de registro modificada durante uma instalação. Detalhes adicionais com referência à execução de pelo menos uma chamada interceptada sem reboot do sistema operacional são providos em relação à figura 25 abaixo.

[00415] Uma identificação de um tipo de arquivo de pelo menos um arquivo aplicativo é recebida (etapa 2008). Pelo menos um método de execução está associado a pelo menos um arquivo aplicativo instalado, em resposta ao tipo de arquivo identificado (etapa 2010). Em uma modalidade, pelo menos um método de execução permite uma transmissão de pelo menos um arquivo aplicativo para um cliente. Em uma outra modalidade, pelo menos um método de execução permite a execução de pelo menos um arquivo aplicativo instalado em um cliente. Em ainda uma outra modalidade, pelo menos um método de execução permite a execução de pelo menos um arquivo aplicativo instalado em um servidor. Ainda em uma outra modalidade, pelo menos um método de execução permite a transmissão de pelo menos um arquivo aplicativo para um servidor.

[00416] Pelo menos um arquivo aplicativo instalado é armazenado em pelo menos um servidor (etapa 2012). Em algumas modalidades, o programa aplicativo instalado é executado no ambiente de isolamento

antes do armazenamento de pelo menos um arquivo aplicativo instalado em pelo menos um servidor. Em uma destas modalidades, um arquivo aplicativo adicional é gerado em resposta à execução do programa aplicativo instalado. Em uma outra destas modalidades, um arquivo de dados é gerado. Ainda em uma outra destas modalidades, o programa aplicativo instalado requer uma informação para completar a instalação, a informação sendo requerida após um processo de instalação inicial. Ainda em uma outra destas modalidades, uma informação, tal como identificadores de produto de software, identificadores de licença ou outras credenciais, é requerida.

[00417] Em algumas modalidades, um identificador é provido identificando uma localização de pelo menos um arquivo aplicativo instalado em pelo menos um servidor. Em uma destas modalidades, o identificador se conforma a uma Convenção de Denominação Universal (UNC). Em outras modalidades, pelo menos um arquivo aplicativo instalado é colocado em um arquivo de armazenamento de backup, tal como um arquivo .CAB. Em uma destas modalidades, uma pluralidade de arquivos aplicativos é armazenada em um arquivo de armazenamento de backup e o arquivo de armazenamento de backup é armazenado em pelo menos um servidor. Ainda em uma outra destas modalidades, pelo menos um arquivo aplicativo instalado é armazenado em múltiplos servidores. Ainda em outras modalidades, pelo menos um arquivo aplicativo é colocado em um diretório armazenando arquivos aplicativos.

[00418] Uma enumeração é gerada do segundo aplicativo, pelo menos um arquivo aplicativo instalado, uma localização de pelo menos um servidor e pelo menos um método de execução (etapa 2014). Em algumas modalidades, a enumeração é armazenada em um arquivo. Em outras modalidades, a enumeração é armazenada em um arquivo de manifesto. Ainda em outras modalidades, a enumeração é

armazenada em um arquivo XML.

[00419] Em uma modalidade, uma enumeração é gerada de múltiplos aplicativos, uma pluralidade de arquivos aplicativos instalados associados a cada um dos múltiplos aplicativos, e uma localização de pelo menos um servidor armazenando a pluralidade de arquivos aplicativos instalados. Em uma outra modalidade, uma enumeração é gerada incluindo uma associação entre o segundo aplicativo e uma pluralidade de arquivos aplicativos instalados. Ainda em uma outra modalidade, uma enumeração é gerada incluindo uma associação entre o segundo aplicativo e um arquivo comprimido contendo pelo menos um arquivo aplicativo instalado.

[00420] Com referência, agora, à figura 21, um fluxograma descreve uma modalidade das etapas seguidas em um método baseado em política para instalação de um programa aplicativo sem rebutar um sistema operacional. Em uma breve visão geral, um mecanismo de formação de pacote executa um programa instalador em um ambiente de isolamento, o programa instalador instalando pelo menos um arquivo aplicativo associado a um segundo aplicativo no ambiente de isolamento (etapa 2102). Uma chamada pelo programa instalador para pelo menos uma interface de programação de aplicativo (API) é interceptada, a chamada requerendo a execução de uma ação após um reboot de um sistema operacional (etapa 2104). A ação de pelo menos uma chamada interceptada é executada sem reboot do sistema operacional (etapa 2106). Uma identificação de uma característica de pelo menos um arquivo aplicativo é recebida (etapa 2108). Pelo menos um pré-requisito de execução está associado a pelo menos um arquivo aplicativo instalado, em resposta à característica identificada (etapa 2110). Pelo menos um arquivo aplicativo instalado é armazenado em pelo menos um servidor (etapa 2112). Uma enumeração é gerada do segundo aplicativo, pelo menos

um arquivo aplicativo instalado, uma localização de pelo menos um servidor e pelo menos um pré-requisito de execução (etapa 2114).

[00421] Com referência, agora, à figura 21, e em maiores detalhes, um mecanismo de formação de pacote executa um programa instalador em um ambiente de isolamento, o programa instalador instalando pelo menos um arquivo aplicativo associado a um segundo aplicativo no ambiente de isolamento (etapa 2102). Em uma modalidade, a execução do programa instalador no ambiente de isolamento permite que o mecanismo de formação de pacote isole mudanças feitas pelo programa instalador para um arquivo ou registro na máquina local. Em uma outra modalidade, o mecanismo de formação de pacote intercepta uma mudança requisitada pelo programa instalador e redireciona a mudança para o ambiente de isolamento para se evitar que a mudança ocorra na máquina local. Ainda em outras modalidades, o mecanismo de formação de pacote executa um segundo programa instalador no ambiente de isolamento, o segundo aplicativo instalando pelo menos um arquivo aplicativo associado a um terceiro aplicativo no ambiente de isolamento.

[00422] Em algumas modalidades, o mecanismo de formação de pacote executa o programa de instalação no ambiente de isolamento, o programa de instalação executando pelo menos um aplicativo executável associado a um aplicativo dentro do ambiente de isolamento. Em uma modalidade na qual o instalador executa um aplicativo, a execução do aplicativo permite a instalação de um segundo aplicativo. Em uma outra destas modalidades, a instalação de um aplicativo requer a execução de pelo menos um aplicativo executável, além da execução do programa instalador. Em uma outra destas modalidades, a instalação de um aplicativo requer a execução de um aplicativo navegador da Internet, além da execução do programa instalador.

[00423] Com referência à frente à figura 23, um diagrama de blocos descreve uma modalidade de um sistema incluindo um mecanismo de pacote 530 executando um programa instalador 2350 em um ambiente de isolamento 532 e um driver de filtro de sistema de arquivo 534 em comunicação com o mecanismo de pacote 530 e o ambiente de isolamento 532.

[00424] Em uma modalidade, o mecanismo de pacote 530 gera um pacote (conforme descrito acima em relação à figura 21) pela instalação de um programa aplicativo em um ambiente de isolamento 532. Em uma outra modalidade, o mecanismo de pacote 530 instala o programa aplicativo no ambiente de isolamento 532 pela execução do programa instalador 2350. Em algumas modalidades, o mecanismo de pacote 530 inclui uma interface gráfica de usuário. Em uma destas modalidades, a interface gráfica de usuário permite que um usuário do mecanismo de pacote 530 personalize a geração de um pacote pelo mecanismo de pacote 530. Em uma outra destas modalidades, o mecanismo de pacote 530 está em comunicação com uma interface gráfica de usuário na suíte de controle de acesso 520, permitindo que um usuário da suíte de controle de acesso 520 personalize a geração de um pacote pelo mecanismo de pacote 530.

[00425] Em algumas modalidades, o driver de filtro de sistema de arquivo 532 permite a instalação do programa aplicativo em um ambiente de isolamento 532. Em uma destas modalidades, o driver de filtro de sistema de arquivo 532 intercepta uma requisição pelo programa instalador 2350. Em uma outra destas modalidades, o driver de filtro de sistema de arquivo 532 redireciona a requisição pelo programa instalador 2350 para o ambiente de isolamento 532. Ainda em uma outra destas modalidades, o driver de filtro de sistema de arquivo 532 armazena um registro da requisição feita pelo programa instalador 2350. Ainda em uma outra destas modalidades, o driver de

filtro de sistema de arquivo 532 armazena uma cópia de um arquivo criado ou modificado pelo programa instalador 2350. Em algumas modalidades, os registros armazenados gerados pelo driver de filtro de sistema de arquivo 532 são armazenados em conjunto como uma pluralidade de arquivos aplicativos compreendendo um programa aplicativo. Em outras modalidades, a pluralidade de arquivos aplicativos é armazenada em um servidor de arquivo 540.

[00426] Com referência de volta à figura 21, uma chamada pelo programa instalador para pelo menos uma interface de programação de aplicativo (API) é interceptada, a chamada requerendo a execução de uma ação após um reboot de um sistema operacional (etapa 2104). A ação de pelo menos uma chamada interceptada é executada sem reboot do sistema operacional (etapa 2106). Em algumas modalidades, a execução da ação compreende a instalação de um driver configurado para ser começado quando do boot do sistema de computador. Em outras modalidades, a execução da ação compreende a execução de uma ação de uma entrada de registro modificada durante uma instalação.

[00427] Uma identificação de um tipo de arquivo de pelo menos um arquivo aplicativo é recebida (etapa 2108). Em algumas modalidades, uma identificação de um tipo de sistema operacional é recebida. Em outras modalidades, uma identificação de uma linguagem usada pelo sistema operacional é recebida. Ainda em outras modalidades, uma identificação de uma versão do segundo aplicativo é recebida.

[00428] Pelo menos um pré-requisito de execução está associado a pelo menos um arquivo aplicativo instalado, em resposta à característica identificada (etapa 2110). Em uma modalidade, pelo menos um pré-requisito de execução está associado a pelo menos um arquivo aplicativo instalado em resposta a uma aplicação de uma política à característica. Em uma outra modalidade, um script está

associado a pelo menos um arquivo aplicativo instalado, o script compreendendo um programa executável determinando a existência de pelo menos um pré-requisito de execução em um cliente. Com referência à frente à figura 22, um instantâneo de tela descreve uma modalidade de uma enumeração de scripts a serem executados na máquina local. Um tipo de script 2202 indica quando o script deve ser executado, por exemplo, antes da execução do aplicativo, ou após a terminação da execução do aplicativo. Um indicador de isolamento 24 indica se o script deve ser executado em um ambiente de isolamento na máquina local 10. Conforme mostrado na figura 22, em algumas modalidades, o script foi associado ao programa aplicativo no momento em que a pluralidade de arquivos aplicativos foi empacotada em conjunto e armazenada na máquina remota 30' hospedando a pluralidade de arquivos aplicativos.

[00429] Em algumas modalidades, pelo menos um pré-requisito de execução requer a instalação de uma versão de um sistema operacional em um sistema executando pelo menos um arquivo aplicativo instalado. Em outras modalidades, pelo menos um pré-requisito de execução requer a instalação de uma versão do segundo aplicativo em um sistema executando pelo menos um arquivo aplicativo instalado. Ainda em outras modalidades, uma instrução está associada a pelo menos um arquivo aplicativo instalado, a instrução indicando um segundo arquivo aplicativo instalado para uso por um cliente falhando em satisfazer à pelo menos um pré-requisito de execução. Ainda em outras modalidades, uma instrução está associada a pelo menos um arquivo aplicativo instalado, a instrução indicando um segundo método de execução para execução de pelo menos um arquivo aplicativo instalado em um cliente falhando em satisfazer à pelo menos um pré-requisito de execução. Em uma destas modalidades, o método de execução está associado a pelo menos um

arquivo aplicativo instalado, o método de execução autorizando uma transmissão de uma pluralidade de arquivos aplicativos compreendendo o segundo aplicativo para uma máquina local para execução na máquina local. Em uma outra destas modalidades, uma avaliação de uma máquina local identifica pelo menos uma característica associada a pelo menos um arquivo aplicativo instalado não incluído na máquina local. Ainda em uma outra destas modalidades, uma autorização para execução da pluralidade de arquivos aplicativos é revogada. Ainda em uma outra destas modalidades, um segundo método de execução é provido para execução da pluralidade de arquivos aplicativos, o segundo método de execução permitindo a execução da pluralidade de arquivos aplicativos em uma máquina remota e a transmissão de dados de saída de aplicativo a partir da máquina remota para a máquina local.

[00430] Pelo menos um arquivo aplicativo instalado é armazenado em pelo menos um servidor (etapa 2112). Em algumas modalidades, o programa aplicativo instalado é executado no ambiente de isolamento antes do armazenamento de pelo menos um arquivo aplicativo instalado em pelo menos um servidor. Em uma destas modalidades, um arquivo aplicativo adicional é gerado em resposta à execução do programa aplicativo instalado. Em uma outra destas modalidades, um arquivo de dados é gerado. Ainda em uma outra destas modalidades, o programa aplicativo instalado requer uma informação para completar a instalação, a informação sendo requerida após um processo de instalação inicial. Ainda em uma outra destas modalidades, uma informação, tal como identificadores de produto de software, identificadores de licença ou outras credenciais, é requerida.

[00431] Em algumas modalidades, um identificador é provido identificando uma localização de pelo menos um arquivo aplicativo instalado em pelo menos um servidor. Em uma destas modalidades, o

identificador se conforma a uma Convenção de Denominação Universal (UNC). Em outras modalidades, pelo menos um arquivo aplicativo instalado é colocado em um arquivo de armazenamento de backup, tal como um arquivo .CAB. Em uma destas modalidades, uma pluralidade de arquivos aplicativos é armazenada em um arquivo de armazenamento de backup e o arquivo de armazenamento de backup é armazenado em pelo menos um servidor. Ainda em uma outra destas modalidades, pelo menos um arquivo aplicativo instalado é armazenado em múltiplos servidores. Ainda em outras modalidades, pelo menos um arquivo aplicativo é colocado em um diretório armazenando arquivos aplicativos.

[00432] Uma enumeração é gerada do segundo aplicativo, pelo menos um arquivo aplicativo instalado, uma localização de pelo menos um servidor e pelo menos um pré-requisito de execução (etapa 2114). Em algumas modalidades, a enumeração é armazenada em um arquivo. Em outras modalidades, a enumeração é armazenada em um arquivo de manifesto. Ainda em outras modalidades, a enumeração é armazenada em um arquivo XML.

[00433] Em uma modalidade, uma enumeração é gerada de múltiplos aplicativos, uma pluralidade de arquivos aplicativos instalados associados a cada um dos múltiplos aplicativos, e uma localização de pelo menos um servidor armazenando a pluralidade de arquivos aplicativos instalados. Em uma outra modalidade, uma enumeração é gerada incluindo uma associação entre o segundo aplicativo e uma pluralidade de arquivos aplicativos instalados. Ainda em uma outra modalidade, uma enumeração é gerada incluindo uma associação entre o segundo aplicativo e um arquivo comprimido contendo pelo menos um arquivo aplicativo instalado.

[00434] Com referência de volta à etapa 2106, quando uma ação de pelo menos uma chamada interceptada é executada sem um reboot

do sistema operacional, em algumas modalidades, uma instalação virtualizada e um ambiente de execução são providos, que removem a exigência de rebotar o sistema antes da execução de um aplicativo instalado.

[00435] Com referência, agora, à figura 24, um fluxograma descreve uma modalidade na qual a execução de um programa instalador requer um reboot de um sistema operacional em uma máquina local na qual o programa instalador é executado. Um instalador de aplicativo convencional copia arquivos em uma máquina remota em que o aplicativo estiver sendo instalado (etapa 2402). Em algumas modalidades, a cópia dos arquivos pode causar um reboot da máquina remota. O instalador de aplicativo tenta copiar pelo menos um dos arquivos para arquivos travados (etapa 2404). Em uma modalidade, um arquivo travado pode ser escrito apenas quando um sistema operacional for executado (ou "rebutado"). A opção `MOVE_FILE_DELAY_UNTIL_REBOOT` é regulada na `MoveFileEx()` Win32 API (etapa 2406). E o instalador de aplicativo chama uma função de parada/reboot de sistema (etapa 2408). Seguindo-se a um reboot, os arquivos travados originalmente então são instalados mediante um reboot (etapa 2410).

[00436] Com referência, agora, à figura 25, um diagrama de blocos descreve uma modalidade de uma máquina remota 30 na qual um mecanismo de formação de pacote instala um programa aplicativo. A máquina remota 30 inclui recursos de sistema 2502, APIs de sistema 2504 e um instalador de aplicativo 2506 usado para instalação de um aplicativo. A máquina remota 30 também inclui um mecanismo de enganchamento de função 2508, um módulo de processador pós-instalação 2510 e um ambiente de isolamento de aplicativo 2512. Em algumas modalidades, a instalação de um programa aplicativo em um ambiente de isolamento 2512 permite uma instalação sem um reboot

da máquina remota 30. Em uma destas modalidades, uma mudança feita em um recurso de sistema 2502 virtualizado em um ambiente de isolamento 2512 não muda um recurso de sistema 2502 correspondente na máquina remota 30. Uma vez que o recurso de sistema na máquina remota 30 não é mudado, um reboot da máquina para proteção do recurso de sistema de mudanças impróprias não é requerido.

[00437] Com referência, agora, à figura 25, e em maiores detalhes, os recursos de sistema 2502 podem incluir entradas de registro, DLLs de sistema e outros arquivos travados que o sistema operacional impede que sejam gravados enquanto a máquina remota 30 estiver em execução. As APIs de sistema 2504 incluem APIs usadas para se rebutar o sistema que são chamadas pelo instalador de aplicativo 2506 e enganchadas pelo mecanismo de enganchamento de função 2508 para se evitar o reboot da máquina remota 30.

[00438] O ambiente de isolamento de aplicativo 2512 provê um ambiente com uma visão de recursos de sistema operacional para um instalador de aplicativo 2506. Em uma modalidade, o ambiente de isolamento de aplicativo 2512 é um ambiente de isolamento 556. Em algumas modalidades, o ambiente de isolamento de aplicativo 2512 provê uma virtualização de recursos de sistema operacional, tal como o sistema de arquivo, registro e objetos denominados. Em uma modalidade, o instalador de aplicativo 2506 é executado no ambiente de isolamento de aplicativo 2512. Em uma modalidade, o instalador de aplicativo 2506 instala o programa aplicativo no ambiente de isolamento de aplicativo 2512. Ainda em uma outra modalidade, o instalador de aplicativo 2506 é executado fora do ambiente de isolamento de aplicativo 2512 e instala o programa aplicativo dentro do ambiente de isolamento de aplicativo 2512.

[00439] Em algumas modalidades, o ambiente de isolamento 2512

contorna a exigência para rebotar a máquina remota 30, quando o instalador de aplicativo 2506 instalar um aplicativo no ambiente de isolamento de aplicativo 2512. Em uma modalidade, o ambiente de isolamento de aplicativo 2512 intercepta uma requisição para copiar um arquivo aplicativo para um arquivo travado. Em uma outra modalidade, o ambiente de isolamento de aplicativo 2512 redireciona a requisição para copiar o arquivo aplicativo para um arquivo não travado. Ainda em uma outra modalidade, o ambiente de isolamento de aplicativo 2512 redireciona a requisição para copiar o arquivo aplicativo para um arquivo virtualizado. Ainda em uma outra modalidade, um redirecionamento da requisição para copiar o arquivo aplicativo permite a instalação de arquivos aplicativos sem a exigência de um reboot da máquina remota 30. Como um exemplo, se um instalador de aplicativo 2506 tentar gravar em um arquivo travado, tal como c:\windows\system32\mfc40.dll, o ambiente de isolamento de aplicativo 2512 intercepta a requisição e redireciona o arquivo para uma outra localização não travada. Esta capacidade de evitar arquivos travados significa que o arquivo pode ser instalado sem se ter que fazer uso de MoveFileEx() API e do indicador (flag) MOVE_FILE_DELAY_UNTIL_REBOOT. Esta capacidade remove a necessidade de um reboot da máquina remota 30.

[00440] Em uma modalidade, o mecanismo de enganchamento de função 2508 é um driver de filtro de sistema de arquivo 564. Em uma outra modalidade, um driver de filtro de sistema de arquivo 564 inclui o mecanismo de enganchamento de função 2508. Ainda em uma outra modalidade, o mecanismo de enganchamento de função 2508 intercepta requisições a partir do instalador de aplicativo 2506 para se recomençar a máquina remota 30. Em algumas modalidades, o ambiente de isolamento de aplicativo 2512 provê uma cópia de arquivos aplicativos nos arquivos não travados. Contudo, o ambiente

de isolamento de aplicativo 2512 não se dirige a uma requisição pelo instalador de aplicativo 2506 para um reboot da máquina remota 30. O mecanismo de enganchamento de função 2508 intercepta a requisição para reboot e responde ao instalador de aplicativo 2506.

[00441] O ambiente de isolamento de aplicativo 2512 permite uma cópia de arquivos aplicativos para arquivos não travados. Contudo, em algumas modalidades, outras ações são requeridas para a instalação de um aplicativo, e estas ações podem ocorrer quando do reboot. Impedir o reboot não impede a necessidade de se completarem estas ações do processo de instalação. O mecanismo de enganchamento de função 2508 pode prover uma funcionalidade para realização de uma ação associada a uma instalação de um aplicativo.

[00442] Por exemplo, durante a instalação de um aplicativo, entradas de registro, tal como HKLM\SYSTEM\CurrentControlSet\Control\Session_Manager\Pending-FileRenameOperations, podem ser escritas. Outros aplicativos podem instalar serviços ou drivers os quais precisam ser começados quando do boot de uma máquina. O módulo de processador pós-instalação 2510 identifica arquivos aplicativos que foram modificados durante uma instalação, e realiza as ações associadas aos arquivos aplicativos.

[00443] Com referência, agora, à figura 26, um fluxograma descreve uma modalidade das etapas seguidas para a instalação de um aplicativo em um ambiente de isolamento de aplicativo 2512. O ambiente de isolamento de aplicativo 2512 provê uma vista virtualizada do sistema operacional do servidor para o instalador de aplicativo (etapa 2602). As APIs no servidor relativas a reboots de sistema e paradas são enganchadas (etapa 2604) para se evitar que o instalador de aplicativo 2506 cause um reboot. O instalador de aplicativo 2506 requisita operações de cópia de arquivo para arquivos

travados, a requisição sendo interceptada e redirecionada para localizações não conflitantes (etapa 2606). Quando o instalador de aplicativo 2506 tenta rebutar ao chamar uma API de sistema, a requisição é interceptada e o reboot é impedido (etapa 2608). O módulo de processador pós-instalação 2510 realiza ações que comumente ocorrem após um reboot (etapa 2610) e o aplicativo então pode ser executado no ambiente de isolamento de aplicativo 2512, sem um reboot de uma máquina remota 30 (etapa 2612).

[00444] Em algumas modalidades, seguindo-se à instalação do programa aplicativo no ambiente de isolamento de aplicativo 2512, um mecanismo de formação de pacote identifica uma pluralidade de arquivos aplicativos criados ou modificados durante uma instalação de um programa aplicativo. Em uma destas modalidades, a pluralidade de arquivos aplicativos é armazenada em uma máquina remota. Em uma outra destas modalidades, uma máquina local recuperando a pluralidade de arquivos aplicativos pode executar o programa aplicativo.

[00445] Em algumas modalidades, o mecanismo de pacote 530 é executado em uma máquina remota incluindo um ambiente de isolamento 532 e um driver de filtro de sistema de arquivo 534 e instala um programa aplicativo no ambiente de isolamento 532. Em uma destas modalidades, a máquina remota é referida como uma "máquina limpa" ou uma "máquina em estágio". Em uma outra destas modalidades, o ambiente de isolamento 532 inclui um escopo de isolamento de aplicativo provendo uma instância virtualizada modificável de um recurso nativo provido por um sistema operacional na máquina limpa. Ainda em uma outra destas modalidades, o ambiente de isolamento 532 inclui um escopo de isolamento de sistema provendo uma vista apenas de leitura do recurso nativo. Ainda em uma outra destas modalidades, a vista apenas de leitura do

recurso nativo compreende um instantâneo de tela de um sistema de arquivo e registro residente na máquina limpa.

[00446] Em uma modalidade, um redirecionador intercepta uma requisição por uma mudança para o recurso nativo. Em algumas modalidades, o redirecionador é um driver de filtro de sistema de arquivo 534. Em uma outra modalidade, um programa instalador executado pelo mecanismo de pacote 530 faz a requisição pela mudança. Ainda em uma outra modalidade, a mudança para o recurso nativo é requerida para a instalação de um programa aplicativo na máquina limpa. Ainda em uma outra modalidade, o redirecionador redireciona a requisição para o ambiente de isolamento 532.

[00447] Em algumas modalidades, um redirecionamento de requisições para mudar recursos nativos para o ambiente de isolamento 532 resulta em um isolamento de mudanças associadas à instalação de um programa aplicativo. Em outras modalidades, as requisições para mudança de recursos nativos são gravadas e armazenadas em um elemento de armazenamento. Em uma destas modalidades, todas as mudanças associadas a uma instalação de um programa aplicativo residem no elemento de armazenamento. Em uma outra destas modalidades, uma máquina local 552 recuperando o conteúdo do elemento de armazenamento e implementando as mudanças para recursos nativos residentes em um ambiente de isolamento 556 na máquina local 552 resultam na instalação do programa aplicativo na máquina local 552.

[00448] Em algumas modalidades, uma análise de pré-abertura da máquina local 10 pode ser requerida. Em uma destas modalidades, a máquina local 10 verifica que pelo menos uma característica é incluída na máquina local 10. Em uma outra destas modalidades, pelo menos uma característica é adicionada à máquina local 10 após a análise de pré-abertura determinar que a máquina local 10 carece de pelo menos

uma característica. Em ainda uma outra destas modalidades, pelo menos uma característica é incluída em uma máquina remota hospedando um programa aplicativo e uma falha da máquina local em incluir pelo menos uma característica impedirá a execução do programa aplicativo. Ainda em uma outra modalidade, o programa aplicativo requer a existência de pelo menos uma característica na máquina local para execução.

[00449] Em algumas modalidades, o mecanismo de formação de pacote permite a identificação de pelo menos uma característica para uso em uma análise de pré-abertura na máquina local. Em outras modalidades, o mecanismo de formação de pacote permite a associação de pelo menos uma característica a um programa aplicativo disponível para execução na máquina local. Ainda em outras modalidades, o mecanismo de formação de pacote permite uma associação de um script executável a um programa aplicativo, a máquina local executando o script executável para completar a análise de pré-abertura. Ainda em outras modalidades, pelo menos uma característica é requerida para existir na máquina local, após a execução do programa aplicativo.

[00450] O mecanismo de formação de pacote pode prover uma funcionalidade para assinalar uma pluralidade de arquivos aplicativos. Em uma modalidade, a sinalização da pluralidade de arquivos aplicativos permite que uma máquina local verifique a integridade da pluralidade de arquivos aplicativos. Em uma outra modalidade, a sinalização da pluralidade de arquivos aplicativos impede uma máquina local de executar um programa aplicativo corrompido. Em algumas modalidades, uma soma de verificação criptográfica, tal como MD4 hash, MD5 hash, ou SHA-1 hash, de um arquivo na pluralidade de arquivos aplicativos é computado.

[00451] Em outras modalidades, uma soma de verificação

criptográfica de todo arquivo na pluralidade de arquivos aplicativos é computada. Em uma destas modalidades, a soma de verificação criptográfica é armazenada em um segundo arquivo. Em uma outra destas modalidades, o segundo arquivo está associado à pluralidade de arquivos aplicativos. Em algumas modalidades, o segundo arquivo é adicionado à pluralidade de arquivos aplicativos. Em outras modalidades, o segundo arquivo é sinalizado usando-se um certificado, tal como um certificado X.509. Em ainda outras modalidades, uma máquina local recuperando a pluralidade de arquivos aplicativos verifica a assinatura usando uma porção pública do certificado. Ainda em outras modalidades, a máquina local recebe a porção pública do certificado e uma identificação de uma lista de confiança de certificado para verificação da assinatura. Em uma destas modalidades, a máquina local recebe uma chave de registro contendo a identificação de uma lista de confiança de certificado.

[00452] Em uma modalidade, o mecanismo de formação de pacote provê uma funcionalidade para personalização de um ambiente de isolamento. Em uma outra modalidade, o mecanismo de formação de pacote provê uma funcionalidade para a geração de um arquivo armazenando uma definição de um ambiente de isolamento. Ainda em uma outra modalidade, o mecanismo de formação de pacote inclui o arquivo com a pluralidade de arquivos aplicativos compreendendo um programa aplicativo. Ainda em uma outra modalidade, uma máquina local recebe o arquivo com uma informação de acesso a partir de uma máquina remota.

[00453] Em algumas modalidades, uma pluralidade de arquivos aplicativos é armazenada em um arquivo de armazenamento de backup. Em uma destas modalidades, o arquivo de armazenamento de backup está em um formato de arquivo CAB. Em uma outra destas modalidades, o formato de arquivo de armazenamento de backup não

provê suporte para uma especificação por um programa aplicativo de nomes de arquivo curtos de um arquivo. Ainda em uma outra destas modalidades, um sistema operacional, tal como WINDOWS 2000, não pode prover suporte para uma especificação por um programa aplicativo de nomes de arquivo curtos de um arquivo. Em outras modalidades, um sistema operacional, tal como WINDOWS XP, provê suporte para especificação por um programa aplicativo de um nome de arquivo curto de um arquivo. Em uma destas modalidades, uma requisição para execução do arquivo deve incluir o nome de arquivo curto correto do arquivo.

[00454] Em uma modalidade, um mapeamento pode ser gerado para associação de um nome de arquivo longo de um arquivo na pluralidade de arquivos aplicativos a um nome de arquivo curto do arquivo. Em uma outra modalidade, o mapeamento é armazenado em um arquivo na pluralidade de arquivos aplicativos. Em ainda uma outra modalidade, um arquivo tem um nome de arquivo curto apenas se o nome de arquivo longo do arquivo for mais longo do que doze caracteres. Em algumas modalidades, o nome de arquivo curto é um nome de arquivo virtual associado ao arquivo. Em uma destas modalidades, o arquivo é transmitido para a máquina local 10 para execução, onde ele é armazenado com um nome de arquivo longo. Em uma outra destas modalidades, um arquivo aplicativo na máquina local 10 requisita uma execução do arquivo usando o nome de arquivo curto. Em ainda uma outra destas modalidades, o mapeamento permite a execução do arquivo embora a requisição para execução do arquivo não usasse o nome do arquivo na máquina local (o nome de arquivo longo).

[00455] Em algumas modalidades, o mecanismo de pacote 530 gera o mapeamento. Em uma destas modalidades, o mecanismo de pacote 530 seleciona um nome de arquivo curto para um arquivo

tendo um nome de arquivo longo. Em uma outra destas modalidades, um sistema operacional na máquina remota 30' na qual o mecanismo de pacote 530 está em execução seleciona um nome de arquivo curto para um arquivo tendo um nome de arquivo longo. Ainda em uma outra destas modalidades, um nome de arquivo curto único é selecionado, que não está em conflito com um segundo nome de arquivo curto na máquina remota 30'. Em ainda uma outra destas modalidades, o programa instalador executado pelo mecanismo de pacote 530 gera um arquivo incluindo um mapeamento entre um nome de arquivo longo com um nome de arquivo curto. Em outras modalidades, o mapeamento é transmitido para uma máquina local 10 recuperando o arquivo. Em uma outra destas modalidades, a máquina local 10 se refere ao arquivo quando executando o arquivo.

[00456] Os exemplos ilustrativos a seguir mostram como os métodos e sistemas discutidos acima podem ser usados para a seleção, a transmissão para uma máquina local e a execução na máquina local de uma pluralidade de arquivos compreendendo um programa aplicativo. Estes exemplos têm por significado ilustrarem e não limitarem a invenção.

EXEMPLO 1

[00457] Em uma modalidade, um usuário da máquina local 10 requisita acesso a um programa aplicativo, tal como um programa de processamento de texto, um aplicativo de navegação na web, ou um programa de planilha, identificado em uma enumeração de programas aplicativos. Em um exemplo desta modalidade, a máquina local 10 executa um aplicativo de vizinhança de programa que recebe a partir de uma máquina remota 30 uma enumeração de aplicativos disponíveis para a máquina local 10. Em um outro exemplo desta modalidade, a máquina local 10 se comunica com um servidor da web, tal como a máquina remota 30''' para receber a enumeração de aplicativos. O

usuário da máquina local 10 pode requisitar acesso a um programa aplicativo enumerado pela seleção de uma descrição gráfica representando o programa aplicativo enumerado. O usuário da máquina local 10 pode requisitar acesso a um programa aplicativo não previamente instalado na máquina local 10.

[00458] A máquina local 10 transmite a requisição para acesso ao programa aplicativo para uma máquina remota 30. A máquina local 10 recebe uma identificação de uma máquina remota 30” provendo acesso a uma pluralidade de arquivos aplicativos compreendendo o programa aplicativo.

A máquina local 10 identifica pelo menos uma característica requerida para execução do programa aplicativo. Em um exemplo desta modalidade, a máquina local 10 recebe pelo menos uma característica com a identificação da máquina remota 30” transmitida por artigo absorvente máquina local 10 pela máquina remota 30. Em um outro exemplo desta modalidade, a máquina local 10 recupera pelo menos uma característica a partir da máquina remota 30”, após receber a identificação da máquina remota 30”. Pode ser requerido que a máquina local 10 compreenda pelo menos uma característica antes de receber autorização para recuperar a pluralidade de arquivos aplicativos. Alternativamente, pode ser requerido que a máquina local 10 compreenda pelo menos uma característica antes da execução da pluralidade de arquivos aplicativos. Em um exemplo desta modalidade, pode ser requerido que a máquina local 10 compreenda pelo menos uma característica por toda a execução da pluralidade de arquivos aplicativos.

[00459] Mediante uma verificação pela máquina local 10 que a máquina local 10 inclui pelo menos uma característica, a máquina local 10 recupera pelo menos um arquivo aplicativo na pluralidade de arquivos aplicativos e executa o arquivo aplicativo recuperado para

execução do programa aplicativo.

EXEMPLO 2

[00460] Uma máquina remota 30 recebe uma requisição para acessar um programa aplicativo a partir de uma máquina local 10. A máquina remota 30 autentica a máquina local 10. Em um exemplo desta modalidade, a máquina remota 30 requisita credenciais, tais como um nome de usuário e uma senha, a partir da máquina local 10. Em um outro exemplo desta modalidade, a máquina remota 30 transmite um agente de coleta 404 para a máquina local 10. O agente de coleta 404 acumula uma informação sobre a máquina local 10 e transmite a informação para a máquina remota 30 para uso na autenticação da máquina local 10. Ainda em um outro exemplo desta modalidade, a máquina remota 30 provê uma informação sobre a máquina local 10 para um agente de política 406 para autenticação da máquina local 10. A máquina remota 30 pode compreender o agente de política 406. Alternativamente, a máquina remota 30 pode estar em comunicação com uma máquina remota 30' compreendendo o agente de política 406.

[00461] A máquina remota 30 seleciona um método de execução do programa aplicativo. A máquina remota 30 pode fazer a seleção em resposta à autenticação da máquina local 10. Em um exemplo desta modalidade, a máquina remota 30 aplica uma política a uma informação acumulada sobre a máquina local 10. Em um outro exemplo desta modalidade, a máquina remota 30 faz uma seleção em resposta a uma política aplicada ao programa aplicativo. Ainda em um outro exemplo desta modalidade, a máquina remota 30 faz a seleção em resposta a uma política aplicada a um tipo de arquivo associado ao programa aplicativo. A máquina remota 30 pode consultar um arquivo para fazer a seleção do método de execução do programa aplicativo.

[00462] A máquina remota 30 pode selecionar um método de

execução do programa aplicativo permitindo que a máquina local 10 receba dados de saída de aplicativo gerados pela execução do programa aplicativo em uma máquina remota 30'. A máquina remota 30 pode selecionar um método de execução do programa aplicativo permitindo que a máquina local 10 execute o programa aplicativo localmente após recuperar uma pluralidade de arquivos aplicativos compreendendo o programa aplicativo.

[00463] Em uma modalidade, a máquina remota 30 seleciona um método de execução do programa aplicativo permitindo que a máquina local 10 execute o programa aplicativo localmente, enquanto recupera uma pluralidade de arquivos aplicativos compreendendo o programa aplicativo através de uma sessão de transmissão de aplicativo. Em um exemplo desta modalidade, a máquina local 10 estabelece uma sessão de transmissão de aplicativo com uma máquina remota hospedando uma pluralidade de arquivos aplicativos, a máquina local 10 inicia a recuperação da pluralidade de arquivos aplicativos através da sessão de transmissão de aplicativo, e a máquina local 10 executa um primeiro arquivo aplicativo recuperado na pluralidade de arquivos aplicativos enquanto recupera um segundo arquivo aplicativo na pluralidade de arquivos aplicativos. Em um outro exemplo desta modalidade, a máquina local 10 executa um primeiro arquivo aplicativo na pluralidade de arquivos aplicativos e recupera um segundo arquivo aplicativo na pluralidade de aplicativos mediante o recebimento de uma requisição a partir do primeiro arquivo aplicativo ranhura acesso ao segundo arquivo aplicativo.

[00464] Para modalidades nas quais o método selecionado de execução permite que a máquina local 10 recupere pelo menos um arquivo aplicativo em uma pluralidade de arquivos aplicativos compreendendo um programa aplicativo, a máquina remota 30 identifica uma máquina remota 30'' hospedando o programa aplicativo

disponível para acessos pela máquina local 10. A máquina remota 30'' hospeda uma pluralidade de arquivos aplicativos compreendendo o programa aplicativo. A máquina remota 30'' pode hospedar múltiplas pluralidades de arquivos aplicativos compreendendo vários programas aplicativos. Em um exemplo desta modalidade, a máquina remota 30'' hospeda uma pluralidade de arquivos aplicativos para cada uma de várias versões diferentes de um programa aplicativo.

[00465] A máquina remota 30'' hospeda um arquivo associando uma pluralidade de arquivos aplicativos compreendendo um programa aplicativo em particular a uma descrição do programa aplicativo. O arquivo também pode identificar um ou mais pré-requisitos de execução para serem identificados em uma máquina, antes da transmissão da pluralidade de arquivos aplicativos para a máquina. O arquivo ainda pode incluir uma identificação de uma localização em uma rede da máquina remota 30''. Em um exemplo desta modalidade, a máquina remota 30 consulta o arquivo para a identificação da localização na rede da máquina remota 30''.

[00466] A máquina remota 30 seleciona uma máquina remota 30''. A máquina remota 30 pode selecionar uma máquina remota 30'' tendo uma localização em uma rede acessível para a máquina local 10. A máquina remota 30 pode selecionar uma máquina remota 30'' hospedando uma versão do programa aplicativo compatível com a máquina local 10. A máquina remota 30 transmite uma identificação do método selecionado de execução do programa aplicativo e uma identificação da máquina remota 30'' para a máquina local 10, em resposta ao recebimento da requisição para acesso para o programa aplicativo. A máquina remota 30 também pode transmitir o arquivo para a máquina local 10.

EXEMPLO 3

[00467] Em uma modalidade, a máquina local 10 recebe uma

identificação de um método de execução selecionado de um programa aplicativo e uma identificação de uma máquina remota 30'' provendo acesso a uma pluralidade de arquivos aplicativos compreendendo o programa aplicativo. A máquina local 10 verifica uma autorização de acesso ao programa aplicativo. Em um exemplo desta modalidade, a máquina local 10 realiza uma análise de pré-abertura de si mesma. A máquina local 10 identifica pelo menos uma característica e verifica a existência de pelo menos uma característica na máquina local 10. Pelo menos uma característica pode ser um pré-requisito para manutenção de uma autorização para acesso e execução do programa aplicativo. A verificação da existência de pelo menos uma característica na máquina local 10 pode garantir uma compatibilidade entre características da máquina local 10 e as exigências de sistema do programa aplicativo, e pode garantir, adicionalmente, uma conformidade com políticas de segurança ou acordos de licenciamento.

[00468] Mediante uma conclusão bem-sucedida de uma análise de pré-abertura, a máquina local 10 estabelece uma sessão de transmissão de aplicativo com a máquina remota 30'' provendo acesso à pluralidade de arquivos aplicativos. A sessão de transmissão de aplicativo pode ser qualquer conexão pela qual a máquina local 10 pode requisitar e receber um arquivo na pluralidade de arquivos aplicativos. O estabelecimento da sessão de transmissão de aplicativo pode permitir que a máquina local 10 execute um primeiro arquivo aplicativo na pluralidade de arquivos aplicativos antes da recuperação de todos os arquivos na pluralidade de arquivos aplicativos. A máquina local 10 pode iniciar uma execução do programa aplicativo enquanto continua a recuperação de arquivos aplicativos adicionais na pluralidade de arquivos aplicativos. Alternativamente, a máquina local 10 pode recuperar a pluralidade de aplicativos em um arquivo de

armazenamento de backup e executar um primeiro arquivo aplicativo extraído enquanto extrai um segundo arquivo aplicativo a partir do arquivo de armazenamento de backup.

EXEMPLO 4

[00469] Em uma modalidade, um cliente de transmissão de aplicativo 552 em uma máquina local 10 recupera uma pluralidade de arquivos aplicativos a partir da máquina remota 30. O cliente de transmissão de aplicativo inclui um serviço de transmissão 554, um ambiente de isolamento 556 e um driver de filtro de sistema de arquivo 564. O serviço de transmissão 554 estabelece uma sessão de transmissão de aplicativo com a máquina remota 30 para requisitar e recuperar a pluralidade de arquivos aplicativos. O serviço de transmissão 554 executa os arquivos aplicativos no ambiente de isolamento 556. O driver de filtro de sistema de arquivo 564 permite a execução de arquivos aplicativos no ambiente de isolamento 556 pela interceptação de requisições a partir dos arquivos aplicativos de execução e pelo redirecionamento das requisições para o ambiente de isolamento 556.

[00470] Em um exemplo desta modalidade, o serviço de transmissão 554 recupera um arquivo de armazenamento de backup incluindo a pluralidade de arquivos aplicativos compreendendo um programa aplicativo. O serviço de transmissão 554 extrai a partir do arquivo de armazenamento de backup um primeiro arquivo aplicativo a partir da pluralidade de arquivos aplicativos. O primeiro arquivo aplicativo pode ser um arquivo executável. O serviço de transmissão 554 pode executar o primeiro arquivo aplicativo no ambiente de isolamento 556. A execução do primeiro arquivo aplicativo pode iniciar uma execução do programa aplicativo.

[00471] Em uma outra modalidade, um primeiro arquivo aplicativo em execução no ambiente de isolamento 556 requisita a partir da

máquina local 10 uma enumeração da pluralidade de arquivos aplicativos. O driver de filtro de sistema de arquivo 564 intercepta a requisição para a enumeração e redireciona a requisição para o serviço de transmissão 554. Em modalidades em que o serviço de transmissão 554 recuperou a pluralidade de arquivos aplicativos, o serviço de transmissão 554 pode gerar uma enumeração da pluralidade de arquivos aplicativos. Em modalidades em que o serviço de transmissão 554 recuperou um arquivo de armazenamento de backup incluindo a pluralidade de arquivos aplicativos, o serviço de transmissão 554 pode gerar a enumeração da pluralidade de arquivos aplicativos em resposta a uma enumeração incluída no arquivo de armazenamento de backup recuperado. Em outras modalidades, o serviço de transmissão 554 recupera apenas a núcleo da pluralidade de arquivos aplicativos enquanto pelo menos um arquivo aplicativo na pluralidade de arquivos aplicativos reside em uma máquina remota 30 e ainda não foi recuperado para a máquina local 10 pelo serviço de transmissão 554. Nestas modalidades, o serviço de transmissão 554 pode gerar uma enumeração da pluralidade de arquivos aplicativos em resposta à enumeração recuperada. Em um exemplo destas modalidades, o serviço de transmissão 554 indica para o primeiro arquivo aplicativo que a pluralidade de arquivos aplicativos reside na máquina local 10, embora apenas a enumeração resida na máquina local 10.

EXEMPLO 5

[00472] Em uma modalidade, um primeiro arquivo aplicativo em execução no ambiente de isolamento 556 requisita a partir da máquina local 10 acesso a um arquivo identificado pela enumeração da pluralidade de arquivos aplicativos. Se o arquivo requisitado residir em um escopo de usuário no ambiente de isolamento 556 acessível para o primeiro arquivo aplicativo, o primeiro arquivo aplicativo acessará o

arquivo requisitado.

[00473] Se o arquivo requisitado não residir no escopo de usuário ou no ambiente de isolamento 556, o driver de filtro de sistema de arquivo 564 interceptará a requisição e redirecionará a requisição para o serviço de transmissão 554. Se o arquivo requisitado for um arquivo no arquivo de armazenamento de backup contendo a pluralidade de arquivos aplicativos, o serviço de transmissão 554 extrairá o arquivo requisitado e armazenará o arquivo requisitado na máquina local 10. O serviço de transmissão 554 pode armazenar o arquivo no ambiente de isolamento 556. A requisição pelo arquivo é satisfeita quando o arquivo estiver armazenado no ambiente de isolamento 556.

[00474] Se o arquivo requisitado não residir no ambiente de isolamento 556 ou no arquivo de armazenamento de backup incluindo a pluralidade de arquivos aplicativos, o serviço de transmissão 554 requisitará o arquivo a partir da máquina remota 30. O serviço de transmissão 554 pode receber o arquivo a partir da máquina remota 30 através de uma sessão de transmissão de aplicativo. O serviço de transmissão 554 armazena o arquivo recebido no ambiente de isolamento 556. A requisição para o arquivo é satisfeita quando o arquivo estiver armazenado no ambiente de isolamento 556.

[00475] Em um exemplo desta modalidade, um segundo arquivo aplicativo é executado em um segundo escopo no ambiente de isolamento 556. O segundo arquivo aplicativo requisita acesso ao arquivo originalmente requisitado pelo primeiro arquivo aplicativo. Se uma cópia do arquivo requisitado não residir no segundo escopo de usuário, a cópia do arquivo requisitado armazenado no ambiente de isolamento 556 será usada para satisfazer à requisição pelo arquivo aplicativo.

EXEMPLO 6

[00476] Em uma modalidade, uma máquina local 10 recebe de uma

máquina remota 30 uma identificação de um método selecionado de execução de um programa aplicativo e uma identificação de uma máquina remota 30' provendo acesso a uma pluralidade de arquivos aplicativos compreendendo o programa aplicativo. A máquina local 10 completa de forma bem-sucedida uma análise da máquina local 10. A máquina local 10 recebe uma licença da máquina remota 30 autorizando a execução do programa aplicativo. Em um exemplo desta modalidade, a licença requer que a máquina local 10 transmita mensagens de batimento cardíaco para um servidor de gerenciamento de sessão 562 para manutenção de uma autorização para execução do programa aplicativo. As mensagens de batimento cardíaco podem incluir mensagens indicando uma iniciação de execução de um programa aplicativo, uma terminação de execução de um programa aplicativo, e mensagens enviadas em uma base periódica por toda a execução do programa aplicativo. As mensagens de batimento cardíaco também podem incluir mensagens sobre o status da máquina local 10, tal como quando a máquina local 10 se conecta a uma rede ou quando a máquina local 10 termina uma conexão com uma rede. Em um outro exemplo desta modalidade, a licença especifica um período de tempo predeterminado durante o qual a máquina local 10 tem autorização para executar o programa aplicativo.

[00477] A máquina local 10 estabelece uma sessão de transmissão de aplicativo com a máquina remota 30' e recupera pelo menos um dos arquivos aplicativos na pluralidade de arquivos aplicativos. Durante a execução de pelo menos um arquivo aplicativo, em modalidades em que a licença recebida requer uma transmissão de mensagens de batimento cardíaco, a máquina local 10 envia as mensagens de batimento cardíaco para o servidor de gerenciamento de sessão 562 para manutenção da autorização para execução de pelo menos um arquivo aplicativo.

EXEMPLO 7

[00478] Em uma modalidade, a máquina local 10 recebe uma identificação de um método selecionado de execução de um programa aplicativo e uma identificação de uma máquina remota 30' provendo acesso a uma pluralidade de arquivos aplicativos compreendendo o programa aplicativo. A máquina local 10 completa de forma bem-sucedida uma análise de pré-abertura da máquina local 10. A máquina local 10 recebe uma licença especificando um período de tempo predeterminado durante o qual a máquina local 10 tem uma autorização para execução do programa aplicativo.

[00479] A máquina local 10 estabelece uma sessão de transmissão de aplicativo com a máquina remota 30' e recupera pelo menos um dos arquivos aplicativos na pluralidade de arquivos aplicativos. Em um exemplo desta modalidade, a máquina local 10 recupera um subconjunto da pluralidade de arquivos aplicativos, o subconjunto compreendendo cada arquivo necessário para execução do programa aplicativo, quando a máquina local 10 não estiver conectada a uma rede. A máquina local 10 armazena o subconjunto em um cache na máquina local 10.

[00480] Em um ponto no tempo dentro do período de tempo predeterminado, a máquina local 10 é desconectada de uma rede e recebe de um usuário da máquina local 10 uma requisição para acesso ao programa aplicativo. Em um exemplo desta modalidade, a máquina local 10 é um dispositivo, tal como um laptop, e o usuário da máquina local 10 está em um ambiente que proíbe conexão a redes, tal como em um avião. Mediante o recebimento da requisição do usuário, a máquina local 10 pode recuperar a partir do cache um arquivo aplicativo a partir da pluralidade de arquivos aplicativos e executar o programa aplicativo.

EXEMPLO 8

[00481] Em uma outra modalidade, a máquina local 10 recebe uma identificação de um método selecionado de execução de um programa aplicativo e uma identificação de uma máquina remota 30' provendo acesso a uma pluralidade de arquivos aplicativos compreendendo o programa aplicativo. A máquina local 10 pode receber uma identificação de um primeiro agente de cliente residente na máquina local 10 para execução para recuperação da pluralidade de arquivos aplicativos, tal como um cliente de transmissão de aplicativo.

[00482] Em um exemplo desta modalidade, a máquina local 10 falha em completar de forma bem-sucedida uma análise de pré-abertura de si mesma. A máquina local 10 pode carecer de uma característica requerida para compatibilidade com uma exibição do programa aplicativo, tal como um driver de dispositivo em particular ou sistema operacional. A máquina local 10 pode carecer de uma característica requerida para conformidade com uma política de segurança, por exemplo, participação em um Diretório Ativo ou autorização para acesso a uma rede privada. A máquina local 10 pode ser um tipo de máquina incompatível com uma exigência do programa aplicativo, tal como um assistente digital pessoal tentando acessar um programa aplicativo computacionalmente intenso, ou uma máquina pública em um quiosque tentando executar um aplicativo seguro hospedado por uma máquina remota em uma rede privada.

[00483] A máquina local 10 faz uma determinação de não recuperar a pluralidade de arquivos aplicativos através da sessão de transmissão de aplicativo, em resposta à determinação que a máquina local 10 carece de pelo menos uma característica requerida para acesso ao programa aplicativo. A máquina local 10 executa um segundo agente de cliente residente na máquina local 10, ao invés de executar o primeiro agente de cliente identificado. Em um exemplo desta modalidade, a máquina local 10 recebe uma identificação do segundo agente de cliente

a executar, no caso de falha em completar de forma bem-sucedida uma análise de pré-abertura. A máquina local 10 requisita uma execução do programa aplicativo em uma máquina remota 30". O segundo agente de cliente recebe dados de saída de aplicativo gerados pela execução do programa aplicativo na máquina remota 30". O segundo agente de cliente exibe os dados de saída de aplicativo na máquina local 10.

EXEMPLO 9

[00484] Em uma modalidade, um administrador de uma rede provê acesso a um programa aplicativo para usuários de máquinas locais 10. O administrador executa um aplicativo em uma máquina remota 30' para a geração de uma pluralidade de arquivos aplicativos compreendendo o programa aplicativo. O aplicativo pode incluir uma interface gráfica de usuário. O administrador pode usar a interface gráfica de usuário para a identificação do programa aplicativo e de um programa instalador associado ao programa aplicativo, definir políticas a serem aplicadas na autorização de acesso ao programa aplicativo, e especificar características sobre o tipo de acesso provido incluindo exigências a serem satisfeitas por uma máquina local 10 tentando acessar ou executar o programa aplicativo. O administrador pode identificar um programa instalador instalando um programa aplicativo inteiro ou uma porção de um programa aplicativo, tal como uma atualização ou um "patch".

[00485] Em um exemplo desta modalidade, uma máquina remota 30 inclui um mecanismo de pacote 530. O mecanismo de pacote 530 executa o programa instalador em um ambiente de isolamento 532 na máquina remota 30. A execução do programa instalador resulta na instalação, no ambiente de isolamento 532, de pelo menos um arquivo aplicativo associado ao programa aplicativo. A máquina remota 30 pode incluir um driver de filtro de sistema de arquivo 534, o qual

assegura a instalação do arquivo aplicativo no ambiente de isolamento 532 pela interceptação de uma requisição pelo programa instalador para instalação do arquivo aplicativo na máquina local 10, e redirecionamento da requisição para o ambiente de isolamento 532. O mecanismo de pacote 530 pode usar o driver de filtro de sistema de arquivo 534 para manutenção de um registro de cada arquivo aplicativo instalado no ambiente de isolamento 532.

[00486] O programa instalador pode instalar uma pluralidade de arquivos aplicativos no ambiente de isolamento 532. O mecanismo de pacote 530 gera um arquivo incluindo uma enumeração de arquivos aplicativos na pluralidade de arquivos aplicativos. O arquivo pode incluir uma informação associada à pluralidade de arquivos aplicativos, tal como o tipo de programa aplicativo que a pluralidade de arquivos aplicativos compreende, a versão do programa aplicativo, os pré-requisitos de execução associados ao programa aplicativo e exigências de política, tal como um método de execução requerido para um programa aplicativo em particular. O mecanismo de pacote 530 armazena em uma máquina remota 30' a pluralidade de arquivos aplicativos e o arquivo.

[00487] Em uma modalidade, o administrador da rede identifica um programa aplicativo compreendendo uma versão atualizada de um programa aplicativo existente ou arquivo aplicativo em uma pluralidade de arquivos aplicativos compreendendo um programa aplicativo.

C. SISTEMAS E MÉTODOS PARA ACELERAÇÃO DE COMUNICAÇÕES DE CLIENTE – SERVIDOR

[00488] Uma modalidade da presente invenção é dirigida a sistemas e métodos para a aceleração de comunicações de cliente – servidor. Estes sistemas e métodos podem ser usados sozinhos ou em consonância, e podem ser usados em conjunto com qualquer um dos sistemas e métodos para a entrega de um ambiente de computação

discutido acima. Em particular, quatro categorias de técnicas de aceleração serão discutidas.

1. Armazenamento em Cache de Objetos Gerados Dinamicamente: em algumas modalidades, as comunicações de cliente – servidor são aceleradas por uma aplicação 1250 realizando um armazenamento em cache de objetos gerados dinamicamente em uma rede de comunicação de dados.

2. Agrupamento de Conexão: em algumas modalidades, as comunicações de cliente – servidor são aceleradas por uma aplicação 1250 executando técnicas de agrupamento de conexão.

3. Armazenamento em Cache Integrado: em uma outra modalidade, as comunicações de cliente – servidor são aceleradas por uma aplicação 1250 executando um armazenamento em cache integrado com uma pluralidade de técnicas de aceleração.

4. Aceleração de Lado de Cliente: ainda em uma outra modalidade, as comunicações de cliente – servidor são aceleradas por um programa em execução em um cliente 10 realizando uma ou mais técnicas de aceleração.

1. Armazenamento em Cache de Objetos Gerados Dinamicamente

[00489] Conforme será descrito em maiores detalhes aqui, em uma modalidade, uma aplicação 1250 pode integrar uma funcionalidade de armazenamento em cache no nível de kernel do sistema operacional com uma ou mais outras tarefas de processamento, incluindo, mas não limitando, descriptação, descompressão ou autenticação e/ou autorização. Uma arquitetura de exemplo como essa é descrita aqui de acordo com a figura 27, mas outras arquiteturas podem ser usadas na prática das operações descritas aqui.

[00490] A figura 27 ilustra uma arquitetura de exemplo 3200 de uma aplicação 1250. Conforme citado acima, a arquitetura 3200 é provida a título de ilustração apenas e não é pretendida para ser limitante.

Conforme mostrado na figura 2, a arquitetura 3200 de exemplo consiste em uma camada de hardware 3206 e uma camada de software dividida em um espaço de usuário 3202 e um espaço de kernel 3204.

[00491] A camada de hardware 3206 provê os elementos de hardware nos quais os programas e serviços no espaço de kernel 3204 e no espaço de usuário 3202 são executados. A camada de hardware 3206 também provê as estruturas e os elementos os quais permitem que os programas e serviços no espaço de kernel 3204 e no espaço de usuário 3202 comuniquem dados interna e externamente com respeito à aplicação 1250. Conforme mostrado na figura 27, a camada de hardware 3206 inclui uma unidade de processamento 3262 para execução de programas de software e serviços, uma memória 3264 para armazenamento de software e dados, portas de rede 3266 para transmissão e recepção de dados por uma rede, e um processador de encriptação 3260 para a realização de funções relacionadas ao processamento de Camada de Soquetes Segura de dados transmitidos e recebidos pela rede. Em algumas modalidades, a unidade de processamento central 3262 pode realizar as funções do processador de encriptação 3260 em um processador único. Adicionalmente, a camada de hardware 3206 pode compreender múltiplos processadores para cada um dentre a unidade de processamento 3262 e o processador de encriptação 3260. Embora a camada de hardware 3206 da aplicação 1250 geralmente seja ilustrada com um processador de encriptação 3260, o processador 3260 pode ser um processador para a realização de funções relacionadas a qualquer protocolo de encriptação, tal como um protocolo de Camada de Soquete de Segurança (SSL) ou de Segurança de Camada de Transporte (TLS). Em algumas modalidades, o processador 3260 pode ser um processador de

finalidade geral (GPP), e em modalidades adicionais pode ter instruções executáveis para a realização do processamento de qualquer protocolo relacionado à segurança.

[00492] Embora a camada de hardware 3206 de aplicação 1250 seja ilustrada com certos elementos na figura 27, as porções de hardware ou componentes de aplicação 1250 podem compreender qualquer tipo e forma de elementos, hardware ou software, de um dispositivo de computação, tal como o dispositivo de computação 135 ilustrado e discutido em conjunto com as figuras 1C e 1D aqui. Em algumas modalidades, a aplicação 1250 pode compreender um servidor, um gateway, roteador, comutador, ponte ou outro tipo de dispositivo de computação ou de rede, e ter quaisquer elementos de hardware e/ou software associados a isso.

[00493] O sistema operacional de aplicação 1250 aloca, gerencia ou segrega de outra forma a memória de sistema disponível no espaço de kernel 3204 e no espaço de usuário 3202. Em uma arquitetura de software de exemplo 3200, o sistema operacional pode ser qualquer tipo e/ou forma de sistema operacional Unix. Como tal, a aplicação 1250 pode estar rodando qualquer sistema operacional, tais como qualquer uma das versões dos sistemas operacionais Microsoft Windows®, as diferentes versões dos sistemas operacionais Unix e Linux, qualquer versão de Mac OS® para computadores Macintosh, qualquer sistema operacional embutido, qualquer sistema operacional de rede, qualquer sistema operacional em tempo real, qualquer sistema operacional de fonte aberto, qualquer sistema operacional proprietário, quaisquer sistemas operacionais para dispositivos de computação móveis ou dispositivos de rede, ou qualquer outro sistema operacional capaz de rodar na aplicação 1250 e realizar as operações descritas aqui.

[00494] O espaço de kernel 3204 é reservado para rodar o kernel

3230, incluindo quaisquer drivers de dispositivo, extensões de kernel ou outro software relacionado a kernel. Conforme conhecido por aqueles versados na técnica, o kernel 3230 é o núcleo do sistema operacional, e provê acesso, controle e gerenciamento de recursos e elementos relacionados a hardware da aplicação 1250. De acordo com uma modalidade, o espaço de kernel 3204 também inclui vários serviços ou processos de rede funcionando em conjunto com um gerenciador de cache 3232, às vezes referido também como o cache integrado, cujos benefícios são descritos em maiores detalhes abaixo. Adicionalmente, a modalidade do kernel 3230 dependerá da modalidade do sistema operacional instalada, configurada ou usada de outra forma pelo dispositivo 1250.

[00495] Em uma modalidade, o dispositivo 1250 compreende uma pilha de rede 3267, tal como uma pilha baseada em TCP/IP, para comunicação com o cliente 10 e/ou o servidor 30. Em uma modalidade, a pilha de rede 3267 é usada para comunicação com uma primeira rede, tal como a rede 40, e uma segunda rede 40. Em algumas modalidades, o dispositivo 1250 termina uma primeira conexão de camada de transporte, tal como uma conexão de TCP de um cliente 10, e estabelece uma segunda conexão de camada de transporte com um servidor 30 para uso pelo cliente 10, por exemplo, a segunda conexão de camada de transporte é terminada na aplicação 1250 e no servidor 30. As primeira e segunda conexões de camada de transporte podem ser estabelecidas através de uma pilha de rede única 3267. Em outras modalidades, o dispositivo 1250 pode compreender múltiplas pilhas de rede, por exemplo, 3267 e 3267', e a primeira conexão de camada de transporte pode ser estabelecida ou terminada em uma pilha de rede 3267, e a segunda conexão de camada de transporte na segunda pilha de rede 3267'. Por exemplo, uma pilha de rede pode ser para o recebimento e a transmissão de um pacote de rede em uma primeira

rede, e uma outra pilha de rede para o recebimento e a transmissão de pacotes de rede em uma segunda rede. Em uma modalidade, a pilha de rede 3267 compreende um buffer 3243 para enfileiramento de um ou mais pacotes de rede para transmissão pela aplicação 1250.

[00496] Conforme mostrado na figura 27, o espaço de kernel 3204 inclui o gerenciador de cache 3232, um agente de pacote integrado de camada 2-7 de alta velocidade 3240, um agente de encriptação 3234, um agente de política 3236 e uma lógica de compressão de protocolo múltiplo 3238. Rodar estes componentes ou processos 3232, 3240, 3234, 3236 e 3238 no espaço de kernel 3204 ou no modo de kernel ao invés de no espaço de usuário 3202 melhora a performance de cada um destes componentes, sozinhos ou em combinação. Uma operação de kernel significa que estes componentes ou processos 3232, 3240, 3234, 3236 e 3238 rodam no espaço de endereço de núcleo do sistema operacional do dispositivo 1250. Por exemplo, rodar o agente de encriptação 3234 no modo de kernel melhora a performance de encriptação pelo movimento das operações de encriptação e desencriptação para o kernel, desse modo se reduzindo o número de transições entre o espaço de memória ou uma linha de kernel no modo de kernel e o espaço de memória ou uma linha no modo de usuário. Por exemplo, os dados obtidos no modo de kernel podem não precisar ser passados ou copiados para um processo ou uma linha rodando em modo de usuário, tal como a partir de uma estrutura de dados de nível de kernel para uma estrutura de dados de nível de usuário. Em um outro aspecto, o número de comutadores de contexto entre o modo de kernel e o modo de usuário também é reduzido. Adicionalmente, a sincronização de e as comunicações entre qualquer um dos componentes ou processos 3232, 3240, 3234, 3236 e 3238 podem ser realizadas mais eficientemente no espaço de kernel 3204.

[00497] Em algumas modalidades, qualquer porção dos

componentes 3232, 3240, 3234, 3236 e 3238 pode rodar ou operar no espaço de kernel 3204, enquanto outras porções destes componentes 3232, 3240, 3234, 3236 e 3238 podem rodar ou operar no espaço de usuário 3202. Em uma modalidade, uma estrutura de dados de nível de kernel é usada para a provisão de acesso a qualquer porção de um ou mais pacotes de rede, por exemplo, um pacote de rede compreendendo uma requisição de um cliente 10 ou uma resposta de um servidor 30. Em algumas modalidades, a estrutura de dados de nível de kernel pode ser obtida pelo agente de pacote 3240 através de uma interface ou filtro de driver de camada de transporte para a pilha de rede 3267. A estrutura de dados de nível de kernel pode compreender qualquer interface e/ou dados acessíveis através do espaço de kernel 3204 relacionados à pilha de rede 3267, tráfego de rede ou pacotes recebidos ou transmitidos pela pilha de rede 3267. Em outras modalidades, a estrutura de dados de nível de kernel pode ser usada por qualquer um dos componentes ou processos 3232, 3240, 3234, 3236 e 3238 para a realização da operação desejada do componente ou processo. Em uma modalidade, um componente 3232, 3240, 3234, 3236 e 3238 está rodando no espaço de kernel 3204 quando usa a estrutura de dados de nível de kernel, enquanto em uma outra modalidade o componente 3232, 3240, 3234, 3236 e 3238 está rodado em modo de usuário quando usando a estrutura de dados de nível de kernel. Em algumas modalidades, a estrutura de dados de nível de kernel pode ser copiada ou passada para uma segunda estrutura de dados de nível de kernel, ou qualquer estrutura de dados de nível de usuário desejada.

[00498] O gerenciador de cache 3232 pode compreender um software, um hardware ou qualquer combinação de software e hardware para a provisão de acesso, controle e gerenciamento de cache de qualquer tipo e forma de conteúdo, tais como objetos ou

objetos gerados dinamicamente apresentados pelos servidores de origem 30. Os dados, objetos ou conteúdo processados e armazenados pelo gerenciador de cache 3232 podem compreender dados em qualquer formato, tal como uma linguagem de marcação, ou comunicados através de qualquer protocolo. Em algumas modalidades, o gerenciador de cache 3232 duplica os dados originais armazenados em qualquer lugar ou dados previamente computados, gerados ou transmitidos, em que os dados originais podem requerer um tempo de acesso mais longo para buscar, computar ou obter de outra forma em relação à leitura de um elemento de memória cache. Uma vez que os dados sejam armazenados no elemento de memória cache, um uso futuro pode ser feito pelo acesso à cópia armazenada em cache, ao invés de se buscarem de novo e recomputarem os dados originais, desse modo se reduzindo o tempo de acesso. Em algumas modalidades, o NAT de elemento de memória cache compreende um objeto de dados na memória 3264 de dispositivo 1250. Em outras modalidades, o elemento de memória cache pode compreender uma memória tendo um tempo de acesso mais rápido do que a memória 3264. Em uma outra modalidade, o elemento de memória cache pode compreender qualquer tipo e forma de elemento de armazenamento do dispositivo 1250, tal como uma porção de um disco rígido. Em algumas modalidades, a unidade de processamento 3262 pode prover uma memória cache para uso pelo gerenciador de cache 3232. Ainda em outras modalidades, o gerenciador de cache 3232 pode usar qualquer porção ou combinação de memória, armazenamento ou unidade de processamento para armazenamento em cache de dados, objetos e outro conteúdo.

[00499] Mais ainda, o gerenciador de cache 3232 inclui quaisquer lógica, funções, regras ou operações para a realização de quaisquer modalidades das técnicas descritas aqui. Por exemplo, o gerenciador

de cache 3232 inclui uma lógica ou funcionalidade para invalidação de objetos com base na expiração de um período de tempo de invalidação ou mediante o recebimento de um comando de invalidação de um cliente 10 ou servidor 30. Em algumas modalidades, o gerenciador de cache 3232 pode operar como um programa, serviço, processo ou tarefa em execução no espaço de kernel 3204, e, em outras modalidades, no espaço de usuário 3202. Em uma modalidade, uma primeira porção do gerenciador de cache 3232 se executa no espaço de usuário 3202, enquanto uma segunda porção se executa no espaço de kernel 3204. Em algumas modalidades, o gerenciador de cache 3232 pode compreender qualquer tipo de processador de finalidade geral (GPP), ou qualquer outro tipo de circuito integrado, tal como um Arranjo de Porta Programável de Campo (FGPA), um Dispositivo Lógico Programável (PLD) ou um Circuito Integrado Específico de Aplicação (ASIC).

[00500] O agente de política 3236 pode incluir, por exemplo, um agente de estatística inteligente ou aplicativo(s) programável(is). Em uma modalidade, o agente de política 3236 provê um mecanismo de configuração para se permitir que um usuário identifique, especifique, defina ou configure uma política de armazenamento em cache. O agente de política 3236, em algumas modalidades, também tem acesso a uma em para suporte de estruturas de dados tais como tabelas de consulta ou tabelas de comprovação (hash) para se permitirem decisões de política de armazenamento em cache selecionadas por usuário. Em outras modalidades, o agente de política 3236 pode prover quaisquer lógica, regras, funções ou operações para a determinação e a provisão detecção cesso, controle e gerenciamento de objetos, dados ou conteúdo sendo armazenados em cache pela aplicação 1250, além de acesso, controle e gerenciamento de segurança, tráfego de rede, acesso de rede, compressão ou

qualquer outra função ou operação realizada pela aplicação 1250. Em algumas modalidades, o agente de política 3236 pode ser integrado com uma funcionalidade do agente de política 406. Em uma modalidade, o agente de política 3236 pode determinar decisões de política de armazenamento em cache com base em uma informação provida por um agente de coleta 404. Em algumas modalidades, o agente de política 3236 pode determinar decisões de política de armazenamento em cache com base em um tipo de execução de aplicativo. Em uma modalidade, o agente de política pode determinar decisões de política de armazenamento em cache com base em se um aplicativo está sendo transmitido para um cliente 10. Exemplos adicionais de políticas são adicionalmente descritos aqui.

[00501] O agente de encriptação 3234 compreende quaisquer lógica, regras comerciais, funções ou operações para manipulação do processamento de qualquer protocolo relacionado à segurança, tal como SSL ou TLS, ou qualquer função relacionada a isso. Por exemplo, o agente de encriptação 3234 encripta e desencripta pacotes de rede, ou qualquer porção dos mesmos, comunicados através da aplicação 1250. O agente de encriptação 3234 também pode configurar ou estabelecer conexões de SSL ou TLS em nome do cliente 10, do servidor 30 ou da aplicação 1250. Como tal, o agente de encriptação 3234 provê uma diminuição de carga e uma aceleração do processamento de SSL. Em uma modalidade, o agente de encriptação 3234 usa um protocolo de tunelamento para a provisão de uma rede privada virtual entre um cliente 10 e um servidor 30. Em algumas modalidades, o agente de encriptação 3234 está em comunicação com o processador de encriptação 3260. Em outras modalidades, o agente de encriptação 3234 compreende instruções executáveis rodando no processador de encriptação 3260.

[00502] O agente de compressão de protocolo múltiplo 3238

compreende quaisquer lógica, regras comerciais, função ou operações para compressão de um ou mais protocolos de um pacote de rede, tal como qualquer um dos protocolos usados pela pilha de rede 3267 do dispositivo 1250. Em uma modalidade, o agente de compressão de protocolo múltiplo 3238 comprime de forma bidirecional entre os clientes 10 e os servidores 30 qualquer protocolo baseado em TCP/IP, incluindo uma Interface de Programação de Aplicativo de Envio de Mensagem (MAPI) (e-mail), um Protocolo de Transferência de Arquivo (FTP), um Protocolo de Transferência de Hipertexto (HTTP), protocolo de Sistema de Arquivo de Internet Comum (CIFS) (transferência de arquivo), um protocolo de Arquitetura de Computação Independente (ICA), um Protocolo de Área de Trabalho Remota (RDP), um Protocolo de Aplicativo Sem fio (WAP), um protocolo de IP móvel, e um protocolo de Voz por IP (VoIP). Em outras modalidades, o agente de compressão de protocolo múltiplo 3238 provê uma compressão de protocolos baseados em Linguagem de Marcação de Hipertexto (HTML) e, em algumas modalidades, provê a compressão de quaisquer linguagens de marcação, tal como a Linguagem de Marcação Extensível (XML). Em uma modalidade, o agente de compressão de protocolo múltiplo 3238 provê a compressão de qualquer protocolo de alta performance, tal como qualquer protocolo projetado para comunicações de aplicação 1250 para aplicação 1250. Em uma outra modalidade, o agente de compressão de protocolo múltiplo 3238 comprime qualquer carga útil de qualquer comunicação usando um protocolo de controle de transporte modificado, tais como TCP de Transação (T/TCP), TCP com reconhecimentos de seleção (TCP-SACK), TCP com janelas grandes (TCP-LW), um protocolo de predição de congestionamento, tal como o protocolo TCP-Vegas e um protocolo enganador de TCP.

[00503] Como tal, o agente de compressão de protocolo múltiplo

3238 acelera a performance para usuários acessando aplicativos através de clientes de área de trabalho, por exemplo, os clientes de Microsoft Outlook e leves não da web, tal como qualquer cliente aberto por aplicativos de empresa populares, tais como Oracle, SAP e Siebel, e mesmo clientes móveis, tais como Pocket PC. Em algumas modalidades, o agente de compressão de protocolo múltiplo 3238 pela execução no espaço de kernel 3204 e integração com o agente de processamento de pacote 3240 acessando a pilha de rede 3267 é capaz de comprimir qualquer um dos protocolos portados pelo protocolo de TCP/IP, tal como qualquer protocolo de camada de aplicativo.

[00504] O agente de pacote integrado de camada 2-7 de alta velocidade 3240, também referido geralmente como um agente de processamento de pacote ou um agente de pacote, é responsável pelo gerenciamento do processamento de nível de kernel de pacotes recebidos e transmitidos pela aplicação 1250 através das portas de rede 3266. O agente de pacote integrado de camada 2-7 de alta velocidade 3240 pode compreender um buffer para enfileiramento de um ou mais pacotes de rede durante um processamento, tal como para recebimento de um pacote de rede ou transmissão de um pacote de rede. Adicionalmente, o agente de pacote integrado de camada 2-7 de alta velocidade 3240 está em comunicação com uma ou mais pilhas de rede 3267 para o envio e o recebimento de pacotes de rede através das portas de rede 3266. O agente de pacote integrado de camada 2-7 de alta velocidade 3240 trabalha em conjunto com o agente de encriptação 3234, o gerenciador de cache 3232, o agente de política 3236 e a lógica de compressão de protocolo múltiplo 3238. Em particular, o agente de encriptação 3234 é configurado para a realização de um processamento de SSL de pacotes, o agente de política 3236 é configurado para realizar funções relacionadas ao

gerenciamento de tráfego, tais como comutação de conteúdo de nível de requisição e redirecionamento de cache de nível de requisição, e lógica de compressão de protocolo múltiplo 3238 é configurada para realizar funções relacionadas à compressão e à descompressão de dados.

[00505] O agente de pacote integrado de camada 2-7 de alta velocidade 3240 inclui um temporizador de processamento de pacote 3242. Em uma modalidade, o temporizador de processamento de pacote 3242 provê um ou mais intervalos de tempo para disparo do processamento de pacotes de rede chegando, isto é, recebidos, ou saindo, isto é, transmitidos. Em algumas modalidades, o agente de pacote integrado de camada 2-7 de alta velocidade 3240 processa pacotes de rede em resposta ao temporizador 3242. O temporizador de processamento de pacote 3242 provê qualquer tipo e forma de sinal para o agente de pacote integrado de camada 2-7 de alta velocidade 3240 para notificar, disparar ou comunicar um evento relacionado ao tempo, intervalo ou ocorrência. Em muitas modalidades, o temporizador de processamento de pacote 3242 opera na ordem de milissegundos, tal como, por exemplo, 100 ms, 50 ms ou 25 ms. Por exemplo, em algumas modalidades, o temporizador de processamento de pacote 3242 provê intervalos de tempo ou de outra forma faz com que um pacote de rede seja processado pelo agente de pacote integrado de camada 2-7 de alta velocidade 3240 em um intervalo de tempo de 10 ms, enquanto em outras modalidades, em um intervalo de tempo de 5 ms, e ainda em modalidades adicionais, um intervalo de tempo tão curto quanto 3, 2 ou 1 ms. O agente de pacote integrado de camada 2-7 de alta velocidade 3240 pode ter uma interface, ser integrado ou estar em comunicação com o agente de encriptação 3234, o gerenciador de cache 3232, o agente de política 3236 e o agente de compressão de protocolo múltiplo 3238 durante

uma operação. Como tal, quaisquer dentre lógica, funções ou operações do agente de encriptação 3234, do gerenciador de cache 3232, do agente de política 3236 e da lógica de compressão de protocolo múltiplo 3238 podem ser realizadas em resposta ao temporizador de processamento de pacote 3242 e/ou ao agente de pacote 3240. Portanto, qualquer uma dentre lógica, funções ou operações do agente de encriptação 3234, do gerenciador de cache 3232, do agente de política 3236 e da lógica de compressão de protocolo múltiplo 3238 pode ser realizada na granularidade de intervalos de tempo providos através do temporizador de processamento de pacote 3242, por exemplo, em um intervalo de tempo menor do que ou igual a 10 ms. Por exemplo, em uma modalidade, o gerenciador de cache 3232 pode realizar uma invalidação de quaisquer objetos armazenados em cache em resposta ao agente de pacote integrado de camada 2-7 de alta velocidade 3240 e/ou ao temporizador de processamento de pacote 3242. Em uma outra modalidade, o tempo de expiração ou de invalidação de um objeto armazenado em cache pode ser regulado para ser da mesma ordem de granularidade que o intervalo de tempo do temporizador de processamento de pacote 3242, tal como a cada 10 ms.

[00506] Em contraste com o espaço de kernel 3204, o espaço de usuário 3202 é a área de memória ou porção do sistema operacional usada pelos aplicativos ou programas em modo de usuário de outra forma rodando no modo de usuário. Um aplicativo de modo de usuário não pode acessar o espaço de kernel 3204 diretamente, e usa chamadas de serviço de modo a acessar os serviços de kernel. Conforme mostrado na figura 27, o espaço de usuário 3202 de aplicação 1250 inclui uma interface gráfica de usuário (GUI) 3210, uma interface de linha de comando (CLI) 3212, serviços de shell 3214, programa de monitoração de saúde 3216, e serviços de daemon 3218.

A GUI 210 e a CLI 3212 provêm um meio pelo qual um administrador de sistema ou outro usuário pode interagir com e controlar a operação da aplicação 1250, tal como através do sistema operacional da aplicação 1250 e estar no espaço de usuário 3202 ou no espaço de kernel 3204. A GUI 3210 pode ser qualquer tipo e forma de interface gráfica de usuário, e pode ser apresentada através de texto, itens gráficos ou de outra forma, por qualquer tipo de programa ou aplicativo, tal como um navegador. A CLI 3212 pode ser qualquer tipo e forma de interface de linha de comando ou baseada em texto, tal como uma linha de comando provida pelo sistema operacional. Por exemplo, a CLI 3212 pode compreender um shell, o que é uma ferramenta para permitir que usuários interajam com o sistema operacional. Em algumas modalidades, a CLI 3212 pode ser provida através de um shell do tipo bash, csh, tcsh ou ksh. Os serviços de shell 3214 compreendem os programas, serviços, tarefas, processos ou instruções executáveis para suporte de uma interação com a aplicação 1250 ou o sistema operacional por um usuário através da GUI 3210 e/ou da CLI 3212.

[00507] O programa de monitoração de saúde 3216 é usado para a monitoração, a checagem, o relatório e a garantia que os sistemas de rede estejam funcionando apropriadamente e que os usuários estejam recebendo o conteúdo requisitado por uma rede. O programa de monitoração de saúde 3216 compreende um ou mais programas, serviços, tarefas, processos ou instruções executáveis para a provisão de lógica, regras, funções ou operações para a monitoração de qualquer atividade da aplicação 1250. Em algumas modalidades, o programa de monitoração de saúde 3216 intercepta e inspeciona qualquer tráfego de rede passado através da aplicação 1250. Em outras modalidades, o programa de monitoração de saúde 3216 tem uma interface por quaisquer meios e/ou mecanismos adequados com

um ou mais dos seguintes: o agente de encriptação 3234, o gerenciador de cache 3232, o agente de política 3236, a lógica de compressão de protocolo múltiplo 3238, o agente de pacote 3240, os serviços de daemon 3218 e os serviços de shell 3214. Como tal, o programa de monitoração de saúde 3216 pode chamar qualquer interface de programação de aplicativo (API) para a determinação de um estado, status ou saúde de qualquer porção da aplicação 1250. Por exemplo, o programa de monitoração de saúde 3216 pode dar um ping ou enviar uma inquirição de status em uma base periódica para checar se um programa, processo, serviço ou tarefa está ativa e rodando atualmente. Em um outro exemplo, o programa de monitoração de saúde 3216 pode checar quaisquer registros de status, erro ou histórico providos por qualquer programa, processo, serviço ou tarefa para determinar qualquer condição, status ou erro com qualquer porção da aplicação 1250.

[00508] Os serviços de daemon 3218 são programas que rodam ou no fundo e lidam com requisições de serviço periódicas recebidas pela aplicação 1250. Em algumas modalidades, um serviço de daemon pode encaminhar as requisições para outros programas ou processos, tal como um outro serviço de daemon 3218, conforme apropriado. Conforme é conhecido por aqueles versados na técnica, um serviço de daemon 3218 pode rodar sem assistência para a realização de funções amplas de sistema contínuas ou periódicas, tal como controle de rede, ou para a realização de qualquer tarefa desejada. Em algumas modalidades, um ou mais serviços de daemon 3218 rodam no espaço de usuário 3202, enquanto em outras modalidades um ou mais serviços de daemon 3218 rodam no espaço de kernel.

[00509] Um conteúdo dinâmico, tais como um ou mais objetos gerados dinamicamente, pode ser gerado por servidores, referidos como servidores de aplicativo ou de origem 30 e/ou bancos de dados

de back-end que processam requisições de objeto a partir de um ou mais clientes 10, locais ou remotos, conforme descrito na figura 1A. Conforme aqueles aplicativos ou bancos de dados processam dados, incluindo dados relacionados a entradas recebidas a partir de clientes, os objetos de resposta apresentados por estes bancos de dados e aplicativos podem mudar. Os objetos anteriores gerados por aqueles aplicativos ou bancos de dados em um servidor de origem não mais serão novos e, portanto, não devem mais ser armazenados por um cache. Por exemplo, dado o mesmo conjunto de entradas, um objeto gerado dinamicamente de uma primeira instância pode ser diferente de um objeto gerado dinamicamente de uma segunda instância. Em um outro exemplo, o mesmo objeto pode ser gerado dinamicamente com um conjunto diferente de entradas, de modo que uma primeira instância do objeto seja gerada diferentemente de uma segunda instância do objeto.

[00510] De modo a se obter uma performance de rede melhorada, a aplicação 1250 é projetada e configurada para se dirigir aos problemas que surgem no armazenamento em cache de conteúdo gerado dinamicamente através de uma variedade de métodos, conforme descrito em detalhes abaixo. Em algumas modalidades descritas aqui, a aplicação 1250 incorpora um conjunto de uma ou mais técnicas para se tornar mais eficiente e efetiva a invalidação de conteúdo gerado dinamicamente armazenado no cache. Mais ainda, a aplicação pode incorporar técnicas para a realização de controle e armazenamento em cache para preenchimentos rápidos. As memórias cache tipicamente armazenam toda resposta a uma requisição para um objeto, desde que essa resposta não seja marcada como não armazenável em cache. Conforme descrito aqui um armazenamento em cache eficiente de conteúdos gerados dinamicamente requer técnicas que permitam a invalidação tempestiva de objetos na

memória cache que tenham sofrido uma mudança no servidor de origem. Uma invalidação tempestiva permite que o cache evite apresentar um conteúdo deteriorado – uma tarefa de preocupação em particular com conteúdo gerado dinamicamente, especialmente quando mudanças no conteúdo correm de forma irregular. São estabelecidas abaixo várias técnicas para se garantir uma invalidação tempestiva de conteúdo gerado dinamicamente.

a. Funcionalidade Integrada

[00511] Em um aspecto, o armazenamento em cache de objetos gerados dinamicamente está relacionado a técnicas de integração de funções, lógica ou operações do gerenciador de cache 3232, do agente de política 3236, do agente de encriptação 3234 e/ou do agente de compressão de protocolo múltiplo 3238 com as operações de processamento de pacote do agente de pacote integrado de camada 2-7 de alta velocidade 3240, em resposta ao temporizador de processamento de pacote 3242. Por exemplo, as operações do gerenciador de cache 3232 podem ser realizadas nos intervalos de tempo do temporizador de processamento de pacote 3242 usados para operações de processamento de pacote, tal como em uma recepção ou uma transmissão de um pacote de rede. Em uma modalidade, pela integração com as operações de processamento de pacote e/ou pelo uso do temporizador de processamento de pacote, o gerenciador de cache 3232 pode armazenar em cache objetos com tempos de expiração baixos até intervalos de tempo muito pequenos, conforme será descrito em maiores detalhes abaixo. Em outras modalidades, o gerenciador de cache 3232 em resposta ao temporizador de processamento de pacote 3242 também pode receber um comando de invalidação para invalidar um objeto em um período de tempo muito curto de armazenamento em cache do objeto.

[00512] O método 3300 descrito na figura 28A ilustra uma

modalidade de uma técnica para se requisitar que o gerenciador de cache 3232, o agente de política 3236, o agente de encriptação 3234 e/ou o agente de compressão de protocolo múltiplo 3238 realizem uma operação durante um processamento ou em associação com os intervalos de tempo para processamento de um pacote de rede pelo agente de pacote integrado de camada 2-7 de alta velocidade ou agente de processamento de pacote 3240. Em uma visão geral breve, na etapa 3310 do método 3300, o dispositivo 1250 recebe um pacote de rede ou é requisitado para transmitir um pacote de rede. Na etapa 3315, o dispositivo 31250 requisita que o agente de processamento de pacote 3240 processe o pacote de rede, em resposta ao temporizador de processamento de pacote 3242. Como parte de ou associado às operações de processamento de pacote, na etapa 3320, o agente de processamento de pacote 240 requisita que o gerenciador de cache 3232, o agente de política 3236, o agente de encriptação 234 e/ou o agente de compressão de protocolo múltiplo 3238 realize uma operação em um objeto armazenado em cache. Na etapa 3325, o gerenciador de cache 3232, o agente de política 3236, o agente de encriptação 234 e/ou o agente de compressão de protocolo múltiplo 3238 executa a operação requisitada, o que pode incluir qualquer uma ou uma combinação das técnicas descritas aqui. Em uma modalidade, o gerenciador de cache 3232 determina uma invalidação de um objeto armazenado em cache, e marca o objeto armazenado em cache como inválido. Em algumas modalidades, o gerenciador de cache 3232 descarta o objeto inválido em resposta a uma requisição pelo agente de processamento de pacote 3240. Conforme o gerenciador de cache 3232 está realizando estas operações em resposta ao temporizador de processamento de pacote 3242, uma invalidação de objetos pode ocorrer em períodos de tempo da ordem de milissegundos e com objetos tendo uma expiração da ordem dos intervalos de tempo

providos pelo temporizador de processamento de pacote 3242, tal como 10 ms.

[00513] Em maiores detalhes do método 3300, na etapa 3310, a aplicação 1250 recebe um ou mais pacotes de rede, e/ou transmite um ou mais pacotes de rede. Em algumas modalidades, a aplicação 1250 requisita para transmitir um ou mais pacotes de rede pela rede 40 ou pela rede 40'. Em uma outra modalidade, a aplicação 1250 recebe um pacote de rede em uma porta 3266 e transmite um pacote de rede na mesma porta 3266 ou em uma porta diferente 3266'. Em algumas modalidades, o agente de pacote 3240 da aplicação 1250 transmite ou requisita para transmitir um ou mais pacotes de rede. Em uma modalidade, a aplicação 1250 recebe ou transmite um pacote em uma primeira rede 40, enquanto em uma outra modalidade a aplicação 1250 recebe ou transmite um pacote em uma segunda rede 40'. Em outras modalidades, a aplicação 1250 recebe e transmite pacotes na mesma rede 40. Em algumas modalidades, a aplicação 1250 recebe e/ou transmite pacotes de rede para um ou mais clientes 10. Em outras modalidades, a aplicação 1250 recebe e/ou transmite pacotes de rede para um ou mais servidores 30.

[00514] Na etapa 3315, o dispositivo 1250 pode requisitar ou disparar operações de processamento de pacote do agente de processamento de pacote 3240 mediante o recebimento de um pacote de rede na porta de rede porta de rede 3266 do dispositivo 1250 ou mediante uma requisição para transmissão de um pacote de rede a partir do dispositivo 1250, ou mediante qualquer combinação de recepção e/ou transmissão de um ou mais pacotes de rede. Em algumas modalidades, as operações de processamento de pacote do agente de processamento de pacote 3240 são disparadas através de um sinal provido por um temporizador de processamento de pacote 3242. Em uma modalidade, o temporizador de processamento de

pacote 3242 pode prover uma funcionalidade de temporizador acionada por interrupção ou acionada por evento relacionada à recepção e/ou transmissão de um ou mais pacotes de rede. Em algumas modalidades, o temporizador de processamento de pacote 3242 é acionado por uma taxa de recepção e/ou transmissão de pacotes de rede através do dispositivo 1250 ou pela taxa pela qual cada pacote ou lote de pacotes é processado. Como tal, o temporizador de processamento de pacote 3242 pode ser disparado e reinicializado após cada conjunto de uma ou mais operações de processamento de pacote. Em uma outra modalidade, o temporizador de processamento de pacote 3242 provê intervalos de tempo, intervalos de tempo iguais ou variáveis, para disparo, ativação ou sinalização para o agente de processamento de pacote 3240 realizar uma função ou operação, tal como a manipulação de um pacote recebido ou a transmissão de um pacote submetido. Conforme discutido acima com respeito ao dispositivo 1250 da figura 27, o temporizador de processamento de pacote 3242 pode operar em da ordem de milissegundos, tal como causando intervalos de tempo ou disparo de operações de processamento de pacote em intervalos de 10 ms ou menos. A funcionalidade de temporizador granular do temporizador de processamento de pacote pode ser provida de várias formas e usada em operações das operações de processamento de pacote do agente de processamento de pacote 3240.

[00515] Na etapa 3320 do método 3300, o agente de processamento de pacote 3240 requisita que um ou mais dentre o gerenciador de cache 3232, o agente de política 3236, o agente de encriptação 3234 e/ou o agente de compressão de protocolo múltiplo 3238 executem uma operação. Em uma modalidade, o agente de processamento de pacote 3240 ou o temporizador de processamento de pacote 3242 gera um sinal ou sinais para um ou mais dentre o

gerenciador de cache 3232, o agente de política 3236, o agente de encriptação 3234 e/ou o agente de compressão de protocolo múltiplo 3238. O agente de processamento de pacote 3240 pode requisitar ou sinalizar a operação em qualquer ponto antes, durante ou depois de uma operação de processamento de pacote de um pacote de rede, ou um ou mais pacotes. Em uma modalidade, o agente de processamento de pacote 240 faz a requisição mediante um disparo do temporizador de processamento de pacote 3242 ou uma expiração de um intervalo de tempo provido pelo temporizador de processamento de pacote 3242, e antes da realização de uma operação de processamento de pacote em um pacote de rede. Em uma outra modalidade, no decorrer da realização de uma ou mais operações de processamento de pacote, o agente de processamento de pacote 3240 faz a requisição. Por exemplo, durante uma execução de uma operação, tal como em uma chamada de função, o agente de processamento de pacote 240 pode fazer uma chamada de interface de programação de aplicativo (API) para um dentre o gerenciador de cache 3232, o agente de política 3236, o agente de encriptação 3234 e/ou o agente de compressão de protocolo múltiplo 238. Em outras modalidades, o agente de processamento de pacote 3240 faz a requisição mediante a conclusão de uma operação de processamento de pacote de rede.

[00516] Na etapa 3325, a operação requisitada é realizada por um ou mais dentre o gerenciador de cache 3232, o agente de política 3236, o agente de encriptação 3234 e/ou o agente de compressão de protocolo múltiplo 3238. Em algumas modalidades, qualquer funcionalidade ou operação provida através do kernel 3204 pode ser requisitada para ser executada, tal como através de uma interface de programação de aplicativo de kernel (API). Como tal, qualquer uma das funções do dispositivo 1250 pode ser realizada em conjunto com o

sincronismo ou os intervalos de sincronismo de processamento de pacote através do temporizador de processamento de pacote 3242. Em algumas modalidades, a operação requisitada é realizada de forma síncrona e em conjunto com as operações de processamento de pacote do agente de processamento de pacote 3240. Por exemplo, as operações de processamento de pacote esperam e continuam mediante a conclusão de ou em resposta à operação requisitada. Em outras modalidades, a operação requisitada é realizada de forma assíncrona com as operações de processamento de pacote. Por exemplo, o agente de processamento de pacote 3240 envia uma requisição para a realização da operação, mas não bloqueia ou espera para receber uma resposta da operação. Conforme será discutido em maiores detalhes em conjunto com o método 3350 descrito na figura 28B, o agente de processamento de pacote 3240 pode requisitar que o gerenciador de cache 3232 realize qualquer função de gerenciamento de cache, tais como a checagem quanto à expiração ou invalidação de observação, marcação de objetos como inválidos, ou descarte de objetos inválidos ou expirados.

[00517] Em algumas modalidades, o agente de processamento de pacote 3240 na etapa 3320 envia múltiplas requisições, tal como uma primeira requisição para o gerenciador de cache 232 e uma segunda requisição para o agente de encriptação 3234. Em outras modalidades, o agente de processamento de pacote 3240, na etapa 3320, envia uma requisição única compreendendo múltiplas requisições a serem distribuídas pelo dispositivo 1250, tal como através do kernel 3230 para o componente pretendido do dispositivo 1250. Em uma modalidade, as requisições são comunicadas subsequente a cada outra. Em uma outra modalidade, as requisições podem ser dependentes do status, resultado, sucesso ou conclusão de uma requisição prévia. Por exemplo, uma primeira

requisição para o agente de política 3236 pode ser usada para se determinar uma política para processamento de um pacote de rede a partir de um outro dispositivo ou um usuário associado ao pacote de rede. Com base em uma política do agente de política 3236, uma segunda requisição para o cache pode ser feita ou não ser feita, dependendo de um resultado da primeira requisição. Com o gerenciador de cache 3232, o agente de política 3236, o agente de encriptação 3234 e/ou o agente de compressão de protocolo múltiplo 3238 integrados no espaço de kernel 3204 do dispositivo 1250 com o agente de processamento de pacote 3240, há várias operações do dispositivo 1250, conforme descrito aqui, que podem ser disparadas por e integradas com as operações de processamento de pacote.

b. Granularidade de Invalidação

[00518] Em um outro aspecto, um armazenamento em cache de objetos gerados dinamicamente está relacionado a e incorpora a capacidade de configuração do tempo de expiração de objetos armazenados pelo cache para intervalos de tempo granulares finos, tal como a granularidade de intervalos de tempo providos pelo temporizador de processamento de pacote. Esta característica é referida como "granularidade de invalidação". Como tal, em uma modalidade, os objetos com tempos de expiração baixos até intervalos de tempo muito pequenos podem ser armazenados em cache. Em uma modalidade, o gerenciador de cache em resposta a um temporizador de processamento de pacote também pode receber um comando de invalidação para invalidação de um objeto em um período de tempo muito curto de armazenamento em cache do objeto. Pela provisão desta granularidade fina no tempo de expiração, o cache pode armazenar em cache e apresentar objetos que mudam freqüentemente, às vezes mesmo muitas vezes em um segundo. Uma técnica é alavancar o temporizador de processamento de pacote

usado pelo dispositivo em uma modalidade, que é capaz de operar em incrementos de tempo da ordem de milissegundos, para se permitir uma granularidade de invalidação ou expiração baixa de até 10 ms ou menos. Caches tradicionais, em contraste, tipicamente, não regulam a granularidade de expiração ou de invalidação em menos de um segundo.

[00519] Com referência, agora, à figura 28B, uma modalidade de um método 3350 é descrita para invalidação ou expiração de um objeto armazenado em cache em resposta ao temporizador de processamento de pacote 3242 e/ou ao agente de processamento de pacote 3240. Como tal, em algumas modalidades, os objetos armazenados em cache podem ser invalidados ou expirados em da ordem de milissegundos, tal como em 10 ms ou menos. Em uma visão geral, na etapa 3355 de método 3350, o gerenciador de cache 3232 recebe um sinal ou requisição para a realização de uma operação através do agente de processamento de pacote 3240, em resposta ao temporizador de processamento de pacote 3242. Na etapa 3360, o gerenciador de cache 3232 determina se um objeto armazenado em cache, tal como um objeto gerado dinamicamente, é inválido ou expirado. Na etapa 3365, se o objeto for inválido, o gerenciador de cache 3232 marcará o objeto como inválido, e na etapa 3370 descartará o objeto inválido do gerenciador de cache 3232.

[00520] Em maiores detalhes da etapa 3355, em algumas modalidades, o gerenciador de cache 3232 pode ser sinalizado ou requisitado para a realização de uma operação relacionada a cache em qualquer ponto de tempo durante o processamento de pacote de rede. Em uma modalidade, na etapa 3355, o gerenciador de cache 3232 recebe uma requisição de operação antes do processamento de um pacote de rede recebido ou a ser transmitido pelo dispositivo 1250. Em uma modalidade, o gerenciador de cache 3232 recebe uma

requisição de operação mediante a conclusão de processamento de um pacote de rede. Por exemplo, o agente de processamento de pacote 3240 completa o processamento de um pacote de rede, e antes de esperar pelo próximo intervalo de tempo do temporizador 3242 ou antes do processamento do próximo pacote, requisita que o cache realize uma operação. Em outras modalidades, durante uma operação de processamento de pacote, o agente de processamento de pacote 3240 comunica uma requisição de operação para o gerenciador de cache 3232. Em uma outra modalidade, o gerenciador de cache 3232 recebe um sinal, tal como a partir do agente de processamento de pacote 3240 ou do temporizador de processamento de pacote 3242 para disparar o gerenciador de cache 3232 para a realização de uma operação. Em algumas modalidades, o sinal indica para invalidar um objeto armazenado em cache ou para expirar uma expiração de um objeto armazenado em cache.

[00521] Em algumas modalidades, o gerenciador de cache 3232 pode receber uma requisição para realizar uma operação de cache a partir de uma entidade externa ao gerenciador de cache 3232, tal como uma requisição para invalidação de um objeto comunicado por um servidor 30, e processado pelo agente de processamento de pacote 3240. Em uma modalidade, o gerenciador de cache 3232 pode receber uma requisição de invalidação em 10 ms ou menos de armazenamento em cache do objeto, enquanto em uma outra modalidade, tão curto quanto 5 ms, 2 ms ou 1 ms. Em outras modalidades, o gerenciador de cache 3232 pode realizar uma operação de cache em resposta às operações ou à funcionalidade do gerenciador de cache 3232, tal como a expiração de um temporizador, para fazer com que um objeto seja invalidado ou durante o processamento de qualquer comando de cache. Em outras modalidades, o gerenciador de cache 3232 usa o temporizador de

processamento de pacote 3242 do dispositivo 1250 para disparar as operações de cache. Por exemplo, o temporizador 3242 pode disparar ou sinalizar para o cache checar quanto a uma invalidação ou expiração de um objeto armazenado em cache em qualquer intervalo de tempo capaz de ser regulado pelo temporizador 3242. Em uma modalidade, o temporizador 3242 pode ser regulado para disparar ou sinalizar o cache em 10 ms ou menos de ser regulado, ou em uma outra modalidade, tão curto quanto 5 ms, 2 ms ou 1 ms de ser regulado. Em algumas modalidades, o servidor de origem 30 pode regular o tempo de expiração do objeto. Em outras modalidades, o dispositivo 1250 ou o cliente 10 pode regular o tempo de expiração do objeto.

[00522] Na etapa 3360, o gerenciador de cache 3232 determina a invalidação ou expiração de um objeto armazenado no cache. Em algumas modalidades, um objeto em cache é invalidado com base na expiração de um temporizador. Em uma modalidade, o gerenciador de cache 3232 pode emitir um comando de invalidação sobre um objeto com base na expiração de um temporizador. Em uma outra modalidade, o objeto armazenado em cache é automaticamente invalidado pelo gerenciador de cache 3232, em resposta à expiração de um temporizador, tal como um temporizador regulado com o temporizador de processamento de pacote 3242. Em algumas modalidades, em resposta ao temporizador de processamento de pacote 3242, o gerenciador de cache 3232 checa quanto à expiração de quaisquer temporizadores quanto a objetos armazenados em cache. Em uma modalidade, o gerenciador de cache 3232 determina que um temporizador de objeto expirou, enquanto em uma outra modalidade, o gerenciador de cache 3232 determina que o temporizador de objeto não expirou. Em uma modalidade adicional, o gerenciador de cache 3232 em resposta a um segundo gatilho ou

segundo intervalo de tempo do temporizador de processamento de pacote 3242 checará uma segunda vez se um temporizador de objeto previamente armazenado em cache expirou.

[00523] Em algumas modalidades, o gerenciador de cache 3232 analisa gramaticalmente, interpreta, acessa, lê ou processa de outra forma um comando de invalidação ou uma requisição para a identificação do objeto para invalidação no cache. Em uma modalidade, uma entidade externa ao gerenciador de cache 3232 emite um comando de invalidação para o gerenciador de cache 3232, para invalidação do objeto. Em uma outra modalidade, a entidade externa pode emitir o comando de invalidação em resposta a um temporizador de processamento de pacote 3242. Se o objeto for válido e/ou não tiver sido invalidado, o gerenciador de cache 3232 invalidará o objeto em resposta à requisição. Em algumas modalidades, a requisição de invalidação processada pelo gerenciador de cache 3232 é em resposta às operações de processamento de pacote do agente de processamento de pacote 3240 processando a requisição, o que, por sua vez, pode ser em resposta ao temporizador de processamento de pacote 3242.

[00524] Na etapa 3365, o gerenciador de cache 3232 marca o objeto como inválido. O gerenciador de cache 3232 pode marcar cada objeto como inválido de qualquer maneira adequada ou desejada. Em uma modalidade, um objeto é marcado como inválido pela regulação de um indicador (flag), atributo ou propriedade do objeto armazenado. Por exemplo, um indicador (flag) pode ser regulado para qualquer valor identificando para o gerenciador de cache 3232 que o objeto é inválido. Em uma outra modalidade, um objeto pode ser marcado como inválido pelo movimento do objeto para uma área ou porção do cache para armazenamento de objetos inválidos. Em outras modalidades, o gerenciador de cache 3232 pode identificar ou

acompanhar o estado inválido e/ou válido de um objeto armazenado por um banco de dados ou uma lista com link ou qualquer tipo e forma de estrutura de dados. Em algumas modalidades, o gerenciador de cache 3232 usa um ou mais objetos para a identificação ou o acompanhamento da validade ou invalidade de um ou mais objetos armazenados no cache. Em uma outra modalidade, o objeto é marcado como inválido pela mudança, modificação ou alteração do objeto armazenado, por exemplo, apagando-se ou removendo-se uma porção do objeto, de modo que ele não possa ser usado, ou pela mudança ou mutilação do nome do objeto.

[00525] Na etapa 3370, o gerenciador de cache 3232, em algumas modalidades, descarta a partir do cache aqueles objetos marcados como inválidos. Em uma outra modalidade, o gerenciador de cache 3232 descarta o objeto inválido do cache mediante uma requisição do objeto, tal como por um cliente 10. Em algumas modalidades, o gerenciador de cache 3232 grava em cima do objeto inválido uma cópia ou versão atualizada do objeto recebido após uma invalidação ou expiração do objeto. Em uma outra modalidade, o gerenciador de cache 3232 reusa a memória cache ocupada pelo objeto inválido pelo armazenamento de uma outra ou da mesma porção de memória cache. Ainda em uma outra modalidade, o gerenciador de cache 3232 não descarta o objeto marcado como inválido, e sim mantém o objeto armazenado em memória ou no armazenamento do cache.

[00526] Embora o método 3350 descreva uma invalidação e um descarte de objetos armazenados em cache em resposta a um temporizador de processamento de pacote e/ou em conjunto com operações de processamento de pacote para a provisão de granularidade de invalidação, qualquer operação do cache e quaisquer técnicas de armazenamento em cache, bem como qualquer outra operação do dispositivo 1250 descrito aqui podem ser executadas em

intervalos de tempo granulares finos providos pelo temporizador de processamento de pacote. Em algumas modalidades, a invalidação ou a expiração de objetos armazenados em cache pode ocorrer em um intervalo de tempo tão curto quanto 100 ms, enquanto em uma outra modalidade um intervalo de tempo tão curto quanto 50 ms. Em algumas modalidades, a invalidação ou expiração de objetos armazenados em cache pode ocorrer em um intervalo de tempo tão curto quanto 25 ms e, em outras modalidades, em um intervalo de tempo tão curto quanto 10 ms. Enquanto isso, em outras modalidades, a invalidação ou expiração de objetos armazenados em cache pode ocorrer em um intervalo de tempo tão curto quanto 5 ms, e ainda em modalidades adicionais, em um intervalo de tempo tão curto quanto de 3, 2 ou 1 ms.

[00527] Pela incorporação da capacidade de invalidar objetos após o decorrer de incrementos de tempo muito pequenos, conforme descrito nos métodos 3300 e 3350, em conjunto com as figuras 28A e 28B acima, um armazenamento em cache melhorado de objetos gerados dinamicamente é habilitado. Algum conteúdo dinâmico de fato é propício a ser armazenado e apresentado a partir de um cache para períodos de tempo muito curtos. Para se armazenar em cache de forma bem-sucedida esse conteúdo, contudo, uma abordagem de acordo com uma modalidade provê o armazenamento em cache de objetos por períodos de tempo muito curtos, antes de o objeto ser invalidado e descartado da memória cache. Por exemplo, certos objetos gerados dinamicamente podem ser armazenados em cache por tanto quanto 1 segundo, mas qualquer coisa mais longa freqüentemente é inaceitável para um conteúdo que esteja mudando constantemente. Em uma modalidade, a abordagem incluía a invalidação ou expiração de um conteúdo armazenado em cache após frações pequenas de um segundo. Como um exemplo, se um

aplicativo levar 100 milissegundos para gerar uma resposta dinâmica, então, o cache poderá armazenar e apresentar aquela resposta por uma duração menor do que ou igual ao período de 100 milissegundos, sem comprometimento do frescor dos dados. Não haverá um novo objeto gerado durante aquele período de 100 milissegundos, porque ele é mais curto do que o tempo que se leva para gerar um novo objeto. A aplicação 1250 assim pode ser configurada para apresentar o objeto prévio por aquela duração. A capacidade da aplicação 1250 de invalidar incrementos de tempo realmente muito pequenos freqüentemente é útil para ambientes de aplicativo em que o nível de isolamento de transação de banco de dados é regulado para permitir Leituras Repetíveis ou Leituras Serializadas.

c. Comandos de Invalidação

[00528] A tecnologia de armazenamento em cache tradicional invalida um conteúdo armazenado com base em um tempo de expiração predefinido para o conteúdo, o qual tipicamente é configurado pelo administrador ou é recebido a partir do servidor que apresentou o objeto. É descrita abaixo uma outra técnica para invalidação de conteúdo, de modo a se armazenar em cache de forma mais eficiente um conteúdo gerado dinamicamente. Uma técnica inclui a capacidade de receber na aplicação 1250 um comando de invalidação que identifica um ou mais dos objetos armazenados previamente no cache como inválidos em tempo real. Por exemplo, o comando de invalidação pode ser comunicado através de um pacote de rede transmitido para o cliente ou uma chamada de interface de programação de aplicativo (API) feita por um servidor para a aplicação. Isto difere da abordagem tradicional pela qual o servidor simplesmente regula um tempo de expiração de cache que ele inclui no cabeçalho de objeto no momento em que o objeto é apresentado.

[00529] Uma técnica é ilustrada mais especificamente nas figuras 29A e 29B. A figura 29A é um fluxograma que ilustra um método para manutenção de um cache, tal como um cache de memória de computador. Em uma visão geral breve e de acordo com a etapa 3410, os objetos gerados dinamicamente previamente apresentados a partir de um servidor de origem 30 são armazenados no cache. Por exemplo, o objeto gerado dinamicamente pode não ser identificado como armazenável em cache ou incluir de outra forma qualquer cache ou informação de controle de cache. Na etapa 3420, um comando de invalidação é recebido no cache ou gerenciador de cache 3232. O comando de invalidação identifica um ou mais objetos apresentados previamente como inválidos. Na etapa 3430, em resposta ao comando de invalidação, o cache ou gerenciador de cache 3232 marca o objeto identificado como inválido.

[00530] Em maiores detalhes, na etapa 3410, o gerenciador de cache 3232 armazena em um elemento de memória cache um objeto gerado dinamicamente recebido, obtido ou comunicado a partir de qualquer fonte. Em algumas modalidades, o objeto gerado dinamicamente pode ser gerado e apresentado a partir de um servidor 30. Em outras modalidades, o objeto gerado dinamicamente pode ser gerado e comunicado por um cliente 10. Em algumas modalidades, uma outra porção, um componente ou processo da aplicação 1250 gera o objeto e armazena o objeto no cache. Em modalidades adicionais, o objeto gerado dinamicamente pode ser gerado por uma outra aplicação 1250 ou um outro dispositivo de computação na rede e transmitido ou comunicado para a aplicação 1250. Em algumas modalidades, o objeto gerado dinamicamente não é identificado como armazenável em cache ou identificado como não armazenável em cache. Em outras modalidades, o objeto gerado dinamicamente é identificado como armazenável em cache ou está sob um controle de

cache.

[00531] Na etapa 3420, o gerenciador de cache 3232 recebe um comando de invalidação identificando um objeto como inválido, tal como um objeto gerado dinamicamente armazenado no cache. Em uma modalidade, o comando de invalidação pode compreender qualquer tipo de diretiva ou instrução indicando para o cache que um objeto é inválido ou de outra forma pode estar deteriorado. Em algumas modalidades, o comando de invalidação identifica o objeto e também pode identificar o tempo em que o objeto está inválido, bem como que porções do objeto podem estar inválidas. Em uma modalidade, o gerenciador de cache 3232 provê uma interface de programação de aplicativo (API) que pode ser chamada remotamente por um servidor de origem 30. Em algumas modalidades, o gerenciador de cache 3232 pode prover qualquer tipo e forma de protocolo para o recebimento de comandos e a resposta a comandos através de um ou mais pacotes de rede. Em uma modalidade, o gerenciador de cache 3232 ou dispositivo 1250 provê uma interface API de Linguagem de Marcação Extensível (XML) para o recebimento e o processamento de comandos de invalidação. Por exemplo, o gerenciador de cache 3232 pode prover uma interface de serviço da web. Em algumas modalidades, o gerenciador de cache 3232 responde ao comando de invalidação pelo envio de um reconhecimento, status ou outra resposta ao servidor de origem 30. Em outras modalidades, o gerenciador de cache 3232 não responde ao comando de invalidação. Em uma modalidade, um objeto é marcado como inválido se um aplicativo rodando em um servidor de origem 30 realizou uma ação que tornou o objeto armazenado deteriorado, tal como por uma versão nova ou atualizada gerada do objeto. Isto poderia ocorrer, por exemplo, quando novos editores fizessem mudanças em uma reportagem de desenvolvimento rápido e,

portanto, quisessem estar certos que a versão mais recente da história está sendo apresentada a clientes.

[00532] Os comandos de invalidação podem ser emitidos a partir de um servidor de origem pelo aplicativo que gerou o objeto, por um outro servidor 30 ou por uma outra aplicação 1250. Em uma modalidade, o servidor de origem 30 emite ou comunica um comando de invalidação para o cache 3232 automaticamente, em resposta a uma mudança no objeto gerado dinamicamente no servidor de origem 30. O comando de invalidação também pode ser gerado por um controle administrativo exterior ou externo ao servidor 30 e à aplicação 1250. Por exemplo, o controle administrativo pode ser qualquer tipo e forma de programa ou aplicativo rodando na rede e em comunicação com a aplicação 1250, tal como um console de administrador. Mais ainda, um cliente 10 poderia emitir ou comunicar um comando de invalidação para a aplicação 1250 ou o gerenciador de cache 3232. Por exemplo, se o cliente precisasse executar uma ação que o cliente 10 reconhecesse que causaria uma mudança nos objetos requisitados no servidor de origem, o cliente poderia comunicar o comando de invalidação. Qualquer objeto armazenado no cache pode ser invalidado pela transmissão para o cache de um comando de usuário executado localmente no cache ou invocado remotamente usando-se a infraestrutura de API de XML.

[00533] De acordo com a etapa 3430, um objeto armazenado em cache, por exemplo, um objeto gerado dinamicamente previamente apresentado, que tenha sido identificado como inválido é marcado como tal em resposta ao comando de invalidação. Um objeto inválido não será provido para um cliente requisitando a partir do cache, mas, ao invés disso, seria apresentado diretamente a partir do servidor de origem. O gerenciador de cache 3232 pode marcar qualquer objeto como inválido de qualquer maneira adequada ou desejada. Em uma

modalidade, um objeto é marcado como inválido pela regulação de um indicador (flag), atributo ou propriedade do objeto armazenado. Por exemplo, um indicador (flag) pode ser regulado para qualquer valor identificando para o gerenciador de cache 3232 que o objeto é inválido. Em uma outra modalidade, um objeto pode ser marcado como inválido pelo movimento do objeto para uma área ou porção do cache para armazenamento de objetos inválidos. Em outras modalidades, o gerenciador de cache 3232 pode identificar ou acompanhar o estado inválido e/ou válido de um objeto armazenado por um banco de dados ou lista com link ou qualquer tipo e forma de estrutura de dados. Em algumas modalidades, o gerenciador de cache 3232 usa um ou mais objetos para a identificação ou o acompanhamento da validade ou invalidade de um ou mais objetos armazenados no cache. Em uma outra modalidade, o objeto é marcado como inválido pela mudança, modificação ou alteração do objeto armazenado, por exemplo, apagando-se ou removendo-se uma porção do objeto, de modo que ele não possa ser usado, ou pela mudança ou mutilação do nome do objeto.

[00534] Em algumas modalidades, a aplicação 1250 subsequente descarta do cache os objetos marcados como inválido. Em uma outra modalidade, a aplicação 1250 descarta o objeto inválido do cache mediante uma requisição do objeto, tal como por um cliente 10. Em algumas modalidades, a aplicação 1250 grava em cima do objeto inválido uma cópia ou versão atualizada do objeto. Em uma outra modalidade, a aplicação 1250 reusa a memória cache ocupada pelo objeto inválido pelo armazenamento de um outro objeto gerado dinamicamente para a mesma porção da memória cache.

[00535] Com a API de invalidação de comando do gerenciador de cache 3232, qualquer dispositivo de computação ou usuário em comunicação com a aplicação 1250 pode requisitar para invalidar um

objeto, tal como um objeto gerado dinamicamente, armazenado no cache. Como tal, a invalidação de objetos armazenados em cache pode ser controlada em tempo real, ao invés de se usarem períodos de tempo de expiração ou invalidação de configuração predeterminada. Assim, usando-se estas técnicas, a longevidade dos objetos armazenados em cache pode ser controlada a partir de nós de processamento de aplicativo externos, tais como bancos de dados ou servidores de aplicativo de origem. Por exemplo, a aplicação 1250 pode ser configurada para trabalhar com um banco de dados de modo que uma mudança no banco de dados automaticamente dispare um comando de invalidação a partir do banco de dados (ou aplicativo) para a aplicação 1250 para descarte de um objeto ou de objetos em particular.

d. Invalidação de Grupos Usando-se um Comando de Invalidação

[00536] Em uma modalidade adicional, a aplicação 1250 identifica e invalida ao mesmo tempo um grupo de objetos armazenados pelo cache. Os objetos armazenados em uma memória cache tradicional são tratados, cada um, individual e separadamente, pelo cache ao determinar se o objeto está deteriorado. Conforme cada objeto atinge seu tempo de expiração especificado (geralmente conforme regulado pelo servidor e armazenado pelo cache em uma tabela), aquele item é descartado da memória cache. Esta abordagem tradicional é ineficiente e, finalmente, insuficiente, contudo, para lidar de forma bem-sucedida com os desafios que surgem na tentativa de armazenar em cache um conteúdo gerado dinamicamente.

[00537] A figura 29B ilustra uma outra modalidade de um método para manutenção de um cache, tal como um cache de memória de computador, onde a aplicação 1250 tem a capacidade de criar, armazenar e invalidar grupos de objetos relacionados que foram previamente apresentados a partir de um servidor de origem 30. Em

uma visão geral breve, na etapa 3410, um objeto, tal como um objeto gerado dinamicamente apresentado a partir de um servidor de origem 30 é armazenado no cache. Na etapa 3412, o gerenciador de cache 3232 forma um grupo de objetos previamente apresentados armazenados no cache. Em uma modalidade, o grupo pode ser associado a ou identificado por um ou mais determinantes de objeto, conforme será descrito em maiores detalhes abaixo. Na etapa 3414, o gerenciador de cache 3232 mantém um registro do grupo de objetos. Na etapa 3422, o gerenciador de cache 3232 recebe um comando de invalidação para invalidação do grupo de objetos. Na etapa 3432, o gerenciador de cache 3232 marca o grupo de objetos como inválido, em resposta ao comando de invalidação.

[00538] A etapa 3410 é a mesma que na figura 29A, onde um objeto é armazenado no cache da aplicação 1250, tais como objetos gerados dinamicamente previamente apresentados a partir de um servidor de origem 30. Em algumas modalidades, um ou mais objetos podem não ser identificados como armazenáveis em cache, ou de outra forma podem não ter qualquer cache ou informação de controle de cache. Por exemplo, o servidor 30 pode assumir que os objetos gerados dinamicamente não serão armazenados em cache.

[00539] De acordo com a etapa 3412, a aplicação 1250 forma um grupo de um conjunto dos objetos apresentados previamente a partir do servidor de origem 30 e armazenados no cache. Qualquer conjunto adequado ou desejado de objetos pode ser associado a cada outro para a formação de um grupo. Por exemplo, quaisquer objetos gerados dinamicamente gerados para ou associados a uma apresentação de uma página da web podem formar um grupo. Em outras modalidades, um grupo de objetos pode formar um subconjunto de outros grupos de objeto. Em algumas modalidades, o grupo formado de objetos pode ter objetos apresentados a partir do mesmo servidor 30, enquanto em

outras modalidades o grupo formado de objetos tem objetos apresentados a partir de servidores diferentes 30. Em modalidades adicionais, o grupo formado de objetos pode compreender objetos a partir de um cliente 10, objetos a partir de um servidor 30, ou objetos gerados por ou apresentados a partir de clientes 10 e servidores 30. Em uma modalidade, um objeto no grupo é estático, enquanto um outro objeto no grupo é gerado dinamicamente. Em alguns casos, um objeto no grupo não é identificado como armazenável em cache, enquanto um outro objeto no grupo é identificado como armazenável em cache. Em outros casos, os objetos no grupo podem ser logicamente relacionados, de acordo com uma funcionalidade ou um aplicativo provido por um servidor 30. Em um outro caso, os objetos no grupo podem ser relacionados com o mesmo cliente 10 ou o mesmo usuário.

[00540] Na etapa 3414, um registro do grupo de objetos é mantido. Várias técnicas para gravação e manutenção de um registro de um grupo de objetos, ou associação de outra forma de objetos, podem ser usadas na prática de algumas modalidades das operações descritas aqui. Em uma modalidade, o registro pode ser mantido diretamente, por exemplo, em uma tabela de consulta. Em outras modalidades, os registros poderiam ser representados em um formato de tabela de hash. Em algumas modalidades, o gerenciador de cache 3232 mantém a associação de objetos em um banco de dados, ou uma estrutura de dados ou objeto em memória. Em modalidades adicionais, um indicador, uma propriedade ou um atributo de cada objeto no grupo é atribuído ou regulado para um valor identificando o grupo, tal como um valor igual a, identificando ou referenciando o nome ou identificador do grupo, tal como um determinante de objeto de grupo que será descrito em maiores detalhes abaixo. Em algumas modalidades, um grupo de objetos é disposto, colocado ou localizado em uma porção de memória cache identificada como mantendo o

grupo.

[00541] Na etapa 3422, um comando de invalidação é recebido na aplicação 1250 ou no gerenciador de cache 3232. De acordo com a modalidade descrita na figura 29B, o comando de invalidação identifica que um ou mais objetos são inválidos ou de outra forma estão deteriorados. Em algumas modalidades, o comando de invalidação referencia, identifica ou especifica um nome ou identificador do grupo de objetos. Em uma modalidade, o comando de invalidação compreende uma requisição de invalidação única para invalidação de todos os objetos no grupo. Em uma outra modalidade, o comando de invalidação identifica um objeto no grupo a invalidar. Em outras modalidades, o comando de invalidação compreende uma pluralidade de requisições de invalidação para invalidação de uma pluralidade de objetos no grupo.

[00542] De acordo com a etapa 3432, o grupo de objetos previamente apresentados é marcado como inválido se um comando de invalidação referenciar, identificar ou especificar um objeto do grupo como inválido, cada objeto no grupo como inválido ou o grupo como inválido. Em algumas modalidades, se o comando de invalidação identificar um objeto no grupo como inválido, o gerenciador de cache 3232 marcará o objeto como inválido. Em outras modalidades, se o comando de invalidação identificar um objeto no grupo como inválido, o gerenciador de cache 3232 marcará o grupo de objetos como inválido ou cada objeto no grupo como inválido. Ainda em modalidades adicionais, o gerenciador de cache 3232 pode invalidar apenas o grupo de objetos quando uma pluralidade de objetos for identificada como inválida através de um ou mais comandos de invalidação. Em uma outra modalidade, o comando de invalidação pode especificar um nome ou um identificador do grupo, e o gerenciador de cache 3232 marca o grupo como inválido, ou cada

objeto no grupo como inválido.

[00543] Em uma modalidade, a aplicação 1250 ou o gerenciador de cache 3232 descarta a partir da memória cache um grupo de objetos que foi marcado como inválido. Em algumas modalidades, os objetos no grupo podem ser descartados a partir da memória cache apenas quando cada objeto no grupo for marcado como inválido. Em outras modalidades, se um objeto do grupo tiver sido marcado como inválido, então, o grupo inteiro será descartado. Em uma outra modalidade, o grupo de objetos, ou qualquer objeto no grupo, marcado como inválido pode ser descartado mediante o recebimento de uma requisição para o grupo de objetos, ou qualquer objeto no grupo, por um cliente 10. Em outras modalidades, o grupo de objetos, ou qualquer objeto no grupo, marcado como inválido, pode ser descartado, mediante o recebimento de uma resposta de um servidor 30 provendo um ou mais novos objetos no grupo.

[00544] Um exemplo das modalidades descritas acima se segue. Os aplicativos de gerenciamento de recurso de consumidor ("CRM") são usados por muitas companhias para acompanhamento e avaliação de todos os aspectos de gerenciamento de recurso. Frequentemente, os aplicativos de CRM são implementados e acessados por redes privadas ou públicas, incluindo a Internet. Estes aplicativos, os quais provêem acesso a grandes quantidades de dados frequentemente estão sendo acessados, desse modo se beneficiando do armazenamento em cache dos dados gerados por esses aplicativos. Por exemplo, relatórios de vendas frequentemente são gerados e apresentados para usuários conectados remotamente. Estes relatórios de vendas são montados por um aplicativo relevante através da compilação de dados de uma informação de venda que é postada para tais servidores de aplicativo e/ou seus bancos de dados subjacentes. Como muitos usuários requisitam o mesmo documento

(isto é, um dado relatório de vendas), sem armazenamento em cache, o servidor de aplicativo deve gerar o objeto para cada requisição. Se, contudo, tais objetos puderem ser armazenados no cache, então, o processamento de aplicativo banco de dados será conservado, incluindo uma largura de banda potencialmente valiosa, já que o cache é colocado mais próximo dos clientes requisitando.

[00545] O desafio para armazenamento em cache desses objetos surge porque a cada vez em que uma nova venda é postada para o aplicativo rodando no servidor de origem (ou em seu banco de dados subjacente), a informação no relatório de vendas precisa ser atualizada. Como resultado, todos os relatórios de vendas que podem ter sido armazenados em quaisquer caches suportando estes servidores de aplicativo devem ser invalidados e o conteúdo descartado da memória cache. A abordagem tradicional para armazenamento em cache, contudo, não tem uma forma de determinar de forma acurada quando a mudança no banco de dados subjacente ou aplicativo está para ocorrer e, portanto, não pode avaliar de forma razoável o frescor de um conteúdo dinâmico. Toda vez em que uma mudança ocorre em um banco de dados ou aplicativo ou servidor de origem, o cache tem que ser capaz de identificar que a mudança foi feita, e qual grupo de objetos deve ser invalidado como uma consequência dessa mudança. A geração de comandos de invalidação que contêm determinantes de objeto ligados a grupos de objetos apresentados previamente, conforme descrito acima, pode se adequar a esta necessidade.

e. Identificação de Determinantes de Objeto em uma Requisição ou Resposta de Cliente

[00546] Uma modalidade também se dirige à necessidade de se ser capaz de identificar todos os objetos afetados por uma mudança de estado no servidor de aplicativo de origem 30 (e/ou em um banco de

dados subjacente) pela geração de agrupamentos de objetos e pela implementação de uma invalidação parametrizada. Nesta modalidade, qualquer objeto ou grupo de objetos predefinido pode ser invalidado por uma requisição de HTTP interceptada, por exemplo, a partir de um cliente, que o cache analisa gramaticalmente, de modo a identificar um determinante de objeto. O termo "determinante de objeto" se refere a qualquer informação, dado, estrutura de dados, parâmetro, valor, padrão de dados, requisição, resposta ou comando que referencie, identifique ou especifique um objeto ou um conjunto de objetos, de forma única ou de outra forma. Em algumas modalidades, uma determinação de objeto é um padrão de bytes ou caracteres em uma comunicação que podem ser associados a um objeto ou usados para a identificação de forma única que a comunicação está associada a ou referenciando o objeto. Em uma modalidade, um determinante de objeto indica se uma mudança ocorreu ou ocorrerá, no servidor de origem, para um grupo de objetos apresentados previamente armazenados no gerenciador de cache 3232 com o qual o determinante de objeto está associado. Em algumas modalidades, os objetos em um grupo de objetos estão relacionados pelo fato de eles estarem associados a pelo menos um determinante de objeto. Exemplos específicos não limitativos de determinantes de objeto e ilustrações adicionais de seu uso são descritos mais plenamente abaixo.

[00547] Em algumas modalidades, da presente modalidade, os determinantes de objeto são certos parâmetros predefinidos ou estruturas de dados incluídos ou embutidos em uma requisição ou resposta de cliente. Em outras modalidades, o cliente 10, o servidor 30 ou a aplicação 1250 embute em uma comunicação um ou mais determinantes de objeto, tais como strings predefinidas ou conjuntos de caracteres representando o determinante de objeto. Os

determinantes de objeto indicam se essa requisição terá o efeito de causar uma mudança no estado de objetos armazenados no servidor de origem 30 ou em bancos de dados ligados a ele. Em uma modalidade, a existência do determinante de objeto em uma requisição indica que uma mudança ocorreu ou ocorrerá em um objeto. Em uma outra modalidade, a sintaxe, uma estrutura, um parâmetro ou valor do determinante de objeto indica que uma mudança ocorreu ou ocorrerá em um objeto. Em uma modalidade, o cache recebe uma requisição de objeto de um cliente 10. A requisição pode incluir certos parâmetros ou valores (determinantes de objeto) que o cache reconhece que mudarão o estado do servidor de origem ou do servidor de aplicativo o que, como uma consequência, tornará deteriorados certos objetos relacionados armazenados pelo gerenciador de cache 3232 que tinham sido gerados previamente por tal servidor de origem ou servidor de aplicativo 30. Dependendo da política de invalidação estabelecida pelo usuário, os parâmetros (determinantes de objeto) podem requerer uma invalidação de um ou mais objetos apresentados previamente ou grupos de objetos, pelo servidor de origem, que foram armazenados pelo cache. O cache é configurado para identificar os objetos relevantes que serão afetados pelo cache. O cache é configurado para identificar os objetos relevantes que serão afetados por esta mudança de estado (isto é, aqueles objetos ou grupos de objetos ligados ao determinante de objeto) e invalidar estes objetos através do método marcando cada um dos objetos como inválidos e/ou descartando tais objetos da memória cache.

[00548] A técnica descrita acima é ilustrada na figura 29C. Como com outras modalidades descritas aqui, a etapa 3410 compreende o armazenamento, em cache, de objetos, tais como objetos gerados dinamicamente apresentados previamente a partir de um servidor de

origem. Os objetos poderiam ser gerados por um aplicativo rodando no servidor de origem 30, ou poderiam ser retirados, por exemplo, de um banco de dados acessado pelo servidor de origem 30. Em algumas modalidades, os objetos gerados dinamicamente são identificados como não armazenáveis em cache ou de outra forma não identificados como armazenáveis em cache.

[00549] De acordo com a etapa 3421, o cache intercepta ou recebe de outra forma uma comunicação entre o cliente e o servidor, tal como uma requisição de um cliente ou uma resposta de um servidor. Em algumas modalidades, a requisição é para um objeto específico, o objeto tendo sido previamente apresentado e armazenado no cache. Em uma outra modalidade, a comunicação inclui uma resposta a partir de um servidor tendo um objeto requisitado. Em uma modalidade, essa recepção ou interceptação ocorre de acordo com um protocolo de armazenamento em cache estabelecido e padrões de comunicações. Embora o gerenciador de cache 3232 ou a aplicação 1250 possa ser geralmente descrito como recebendo uma requisição, resposta ou comunicação, ao receber essa requisição, resposta ou comunicação, o cache 3232 ou a aplicação 1250 pode interceptar ou obter por quaisquer meios e/ou mecanismos adequados a requisição, resposta ou comunicação, embora não comunicada diretamente ou de forma explícita para o cache.

[00550] Na etapa 3423, um determinante de objeto é identificado na comunicação interceptada. O gerenciador de cache 3232 pode extrair, interpretar, analisar gramaticalmente, acessar, ler ou processar de outra forma a comunicação interceptada para determinar ou identificar um ou mais determinantes de objeto nas comunicações. Qualquer parâmetro, valor, sintaxe, dado, estrutura ou conjunto de um ou mais caracteres da comunicação pode ser usado para a identificação de um determinante de objeto. Em uma modalidade, o gerenciador de cache

3232 pode identificar o nome ou identificador de um objeto em uma requisição a partir do cliente 10 para o servidor 30, em que o cliente requisite o objeto. Em uma outra modalidade, o gerenciador de cache 3232 pode identificar o nome ou identificador de um primeiro objeto na requisição do cliente 10 ou uma resposta a partir do servidor 30 que indique que uma mudança ocorreu ou ocorrerá em um segundo objeto armazenado no cache. Em outras modalidades, o gerenciador de cache 3232 determina se quaisquer padrões de caracteres na requisição combinam com quaisquer determinantes de objeto associados a um objeto ou grupo de objetos no cache. Em algumas modalidades, um determinante de objeto pode ser determinado para um objeto não atualmente armazenado em cache. Em outras modalidades, um determinante de objeto pode ser determinado para um objeto atualmente marcado como inválido. Em outras modalidades, um determinante de objeto para um objeto requisitado é determinado como estando associado a um determinante de objeto de um objeto armazenado em cache. Ainda em uma outra modalidade, mediante a primeira referência, requisição ou resposta de um objeto em uma comunicação, o gerenciador de cache 3232 estabelece o determinante de objeto identificado como o determinante de objeto para o objeto.

[00551] Pelo recebimento e pela análise gramatical da comunicação, tal como uma requisição de cliente ou uma resposta de servidor, para a identificação de um determinante de objeto, o gerenciador de cache 3232 ou a aplicação 1250 pode efetivamente determinar se é para marcar como inválido um objeto armazenado em cache que foi associado ao determinante de objeto identificado. Assim, de acordo com a etapa 3425, uma determinação é feita quanto a se o determinante de objeto indica uma mudança no objeto armazenado em cache. Em algumas modalidades, o determinante de objeto

identificado pode ser parte de uma comunicação que não altera, modifica ou gera um objeto. Em outras modalidades, o determinante de objeto identificado é uma parte de uma comunicação que indica que uma mudança ocorreu ou ocorrerá no objeto associado ao determinante de objeto. Por exemplo, a comunicação pode ser obter uma requisição para um objeto gerado dinamicamente ou uma requisição de submissão que mudará os dados usados para um ou mais objetos gerados dinamicamente. Em algumas modalidades, a existência do determinante de objeto na comunicação indica que uma mudança ocorreu ou ocorrerá em um ou mais objetos. Em uma outra modalidade, o tipo ou nome de um comando, uma diretiva ou uma instrução na comunicação juntamente com o determinante de objeto indica que uma mudança ocorreu ou ocorrerá em um ou mais objetos. Ainda em uma modalidade adicional, a existência, o valor ou a regulamentação de um parâmetro ou variável de um comando, diretiva ou instrução indica que uma mudança ocorreu ou ocorrerá em um ou mais objetos associados a um determinante de objeto.

[00552] Em outras modalidades, o gerenciador de cache 3232 executa uma função de hash, um algoritmo ou uma operação na comunicação interceptada ou no determinante de objeto para determinar se uma mudança ocorreu no objeto. Em algumas modalidades, o valor de hash é comparado com um valor de hash previamente armazenado para o objeto e, caso diferente, então, o gerenciador de cache 3232 reconhece que o objeto mudou. Ainda em uma outra modalidade, um valor de hash para o objeto pode ser incluído na comunicação ou no determinante de objeto. Em uma modalidade, a comunicação indica que o objeto mudou pelo valor ou pela regulamentação de um parâmetro, tal como um indicador (flag) booleano. Em outras modalidades, um controle de tag de entidade e um mecanismo de validação, conforme descritos em maiores detalhes

abaixo podem ser usados para a identificação de um objeto e para se determinar se o objeto mudou.

[00553] Se uma mudança for indicada, então, na etapa 3431, então, o objeto associado a ou identificado pelo determinante de objeto será marcado como inválido. Em algumas modalidades, um objeto requisitado pela comunicação interceptada é marcado como inválido de acordo com a etapa 3431, e recuperado a partir do servidor de origem 30, de acordo com a etapa 3440. Caso contrário, em outras modalidades, o objeto requisitado é recuperado a partir do cache de acordo com a etapa 3450. Em uma modalidade, qualquer objeto marcado como inválido será descartado do cache.

f. Invalidação de Grupos de Objetos com Base em Determinantes de Objeto

[00554] A modalidade acima descreve o caso de invalidação de um objeto apresentado previamente no gerenciador de cache 3232, com base na identificação de um determinante de objeto na requisição de cliente. Este conceito geral também pode ser usado, em uma outra modalidade, para a identificação e a invalidação de um grupo de objetos com os quais um ou mais determinantes de objeto foram associados. Esta modalidade é ilustrada na figura 29D.

[00555] O método descrito na figura 29D começa da mesma forma que o método da figura 29C. A etapa 3410 compreende o armazenamento, no cache, de objetos, tais como objetos gerados dinamicamente previamente apresentados a partir de um servidor de origem. Em algumas modalidades, um ou mais dos objetos não são identificados como armazenáveis em cache. De acordo com a etapa 3412 e de modo similar à figura 29B, os objetos apresentados previamente são formados em grupos. Em uma modalidade e de acordo com a técnica de determinante de objeto, um grupo de objetos é associado a ou identificado por pelo menos um determinante de

objeto. Conforme descrito mais plenamente abaixo, em algumas modalidades, a associação de grupos com determinantes de objeto depende da natureza e dos detalhes da política de armazenamento em cache de usuários, tal como uma política definida, controlada ou usada pelo agente de política 3236. Em uma outra modalidade, um ou mais determinantes de objeto do grupo compreendem um ou mais determinantes de objeto dos objetos no grupo. Em uma outra modalidade, o determinante de objeto do grupo compreende uma combinação de determinantes de objeto de objetos no grupo.

[00556] De acordo com a etapa 3414, um registro é mantido do grupo, juntamente com seus determinantes de objeto associados, se aplicável. Esta etapa é similar à etapa 3414, ilustrada na figura 29B. Em uma modalidade, o registro e/ou quaisquer determinantes de objeto no grupo são mantidos em uma tabela de consulta. Em outras modalidades, o registro e/ou quaisquer determinantes de objeto no grupo podem ser mantidos em um formato de tabela de hash. A tabela de hash pode ser projetada para armazenar eficientemente chaves não contíguas que podem ter espaços amplos em suas seqüências alfabéticas e numéricas. Em uma outra modalidade, um sistema de indexação pode ser construído no topo de uma tabela de hash. Espessura de material algumas modalidades, o gerenciador de cache 232 mantém a associação de objetos como um grupo de um ou mais determinantes de objeto em um banco de dados, ou uma estrutura de dados ou um objeto em memória. Em modalidades adicionais, um indicador (flag), uma propriedade ou um atributo de cada objeto no grupo é atribuído ou regulado para um valor identificando o grupo, tal como um valor igual a, identificando ou referenciando o nome ou identificador do grupo, ou um determinante de objeto de grupo. Em algumas modalidades, um grupo de objetos é disposto, colocado ou localizado em uma porção de memória cache identificada como

mantendo o grupo. Em uma outra modalidade, um ou mais determinantes de objeto são armazenados em associação com o grupo de objetos.

[00557] As etapas 3421 e 3423 são similares às etapas 3421 e 3423, conforme ilustrado na figura 29C. De acordo com a etapa 3421, o gerenciador de cache 3232 ou a aplicação 1250 intercepta ou recebe de outra forma uma comunicação entre o cliente 10 e o servidor 30, tal como uma requisição de um cliente para um objeto apresentado previamente e armazenado no cache. Em uma modalidade, o gerenciador de cache 3232 intercepta uma requisição do cliente 10 para o servidor 30. Em algumas modalidades, a requisição é para um objeto armazenado em cache. Em outras modalidades, a requisição é uma instrução, um comando ou uma diretiva para o servidor 30 que causará uma mudança em um objeto armazenado em cache, tal como fará com que um objeto seja gerado dinamicamente. Em uma outra modalidade, o gerenciador de cache 3232 intercepta uma resposta de um servidor 30 para o cliente 10 compreendendo ou identificando um objeto armazenado em cache.

[00558] Na etapa 3423, um determinante de objeto é identificado na comunicação interceptada. Conforme citado acima, o determinante de objeto indica se uma mudança ocorreu ou ocorrerá no objeto requisitado, no servidor de origem 30. Contudo, na modalidade da figura 29D, o determinante de objeto pode estar associado a um grupo de objetos. Isto permite uma invalidação eficiente de todos os objetos armazenados no cache que podem ser afetados por um determinante de objeto em particular. Em algumas modalidades, um determinante de objeto de um objeto no grupo é identificado. Em outras modalidades, um determinante de objeto, por exemplo, um determinante de objeto de grupo, para o grupo de objetos é identificado. Em uma outra modalidade, uma combinação de

determinantes de objeto de um ou mais objetos no grupo é identificada.

[00559] Assim, de acordo com a etapa 3427, uma determinação é feita quanto a se o determinante de objeto indica uma mudança no grupo de objetos apresentados previamente. Em algumas modalidades, a existência do determinante de objeto do grupo na comunicação interceptada indica que uma mudança ocorreu ou ocorrerá em um ou mais ou em todos os objetos no grupo. Em outras modalidades, o nome e o tipo de um comando, uma diretiva ou uma instrução na comunicação interceptada indica essas mudanças. Ainda em uma outra modalidade, a existência, o valor ou a regulação de quaisquer parâmetros ou variáveis na comunicação também pode indicar essas mudanças.

[00560] Se, na etapa 3427, o determinante de objeto indicar uma mudança no grupo, então, o grupo de objetos apresentados previamente será marcado como inválido no cache, de acordo com a etapa 3435. Em algumas modalidades, um ou mais ou todos os objetos do grupo são requisitados e recuperados a partir do servidor de origem 30, de acordo com a etapa 3440. Se, na etapa 3427, o determinante de objeto não indicar uma mudança no grupo, então, em algumas modalidades, quaisquer objetos requisitados como parte de uma comunicação interceptada e preferencialmente apresentados e armazenados no cache serão recuperados a partir do gerenciador de cache 3232 de acordo com a etapa 3450. Em uma modalidade, qualquer objeto ou grupo de objetos marcado como inválido pode ser descartado pelo gerenciador de cache 3232 do cache.

g. Designação de Grupos

[00561] O administrador de cache pode designar especificamente quais objetos ficam incluídos em um grupo em particular. Sempre que um objeto for armazenado no cache, o administrador pode tornar

aquele objeto um membro de um dos grupos configurados ou implícitos, dependendo da configuração. Os grupos configurados podem ser com base em configurações que um administrador tenha estabelecido previamente ou, alternativamente, com base em um comportamento de aplicativo e outros dados relacionados a uma invalidação de objeto. Um objeto também pode ser parte de um grupo implícito, se seu grupo configurado for dinâmico. Os objetos no grupo implícito são agrupados pelo valor dos parâmetros de invalidação significativos.

[00562] Ao se permitir um agrupamento muito flexível de objetos, um cache pode obter um nível de flexibilidade e coordenação na invalidação que é necessário para se armazenar em cache um conteúdo gerado dinamicamente. O cache pode invalidar um grupo muito específico de objetos simultaneamente, desse modo se tornando o cache mais passível de responder à necessidade freqüente de invalidação de conteúdo gerado dinamicamente. No momento em que o cache atribui um objeto a um grupo, o grupo determina várias coisas em relação àquele objeto, incluindo os parâmetros de invalidação e os determinantes de acerto, de modo a se associarem um ou mais determinantes de objeto a ele.

[00563] No exemplo de gerenciamento de recurso de consumidor ("CRM"), o administrador de cache pode pré-designar cada um dos agrupamentos. Por exemplo, o administrador configura o cache para agrupar cada um dos departamentos de vendas por nome. Assim, o administrador pode designar um departamento de automóveis, um departamento de motocicletas, etc., e a cada vez que um determinante de objeto for reconhecido em uma requisição vindo para o cache, o cache então pode invalidar todos os objetos em um grupo designado ligado a um departamento apropriado através do determinante de objeto.

h. Agrupamento Baseado em Regra

[00564] Alternativamente, o administrador de cache pode estabelecer regras que permitam que a aplicação de cache determine correndo quais objetos incluir em um grupo em particular ou grupos. Esses agrupamentos baseados em regra podem se apoiar na designação de grupos em virtude de regras estabelecidas que ligam o objeto aos determinantes de objeto significativos que o cache utiliza para a criação dos grupos relevantes. Um exemplo desta abordagem pode envolver a configuração do pôr em cada grupo.

[00565] De novo voltando ao exemplo de CRM, uma regra pode declarar que cada subdivisão do Departamento de Vendas que é estabelecida no aplicativo deve ser reconhecida pelo cache como seu próprio agrupamento. Desta forma, os agrupamentos podem ser criados sem o administrador de cache ter que identificar especificamente cada agrupamento, mas permite que o cache determine com base nas regras relevantes. Esta técnica cria uma forma mais flexível e freqüentemente de trabalho menos intensivo para a designação de agrupamentos. O administrador de cache poderia configurar uma regra que declarasse que toda subdivisão do departamento de vendas (isto é, vendas\auto, vendas\motocicleta, etc.) deve ter gerado um novo agrupamento pelo cache. Conforme uma requisição do Departamento de Vendas de Auto é processada e retornada pelo aplicativo através do cache, o cache pode reconhecer cada subagrupamento de vendas e automaticamente criar um agrupamento para ele, com base na regra pré-configurada.

[00566] A regra pode ser implementada pelo cache a cada vez em que ele vir uma nova requisição para um objeto do tipo relatório/vendas/auto ou relatório/vendas/motocicleta, etc. Este processo então pode ser repetido quando de uma requisição de Departamento de Vendas de Motocicletas mostrando que é um

subagrupamento do Departamento de Vendas, então, o Departamento de Vendas de Bicicletas e assim por diante, conforme o cache reconhecer estes subgrupos e estabelecer um agrupamento de objeto para cada um deles. Quando uma requisição de invalidação conhecida vier do cache ligada a um destes agrupamentos ou, se um determinante de objeto relevante for identificado em uma requisição de cliente (por exemplo, uma postagem de um relatório de vendas para vendas/motocicleta do Departamento de Vendas de Motocicletas encontrada na análise gramatical da requisição), o cache saberá invalidar todos os objetos armazenados em cache no Agrupamento de Departamento de Vendas de Motocicletas.

[00567] Desta forma, quando um cache reconhece que uma mudança ocorreu ou ocorrerá nos dados apresentados pelo aplicativo (porque o cache reconhece que o conteúdo de uma requisição recebida pelo cache disparará uma mudança no aplicativo ou por causa da ocorrência de alguma mudança externa), a técnica acima permite que o cache identifique de forma rápida e simples quais objetos requerem uma invalidação através do processo de agrupamento. Desta forma, o cache é capaz de invalidar grandes números de objetos gerados dinamicamente que não estão mais novos por causa de mudanças no estado do aplicativo ou do banco de dados.

[00568] A capacidade do cache de armazenar de forma bem-sucedida e apresentar a partir de sua memória cache um conteúdo gerado dinamicamente também pode ser melhorada com um agente estatístico inteligente que examine o padrão tráfego de requisição e de resposta, de modo a determinar, por um período de tempo, o conjunto de objetos que proveria o maior benefício de armazenamento em cache. O agente pode ser integrado na aplicação de cache em si, ou rodar em um computador em separado como uma heurística para

seleção de algum subconjunto de objetos para investigação adicional para determinação da adequabilidade para um armazenamento em cache dinâmico.

i. Uso Adicional de Determinantes de Objeto

[00569] Conforme descrito acima, os determinantes de objeto podem ser de qualquer estrutura de dados que indique se uma mudança ocorreu ou ocorrerá no servidor de origem, para o grupo de objetos apresentados previamente armazenados no cache com o qual o determinante de objeto estiver associado. Os determinantes de objeto poderiam ser regulados com base em valores de string predefinidos embutidos na requisição. Por exemplo, quando uma requisição vem com um certo USERID, o USERID pode ser ligado a um grupo de objetos na memória cache que deve ser invalidado a cada vez que uma postagem ou uma outra requisição vier daquele certo USERID. Candidatos potenciais para determinantes de objeto também poderiam incluir usar identificadores de serviço do servidor que originalmente apresentou o objeto. O identificador de serviço contém um endereço de IP de serviço, uma porta TCP e um identificador de serviço presentes na requisição de HTTP.

[00570] Um outro determinante de objeto potencial presente na requisição é um localizador de recurso uniforme ("URL") de requisição. No caso de armazenamento em cache de objetos estáticos, o URL de requisição é suficiente, tipicamente, para a identificação de forma única do objeto. Para requisições para conteúdo gerado dinamicamente, contudo, a informação presente no URL pode não ser suficiente para identificar o objeto armazenado em cache. O cache, portanto, deve inspecionar uma outra informação na requisição para encontrar determinantes de objeto incluindo nos cabeçalhos de HTTP, em um cabeçalho de cookie ou em outros cabeçalhos de HTTP personalizados. O cache adicionalmente pode procurar um

subconjunto de informação de parâmetro relevante em uma variedade de outros locais na requisição de cliente, incluindo, sem limitação: na string de consulta de URL, no corpus de POST, em um cabeçalho de cookie, ou em quaisquer outros cabeçalhos de requisição ou de resposta.

[00571] O problema na análise gramatical de um URL quanto a determinantes de objeto é que o URL e outros cabeçalhos podem conter uma grande quantidade de informação além do que é relevante para a decisão do cache. O cache, portanto, deve ser capaz de analisar gramaticamente uma quantidade bem grande de informação para ser capaz de identificar os determinantes de objeto apropriados. Além disso, os dados no cabeçalho freqüentemente são ordenados arbitrariamente, significando que não há formas padronizadas desses dados serem colocados no cabeçalho de HTTP e, portanto, uma comparação simples freqüentemente é insuficiente para a localização dos determinantes de objeto relevantes nessa string.

[00572] Se não houver uma política pré-configurada para combinação de um determinante de objeto em particular a um objeto relevante ou grupo de objetos armazenados na memória cache, o cache ainda pode, em uma outra modalidade, fazer uma determinação como essa. Por exemplo, o cache pode examinar e analisar gramaticalmente vários aspectos da requisição para descobrir se quaisquer outros determinantes de objeto podem ser encontrados nessa requisição e usados para ligação dessa requisição a objetos em particular armazenados na memória cache que devem ser invalidados. Alternativamente, também se poderia permitir que o cache examinasse uma requisição quanto a certos determinantes de objeto que o cache determinasse, com base em certa heurística predefinida, que podem ser ligados de forma significativa a objetos ou grupos de objeto em particular. Por exemplo, quando a requisição vem para o cache a partir

de uma atualização de uma agenda associado a um USERID em particular, uma modalidade poderia ser configurada para reconhecer que todos os objetos armazenados em cache com USERID igual ao USERID da requisição de atualização da agenda e que contivessem a agenda do usuário para qualquer dia em particular precisariam ser invalidados.

[00573] O cache também pode assumir que os determinantes de objeto estejam presentes como um grupo de nome = valor ou pares similares em uma ordem não especificada no Cerne de URL, nas consultas presentes no URL, no corpus de POST ou em um cabeçalho de cookie. Em uma modalidade, é assumido que a consulta seja formatada como uma lista de pares de nome = valor. O usuário pode configurar, portanto, quais nomes de parâmetro são significativos. Todo objeto armazenado em cache é chaveado usando-se primeiramente seu URL de acesso. O URL pode parecer com /site/application/special/file.ext?p1=v1 &p2=v2&p3=v3. A parte /site/application/special/file.ext é o cerne de URL. A parte p1=v1 &p2=v2&p3=v3 é a consulta de URL e contém pares de parameter – value (parâmetro – valor). Estes pares de parâmetro – valor também podem estar presentes no corpus POST ou nos cabeçalhos de Cookie.

[00574] Em uma modalidade, o usuário ou administrador estabelece que p1 e p2 devem ser os parâmetros de invalidação ou determinantes de objeto. O cache depois disso automaticamente agrupará objetos que tiverem valores de p1 e p2 combinando. Uma forma de implementação deste agrupamento é mapear p1 e p2 para chaves primárias em tabelas de banco de dados, isto é, objetos identificáveis unicamente na tabela que o cache saberá como referenciar de modo a determinar um status de validação. Para atualização de alguma coisa naquelas tabelas de banco de dados, de modo a se refletir o fato que os dados armazenados no cache não são mais válidos, o cache

especificará novos valores para p1 e p2, e, quando o cache reconhecer esses novos valores da próxima vez em que ele apresentar esse conteúdo, ele saberá invalidar os objetos ligados armazenados em sua memória. O cache, quando encontra uma requisição como essa, ao ver a requisição de atualização sabe que ele tem que invalidar o grupo com valores de p1 e p2 combinando – porque o cache entende que os dados na origem mudarão, desse modo afetando todos os objetos que estiverem relacionados àqueles determinantes de objeto p1 e p2.

[00575] Para se dirigir ao caso mais complexo em que o administrador não pré-configurou os parâmetros específicos embutidos na requisição como determinantes de objeto, o cache pode empregar políticas configuradas por usuário para extração dos determinantes de objeto relevantes a partir da requisição para ajudar na identificação de quando invalidar agrupamentos de objetos. A string de determinante então é usada para a localização do grupo de objetos armazenados no cache e invalidar esses objetos. Estes determinantes de objeto podem ser usados para a configuração do cache para a geração de listas de valores de parâmetro significativos. Se uma requisição de escrita entrando tiver valores combinando para os parâmetros significativos, então, os objetos ligados àqueles nomes de parâmetro devem ser invalidados. Alternativamente, um usuário poderia especificar a ação de arquitetura de política para extração da string de determinante de objeto a partir da requisição. A string de determinante de objeto é extraída a partir da requisição de escrita e todos os objetos com strings de determinante combinando são invalidados. Nesta abordagem alternativa, uma requisição chega ao cache, o cache faz uma determinação quanto a se a string de requisição combina com uma política de invalidação. A política de invalidação especifica objetos nos quais um grupo de conteúdo deve ser invalidado.

[00576] Alternativamente, o cache poderia usar qualquer outra informação de usuário que possa estar presente na requisição de cliente. Conforme citado acima, a integração de autenticação e autorização permite que o cache acesse a informação de usuário. O USERID ou GROUPID poderia ser um dos determinantes no caso de um agrupamento relevante de objetos armazenados em cache ser ligado a um usuário ou grupo de usuários. Embora a informação de usuário freqüentemente seja um determinante de objeto importante, a informação de usuário freqüentemente pode não estar presente na requisição de HTTP. Em uma modalidade adicional, os aspectos de armazenamento em cache dinâmico podem ser combinados com um sistema e método para a integração do cache com uma variedade de outros elementos de formação de rede, incluindo a capacidade de realizar certos tipos de infra-estrutura de autenticação, controle de acesso e auditoria (AAA). Assim, o nível de segurança acordado com os dados que são gerados pelos aplicativos é aplicado aos dados que ao invés disso são apresentados a partir de um cache. Esta técnica permite que os aplicativos armazenem em cache uma informação de acesso controlado sensível que de outra forma não poderia ser armazenada em cache.

[00577] Esta abordagem permite que o cache identifique usuários que não incluem uma informação de usuário identificável na requisição de HTTP, mas que podem ser identificáveis através da abordagem de AAA descrita na patente Integrated Caching. Uma abordagem como essa permite que o cache identifique o usuário relevante para uma requisição em particular através do exame da informação de estado de autorização que pode ser compartilhada a partir do processamento de AAA. Em uma modalidade adicional, a integração permite a aplicação de políticas de segurança a uma informação armazenada no cache, para se evitar que usuários não autorizados acessem uma informação

armazenada no cache.

[00578] Esta abordagem também se dirige ao desafio imposto pelo fato de uma porção significativa de dados gerados dinamicamente requerer que o cliente requisitando esses dados esteja autorizado e seja autenticado antes de o cache poder responder à requisição relevante a partir do cliente. O cache deve ter a capacidade de autorizar requisições feitas pelos usuários autenticados, de modo que os aplicativos possam armazenar em cache os objetos de acesso controlado e pela integração dessa tecnologia de armazenamento em cache dinâmico com uma informação de autenticação e autorização, esta segurança pode ser obtida. O USERID ou GROUPID será um dos determinantes de objeto se os objetos forem personalizados para um usuário ou grupo de usuários. Assim, o nível de segurança acordado com os dados que são gerados pelos aplicativos é aplicado à informação armazenada em cache também. Esta técnica permite que os aplicativos armazenem em cache uma informação de acesso controlado sensível, que de outra forma não poderia ser armazenada em cache.

[00579] Finalmente, uma outra informação, tal como hora do dia, estado do banco de dados na origem, etc., pode ser analisada gramaticalmente a partir da requisição e usada como determinantes de objeto para se determinar se os objetos armazenados no cache ainda são válidos. O cache pode cuidar dessa situação pela configuração de um comportamento de expiração apropriado em agrupamentos de objetos que sejam configurados para serem sensíveis a essas variáveis externas.

[00580] Para se dirigir adicionalmente ao desafio apresentado pelo fato de as requisições por conteúdo dinâmico deverem ser analisadas gramaticalmente e interpretadas pelo cache, o cache de acordo com uma modalidade pode limitar quais parâmetros são julgados como

sendo determinantes de objeto relevantes para o cache. Desta forma, a taxa de sucesso para a apresentação de objetos a partir do cache ao invés do encaminhamento dessas requisições para o servidor de aplicativo aplicável pode ser melhorada. A título de exemplo, uma consulta de requisição a partir de um cliente pode conter um parâmetro de cidade e um de estado. Contudo, o cache pode ser configurado para se conformar às exigências do aplicativo para o qual o cache está armazenando um conteúdo para reconhecer que a resposta pode ser apresentada a requisições vindo de clientes que a consulta mostra virem de todos os clientes em um dado estado, sem consideração do valor de cidade. Para esta finalidade, o parâmetro de cidade não é relevante, e o cache poderia reconhecer este fato. Uma modalidade alternativa envolve a configuração do cache de modo que uma resposta possa ser apresentada a partir do cache, se apenas o parâmetro de cidade fizer uma combinação, independentemente do que for especificado para o parâmetro de estado.

[00581] Em resumo, o cache implementa uma combinação de objeto parametrizada generalizada. Nesta abordagem, o cache é configurado para reconhecer o subconjunto de informação na requisição que será útil como determinantes de objeto, e que estiverem ligados a um objeto em particular, de modo que, quando esses determinantes de objeto forem reconhecidos, o cache possa utilizar a presença (ou, inversamente, a ausência desses determinantes) ao avaliar se o objeto ou grupo de objetos permanece novo e capaz de ser apresentado a partir do cache. O cache mantém uma tabela que ele consulta a cada vez em que uma requisição chega para checar em relação aos parâmetros configurados, para determinar se os dados requisitados permanecem novos, e o que também permite que o cache combine os dados relevantes com o objeto apropriado armazenado na memória cache.

j. Números de Encarnação

[00582] Ainda em uma outra modalidade, o cache pode utilizar números de encarnação para invalidação de um grupo de objetos. Quando um cache precisa mudar o estado de cada um de um grupo de objetos de uma vez, por causa de uma mudança no estado na origem, os números de encarnação provêem uma técnica simples para se efetuar esta invalidação. Ao passo que a identificação de cada objeto e a mudança do estado individualmente é uma abordagem ineficiente para se garantir o frescor de dados armazenados em um cache, o uso de números de encarnação permite uma abordagem muito mais simples e efetiva para a invalidação de grupos de objetos. A presente modalidade descreve como cada objeto aponta para uma estrutura de dados que representa o grupo e, portanto, o servidor precisa apenas enviar um comando que mude o estado na estrutura de dados para o grupo. Quando uma requisição subsequente para um objeto armazenado em cache chega de um cliente, o cache deve primeiramente descobrir se o estado mudou. Para fazê-lo, ele consulta a estrutura de dados para referenciar se o estado mudou para o grupo.

[00583] De modo a se implementar a estrutura de dados efetivamente, o cache deve ser capaz de determinar se é para consultar quanto a uma mudança de estado. Portanto, o cache deve ser capaz de determinar se ele já olhou para uma mudança de estado no grupo ou não. Isto é onde os números de encarnação são úteis. O cache associa objetos gerados dinamicamente em grupos de conteúdo. Cada um destes grupos de conteúdo pode ser representado através de um processo de consulta de tabela de hash com um valor de índice em particular ou "número de encarnação" contido em uma estrutura de dados. Após isso, sempre que o cache recebe uma requisição de cliente que o cache reconhece como causando uma mudança de estado, o cliente analisa gramaticalmente a requisição de

cliente quanto a parâmetros relevantes, realiza a consulta de hash com base nos determinantes de objeto reconhecidos, e incrementa o índice ou o número de encarnação na estrutura de dados. A cada vez em que um objeto armazenado em um agrupamento designado é requisitado por um cliente, o cache executa o algoritmo de hash no objeto, e o compara com o valor armazenado original na estrutura de dados para esse grupo de conteúdo. Se o valor armazenado for o mesmo que o número calculado pelo cache para esse objeto, então, o cache sabe que o conteúdo permanece novo e pode ser apresentado para o requisitante. No caso em que o cache detecta uma discrepância entre o número de encarnação atual calculado para esse objeto e o número armazenado para esse grupo de conteúdo na estrutura de dados, o cache sabe que o objeto armazenado não está mais fresco. O cache então invalida o objeto armazenado e envia a requisição para o servidor de aplicativo. Quando a resposta retorna, a aplicação de cache armazenará a nova resposta na memória cache e ligará essa resposta de volta para a nova estrutura de dados. Após isso, a cada vez em que o cache recebe uma requisição para um objeto naquele agrupamento, o cache pode fazer a comparação e assumindo que nenhuma mudança adicional tenha sido feita na estrutura de dados, o cache pode apresentar o objeto recém armazenado.

[00584] Pela utilização de uma invalidação de um grupo de objetos desta forma, o cache é capaz de invalidar muito rapidamente – e o tempo gasto é constante, independentemente do número de objetos invalidados. Através deste processo mais rápido e mais eficiente de invalidação, as técnicas permitem que o cache lide mais efetivamente com objetos gerados dinamicamente. A abordagem permite que as aparelhagens de cache que ficam na frente de aplicativos armazenem mais agressivamente e apresentem objetos gerados dinamicamente sem a apresentação de um conteúdo inválido ou deteriorado, por

causa das mudanças rápidas nesses dados. A modalidade permite que o cache apresente dados que mudem freqüentemente ou de forma não previsível, desse modo melhorando a performance do cache. O cache também é capaz de invalidar objetos e um grupo de objetos armazenados na memória usando comandos de usuário e também pelo exame e pelo agrupamento de vários tipos de tráfego da web.

2. Agrupamento de Conexão

[00585] Em uma modalidade, uma aplicação de rede 1250 (também referida aqui como uma unidade de interface 1250) libera os servidores 30 de muito da carga de processamento causada pela abertura e pelo fechamento repetidamente de conexões para clientes pela abertura de uma ou mais conexões com cada servidor e mantendo essas conexões, para se permitir acessos a dados mais repetidos pelos clientes através da Internet. Esta técnica é referida aqui como um "agrupamento de conexão".

[00586] Para completitude, a operação de agrupamento de conexão é brevemente descrita em seguida, com referência à figura 30. O processo começa na figura 30 quando um cliente 10 requisita acesso a um dos servidores no grupo de servidor abrigado pela unidade de interface 1250. Uma conexão é aberta entre a unidade de interface 1250 e o cliente requisitando, e a unidade de interface 1250 recebe a requisição do cliente para acesso ao servidor, conforme mostrado na etapa 4302. A unidade de interface 1250 determina a identidade do servidor requisitado, conforme mostrado na etapa 4304. Em uma modalidade, isto é realizado pelo exame do endereço de rede de destino especificado pela requisição de cliente. Em uma outra modalidade, isto é realizado pelo exame do endereço de rede e do nome de caminho especificados pela requisição de cliente.

[00587] Após a determinação da identidade do servidor 30 para o

qual a requisição de cliente deve ser dirigida, a unidade de interface 1250 determina se uma conexão livre (isto é, uma que não esteja em uso) para o servidor já está aberta, conforme mostrado na etapa 4306. Se assim for, o processamento retoma na etapa 4310. Caso não, a unidade de interface 1250 abre uma conexão para o servidor, conforme mostrado na etapa 4308. A unidade de interface 1250 então traduz a requisição de cliente e a passa para o servidor, conforme mostrado na etapa 4310, e conforme descrito mais plenamente com respeito à figura 31 abaixo. Após um processamento de servidor, a unidade de interface recebe uma resposta a partir do servidor, conforme mostrado na etapa 4312. A resposta de servidor é traduzida e passada para o cliente requisitante, conforme mostrado na etapa 4314 e descrito adicionalmente abaixo. Finalmente, a unidade de interface 1250 fecha a conexão com o cliente, conforme mostrado na etapa 4316. Contudo, a conexão entre a unidade de interface 1250 e o servidor não é desconectado. Pela manutenção das conexões abertas com os servidores e pela abertura e fechamento de conexões com o cliente, conforme necessário, a unidade de interface 1250 libera os servidores 30 de quase todos os problemas de carregamento de conexão associados aos clientes apresentando pela Internet.

[00588] Conforme será discutido adicionalmente abaixo, algumas modalidades estão relacionadas à etapa 4316, onde a unidade de interface 1250 fecha a conexão com o cliente 10. Há vários cenários que resultam em uma unidade de interface 1250 fechando a conexão com o cliente. Por exemplo, o cliente pode iniciar um comando FIN (finish – terminar) ou um comando RST (reset – reinicializar). Em ambos estes cenários, a unidade de interface 1250 espera até receber um destes comandos, antes de perder a conexão entre ela mesma e o cliente. As ineficiências com um agrupamento de conexão ocorrem quando o cliente não está usando ou terminou a conexão, mas não

retransmite esta informação para a unidade de interface 1250 por um período de tempo. Devido ao fato de a unidade de interface 1250 estar esperando por um comando a partir do cliente, de modo a reusar a conexão para um outro cliente, a conexão fica presa desnecessariamente.

[00589] Conforme será explicado em maiores detalhes abaixo, o Protocolo de Transferência de Hipertexto (HTTP) 1.1 (por default) e HTTP 1.0 (com a Técnica de Connection: Keep-Alive – Conexão: Manter Ativa) permite que o cliente e/ou a unidade de interface 1250 mantenham a conexão aberta com o servidor, mesmo após o recebimento de uma resposta de servidor a uma requisição. O cliente e/ou a unidade de interface 1250 podem emitir, então, outras requisições através da mesma conexão, imediatamente ou após um tempo considerável (ou "tempo para pensar"). Um cliente está no "tempo para pensar" quando o operador humano do cliente está decidindo o próximo link no navegador a clicar, e assim por diante. Isto pode resultar em as conexões serem mantidas pelo servidor, embora o servidor não esteja processando quaisquer requisições através das conexões. Aqui, os administradores de servidor podem ser forçados a se proteger contra conexões simultâneas em demasia no servidor prela regulagem de um tempo de Keep-Alive após o que a conexão a qual estava inativa ou em um "tempo para pensar" é fechada. Uma modalidade permite que a conexão com o servidor seja usada pelo cliente 10', enquanto o cliente 10 está "pensando". Obviamente, se o cliente 10' fizer uma requisição quando o cliente 10 estiver usando a conexão de servidor, então, o cliente 10' deverá usar uma conexão diferente com o servidor. Contudo, a eficiência do agrupamento de conexão de uma modalidade é realizada quando um número muito pequeno de conexões é excedido e se move para o caso geral. O caso geral quando 'n' conexões de cliente podem ser estatisticamente

multiplexada em 'm' conexões de servidor, onde 'n' é maior do que 'm'.

[00590] A figura 31 é um fluxograma que descreve a operação de uma modalidade de tradução de requisições de cliente e servidor, conforme mostrado nas etapas 4310 e 4314 (figura 30). Em uma modalidade, o tráfego de mensagem é na forma de pacotes de TCP/IP, uma suíte de protocolo que é bem conhecida na técnica. A suíte de protocolo de TCP/IP suporta muitas aplicações, tais como Telnet, Protocolo de Transferência de Arquivo (FTP), e-mail e HTTP. A modalidade é descrita em termos do protocolo de HTTP. Contudo, os conceitos se aplicam igualmente bem a outras aplicações de TCP/IP, conforme será evidente para alguém versado na técnica, após uma leitura deste relatório descritivo.

[00591] Cada pacote de TCP inclui um cabeçalho de TCP e um cabeçalho de IP. O cabeçalho de IP inclui um endereço de IP de fonte de 32 bits e um endereço de IP de destino de 32 bits. O cabeçalho de TCP inclui um número de porta de fonte de 16 bits e um número de porta de destino de 16 bits. O endereço de IP de fonte e o número de porta, coletivamente referidos como o endereço de rede de fonte, identificam unicamente a interface de fonte do pacote. Da mesma forma, o endereço de IP de destino e o número de porta, coletivamente referidos como o endereço de rede de destino, identificam unicamente a interface de destino para o pacote. Os endereços de rede de fonte e de destino do pacote identificam unicamente uma conexão. O cabeçalho de TCP também inclui um número de seqüência de 32 bits e um número de reconhecimento de 32 bits.

[00592] A porção de TCP do pacote é referida como um segmento de TCP. Um segmento de TCP inclui um cabeçalho de TCP e um corpo. A parte de corpo do segmento de TCP inclui um cabeçalho de HTTP e a mensagem. Há dois mecanismos para a determinação do

comprimento da mensagem, incluindo um com base em uma codificação de transferência fragmentada e um outro com base no comprimento de conteúdo. Um arquivo de cabeçalho de comprimento de conteúdo é encontrado no cabeçalho de HTTP. Se um campo de cabeçalho de comprimento de conteúdo estiver presente, seu valor em bytes representará o comprimento do corpo de mensagem. Alternativamente, se um cabeçalho de codificação de transferência fragmentada estiver presente no cabeçalho de HTTP, e indicar que a codificação de transferência "fragmentada" foi aplicada, então, o comprimento da mensagem será definido pela codificação fragmentada. A codificação fragmentada modifica o corpo de uma mensagem, de modo a transferir a mensagem como uma série de fragmentos, cada um com seu próprio indicador contido no campo de tamanho de fragmento.

[00593] Conforme será discutido em detalhes abaixo, uma modalidade utiliza o parâmetro de comprimento de conteúdo e/ou o cabeçalho de codificação de transferência fragmentada para aumento da eficiência de agrupamento de conexão entre servidores e clientes ao se evitar a situação em que o cliente está no "tempo para pensar". Sem esta modalidade, a unidade de interface 1250 espera por um comando do cliente, antes de reusar a conexão para um outro cliente ou a conexão expirar quando a conexão tiver ficado inativa por tempo demais.

[00594] O número de seqüência de 32 bits, mencionado acima, identifica o byte na string de dados a partir do TCP de envio para o TCP de recepção que o primeiro byte de dados no segmento de TCP representa. Uma vez que todo byte que é trocado é numerado, o número de reconhecimento contém o próximo número de seqüência que o remetente do reconhecimento espera receber. Isto é, portanto, o número de seqüência mais um dos últimos bytes de dados recebidos

de forma bem-sucedida. A soma de verificação cobre o segmento de TCP, isto é, o cabeçalho de TCP e os dados de resposta (ou corpo). Isto é um campo obrigatório que deve ser calculado e armazenado pelo remetente e, então, verificado pelo receptor.

[00595] De modo a rotear de forma bem-sucedida um pacote chegando de um cliente para o servidor pretendido, ou para rotear um pacote saindo de um servidor para um cliente, a unidade de interface 1250 emprega um processo conhecido como "tradução de endereço de rede". A tradução de endereço de rede é bem conhecida na técnica, e é especificada pela requisição para comentários (RFC) 1631, a qual pode ser encontrada no URL <http://www.safety.net/RFC1631.txt>.

[00596] Contudo, de modo a dividir sem emendas as conexões de cliente e servidor, uma nova técnica de tradução foi descrita em detalhes no Pedido de Patente U.S. comumente possuído Nº 09/188.709, depositado em 10 de novembro de 1998, intitulado "Internet Client-Server Multiplexer", referido aqui como "multiplexação de conexão". De acordo com esta técnica, um pacote é traduzido pela modificação de seu número de seqüência e do número de reconhecimento no nível de protocolo de TCP. Uma vantagem significativa desta técnica é que nenhuma interação de camada de aplicativo é requerida.

[00597] Com referência à figura 31, o endereço de rede do pacote é traduzido, conforme mostrado na etapa 4402. No caso de um pacote entrando (isto é, um pacote recebido a partir de um cliente), o endereço de rede de fonte do pacote é mudado para aquele de uma porta de saída de unidade de interface 1250, e o endereço de rede de destino é mudado para aquele do servidor pretendido. No caso de um pacote saindo (isto é, um recebido a partir de um servidor), o endereço de rede de fonte é mudado daquele do servidor para aquele de uma

porta de saída de unidade de interface 1250, e o endereço de destino é mudado daquele da unidade de interface 1250 para aquele do cliente requisitante. Os números de seqüência e números de reconhecimento do pacote também são traduzidos, conforme mostrado nas etapas 404 e 406 e descrito em detalhes abaixo. Finalmente, a soma de verificação de pacote é recalculada para consideração destas traduções, conforme mostrado na etapa 4408.

[00598] Conforme mencionado acima, uma modalidade está relacionada especificamente a um aparelho, método e produto de programa de computador para se agruparem eficientemente conexões de cliente – servidor de rede através do parâmetro de comprimento de conteúdo e/ou do cabeçalho de codificação de transferência fragmentada para aumento da eficiência de agrupamento de conexão entre servidores e clientes. O aumento na eficiência é o resultado de se evitar ocupar a conexão enquanto o cliente está no "tempo para pensar". Em uma modalidade, os parâmetros de comprimento de conteúdo são usados para se determinar o comprimento da mensagem. Em uma outra modalidade, a codificação de transferência fragmentada é usada para se determinar o comprimento da mensagem. As duas modalidades serão descritas em seguida com referência às figuras 32 e 33, respectivamente.

[00599] A figura 32 ilustra a porção de TCP de um pacote de TCP denominada o segmento de TCP 4500. O segmento de TCP 4500 inclui um cabeçalho de TCP 4502 e um corpo 4504. O corpo 4504 contém, dentre outras informações, um cabeçalho de HTTP e a mensagem. O parâmetro de comprimento de conteúdo 4506 é encontrado no cabeçalho de HTTP. Como uma modalidade utiliza o parâmetro de comprimento de conteúdo 4506 para prover um agrupamento de conexão mais eficiente é descrito abaixo, com referência às figuras 35 e 36.

[00600] A figura 33 ilustra a porção de TCP de um pacote de TCP denominado o segmento de TCP 4600. Conforme declarado acima, se um cabeçalho de codificação de transferência fragmentada estiver presente no cabeçalho de TCP e indicar que a codificação de transferência "fragmentada" foi aplicada, então, o comprimento da mensagem será definido pela codificação fragmentada. A codificação fragmentada modifica o corpo de uma mensagem, de modo a transferir a mensagem como uma série de fragmentos, cada um com seu próprio indicador contido no campo de tamanho de fragmento. O segmento de TCP 4600 inclui um cabeçalho de TCP (não mostrado) e um corpo. O corpo contém, dentre outras informações, um cabeçalho de HTTP 4602A-4602C e a mensagem. O cabeçalho de HTTP 4602A-4602C é compreendido por sete campos de tamanho de fragmento 4606A a 4606G; e seis dados de mensagem de fragmento 4604A a 4604F.

[00601] Os campos de tamanho de fragmento 4606A a 4606G são ligados em conjunto, conforme ilustrado na figura 33. O campo de tamanho de fragmento 4606A indica o comprimento da mensagem no dado de mensagem de fragmento 4604A, o campo de tamanho de fragmento 4606C indica o comprimento da mensagem no dado de mensagem de fragmento 4604C, e assim por diante. O último campo de tamanho de fragmento 4606G sempre contém o valor de comprimento zero indicando que não há mais dados de mensagem a seguir. Isto é uma indicação que toda a mensagem foi enviada para o cliente. Como uma mensagem utiliza os campos de tamanho de fragmento 4606A a 4606G para prover um agrupamento de conexão mais eficiente é descrito abaixo, com referência às figuras 37 e 38. É importante notar que o segmento de TCP 4600 na figura 33 é para fins de ilustração apenas.

[00602] Antes da descrição dos detalhes de como uma modalidade

utiliza o parâmetro de comprimento de conteúdo para aumento da eficiência de agrupamento de conexão, o agrupamento de conexão, conforme é descrito no Pedido de Patente U.S. Nº 09/188.709, depositado em 10 de novembro de 1998, intitulado "Internet Client-Server Multiplexer", será descrito primeiramente para completitude. A figura 34 é um fluxograma de mensagem que ilustra um agrupamento de conexão. A figura 34 mostra a unidade de interface 1250 conectando dois clientes, C1 e C2, a um servidor S. Os dois clientes C1 e C2 podem compreender quaisquer dos clientes 10 discutidos aqui, e o servidor S pode compreender quaisquer dos servidores 30 discutidos aqui. Em primeiro lugar, a unidade de interface 1250 abre uma conexão com o cliente C1 usando o endereço de rede 1 provido pelo cliente C1, conforme mostrado pelo fluxo 4702. A linha de fluxo 4702 é mostrada como um fluxo de duas vias, porque o protocolo de TCP/IP emprega um cumprimento de estágio múltiplo para a abertura de conexões.

[00603] Uma vez que a conexão seja aberta, a unidade de interface 1250 recebe uma requisição GET do cliente C1 especificando um nome de caminho de /sales/forecast.html, conforme mostrado pela linha de fluxo 704. Devido ao fato de nenhuma conexão livre ser aberta entre a unidade de interface 1250 e o servidor S, a unidade de interface 1250 abre uma conexão com o servidor S. A unidade de interface 1250 mapeia esta requisição para o endereço de rede 2, o qual especifica o servidor S, conforme mostrado pela linha de fluxo 4706. A unidade de interface 1250 também passa a requisição GET para aquele servidor, conforme mostrado pela linha de fluxo 4708. O servidor S responde com a página da web requisitada, conforme mostrado pela linha de fluxo 4710. A unidade de interface 1250 encaminha a página da web para o cliente C1, conforme mostrado pela linha de fluxo 4712. Finalmente, a conexão entre o cliente C1 e a

unidade de interface 1250 é fechada, conforme mostrado pela linha de fluxo 4714. De acordo com o protocolo de TCP/IP, o fechamento de uma conexão de rede pode envolver um processo de estágio múltiplo. Portanto, a linha de fluxo 4714 é mostrada como bidirecional. É importante notar que a unidade de interface 1250 não fecha a conexão com o servidor S, mas, ao invés disso, a mantém aberta para acomodar fluxos de dados adicionais.

[00604] Em seguida, uma conexão é aberta entre a unidade de interface 1250 e o cliente C2, usando o endereço de rede 1 provido pelo cliente C2, conforme mostrado pela linha de fluxo 4716. Em seguida, a unidade de interface 1250 recebe uma requisição GET do cliente C2, especificando a página da web /sales/forecast.html, conforme mostrado pela linha de fluxo 4718. Devido ao fato de uma conexão livre já estar aberta entre a unidade de interface 1250 e o servidor S, é desnecessário que a unidade de interface 1250 acumule o servidor S com a carga de processamento de abertura de uma conexão adicional. A unidade de interface 1250 meramente usa uma conexão aberta livre. A unidade de interface 1250 mapeia a requisição GET para o servidor S, a transfere e a encaminha para o servidor S, conforme mostrado pela linha de fluxo 4720. A unidade de interface 1250 recebe a resposta a partir do servidor S, conforme mostrado pela linha de fluxo 4722, e a encaminha para o cliente C2, conforme mostrado pela linha de fluxo 4724. Finalmente, a unidade de interface 1250 fecha a conexão com o cliente C2, conforme mostrado pela linha de fluxo 4726. Mais uma vez, a unidade de interface 1250 não fecha a conexão com o servidor S. Ao invés disso, a unidade de interface 1250 mantém a conexão aberta para acomodar fluxos de dados adicionais.

[00605] Conforme discutido acima, há vários cenários que resultam em a unidade de interface 1250 fechar a conexão com o cliente C2, conforme mostrado pela linha de fluxo 4724. Por exemplo, o cliente

pode iniciar um comando FIN (terminar), o qual corre uma vez que o cliente tenha recuperado todos os dados requisitados (ou mensagem). O cliente também pode iniciar um comando RST (reinicializar). Além de fechar a conexão entre a unidade de interface 1250 e o cliente, o comando RST resulta em várias operações de preparo serem realizadas para se manter a conexão de lado de servidor em boa ordem. Em particular, o protocolo de TCP garante que o comando RST tenha o número de SEQ (seqüência) correto, de modo que o servidor aceite o segmento de TCP; contudo, não é garantido que o comando RST tenha o número de ACK (reconhecimento) correto. Para se lidar com este cenário, a unidade de interface 1250 mantém um acompanhamento dos bytes de dados enviados pelo servidor e dos bytes reconhecidos pelo cliente. Se o cliente não tiver ainda reconhecido todos os dados pelo servidor, a unidade de interface 1250 calculará os bytes não reconhecidos, e enviará um ACK para o servidor. Mais ainda, a PCB de lado de servidor pode ser colocada em uma fila de expiração para se permitir que quaisquer transferências de dados de servidor drenem.

[00606] Mais ainda, embora não mostrado na figura 34, o servidor também pode fechar uma conexão entre ele mesmo e a unidade de interface 1250. O servidor enviaria um comando FIN para a unidade de interface 1250. Neste caso, a conexão entre o servidor e a unidade de interface 1250 e a conexão entre a unidade de interface 1250 e o cliente serão fechadas.

[00607] Um outro aspecto é maximizar o descarregamento de processamento de conexão do servidor pela minimização das ocasiões em que o servidor fecha a conexão. Há três casos:

(1) A versão de protocolo HTTP/1.1 é usada. Neste caso, nenhum cabeçalho de Keep-Alive explícito é requerido. Por default, o servidor mantém a conexão aberta; é papel do cliente fechar a

conexão. Uma modalidade descarrega o servidor pela reutilização da conexão de lado de servidor. Devido ao fato de ser papel do cliente fechar a conexão, ineficiências com agrupamento de conexão ocorrem, quando o cliente termina a conexão, mas não retransmite esta informação para a unidade de interface 1250 por um período de tempo. Devido ao fato de a unidade de interface 1250 estar esperando por um comando do cliente, de modo a reusar a conexão para um outro cliente, a conexão fica presa desnecessariamente.

(2) A versão de protocolo HTTP/1.0 é usada e o cabeçalho "Connection: Keep-Alive" é provido pelo cliente. Neste caso, o servidor mantém a conexão aberta; é papel do cliente fechar a conexão. Uma modalidade descarrega o servidor pela reutilização da conexão de lado de servidor. Como com a versão de protocolo HTTP/1.1, ineficiências com agrupamento de conexão ocorrem, quando o cliente termina a conexão, mas não retransmite esta informação para a unidade de interface 1250 por um período de tempo.

(3) A versão de protocolo HTTP/1.0 é usada e o cabeçalho "Connection: Keep-Alive" não é provido pelo cliente. Neste caso, o servidor normalmente fechará a conexão após satisfazer plenamente uma requisição GET. Se o servidor fechar a conexão após cada requisição, isto negará à unidade de interface 1250 a oportunidade de reusar a conexão de lado de servidor. Constata-se que muito da Internet ainda usa o HTTP/1.1 sem "Connection: Keep-Alive". Uma nova técnica para se permitir a reutilização de conexões de lado de servidor neste caso importante específico foi descrita em detalhes no Pedido de Patente U.S. comumente possuído Nº 09/188.709, depositado em 10 de novembro de 1998, intitulado "Internet Client-Server Multiplexer". A unidade de interface 1250 inspeciona o pacote GET para detectar esta situação. Quando este caso é detectado, a unidade de interface 1250 insere o "Connection: Keep-Alive" no pacote

GET. Uma vez que isto é feito de forma invisível para o cliente, a unidade de interface 1250 deve manter um acompanhamento do número de "Bytes Added" (Bytes Adicionados) na conexão de lado de servidor. O "Bytes Added" não afeta os números de seqüência no pacote GET, uma vez que o número de seqüência é aquele do primeiro byte. Contudo, a unidade de interface 1250 deve adicionar "Bytes Added" ao número de seqüência de pacotes subseqüentes a partir do cliente para o servidor. Inversamente o servidor reconhecerá os bytes adicionais, mas a unidade de interface 1250 deve subtraí-los, antes de enviar o reconhecimento para o cliente – o qual não sabe que estes bytes foram adicionados.

[00608] Conforme mencionado acima, uma multiplexação de conexão é obtida pela manipulação de números de seqüência e de reconhecimento. Os números de seqüência e de reconhecimento de segmentos recebidos pela unidade de interface 1250 são modificados e mapeados para valores esperados pelo destinatário. Para o cliente, os dados parecem estar vindo do servidor e vice-versa. Por exemplo, se "fluxo de entrada" denotar um segmento recebido pela unidade de interface 1250 e "fluxo de saída" denotar o segmento de saída correspondente, a seqüência e os números de reconhecimento serão mudados da maneira a seguir:

Número de seqüência de fluxo de saída = número de seqüência de fluxo de entrada - número de seqüência de começo de fluxo de entrada + número de seqüência de começo de fluxo de saída

Número de reconhecimento de fluxo de saída = número de reconhecimento de fluxo de entrada - número de reconhecimento de começo de fluxo de entrada + número de reconhecimento de começo de fluxo de saída

[00609] Para se dirigir à adição do cabeçalho "Connection: Keep-Alive" a partir de pacotes de HTTP/1.0, a unidade de interface 1250

mantém um acompanhamento de "Bytes Added" na metade apropriada da conexão – neste caso, no lado de servidor. As fórmulas de número de seqüência e de número de reconhecimento são mudadas, conforme se segue:

Número de seqüência de fluxo de saída = número de seqüência de fluxo de entrada - número de seqüência de começo de fluxo de entrada + número de seqüência de começo de fluxo de saída + bytes adicionados de fluxo de saída
 Número de reconhecimento de fluxo de saída = número de reconhecimento de fluxo de entrada - número de reconhecimento de começo de fluxo de entrada + número de reconhecimento de começo de fluxo de saída + bytes adicionados de fluxo de entrada

[00610] Os exemplos específicos de traduções realizadas usando-se estas equações enquanto se incorpora a técnica de parâmetro de comprimento de conteúdo de uma modalidade para a provisão de um agrupamento de conexão mais eficiente são descritos abaixo com referência às figuras 35 e 36 (com relação ao parâmetro de comprimento de conteúdo) e às figuras 37 e 38 (com relação aos campos de tamanho de fragmento).

[00611] A figura 35 é um fluxograma detalhado que ilustra as traduções de números de reconhecimento e de seqüência realizadas por uma modalidade, enquanto incorpora a técnica de parâmetro de comprimento de conteúdo. O rótulo para cada fluxo na figura 35 é da forma T:S,A(L), onde T representa o tipo de segmento de TCP, S é o número de seqüência, S é o número de seqüência, A é o número de reconhecimento e L é o parâmetro de comprimento de conteúdo. O parâmetro de comprimento de conteúdo descreve o número de bytes de dados na mensagem.

[00612] Os fluxos 4802A a 4802C apresentam um método de

abertura da conexão entre o cliente C1 e a unidade de interface 1250. Cada fluxo representa um segmento de TCP. No segmento de TCP 4802A, o indicador (flag) SYN no cabeçalho de TCP é regulado, indicando uma nova requisição de conexão a partir do cliente C1. O cliente C1 estabeleceu um número de seqüência de começo de 2000 e um número de reconhecimento de 2000. A unidade de interface 1250 responde com um segmento SYN ACK especificando um número de seqüência de começo de 4000 e incrementando o número de reconhecimento para 2001, conforme mostrado pelo fluxo 4802B. Cada entidade (por exemplo, cliente, servidor, unidade de interface) na rede regula seu próprio número de seqüência e/ou número de reconhecimento único, conforme é bem conhecido na técnica. O cliente C1 responde com um segmento ACK especificando um número de seqüência de 2001 e incrementando o número de reconhecimento para 4001, conforme mostrado pelo fluxo 4802C. O cliente C1 então envia um segmento GET especificando um comprimento de 49 bytes, conforme mostrado pelo fluxo 4804.

[00613] Assuma que a unidade de interface 1250 determine que não existe nenhuma conexão aberta livre com o servidor S e, portanto, envia um segmento SYN para o servidor S, especificando um número de seqüência de começo de 1950, conforme mostrado no fluxograma 4806A. O servidor S responde com um segmento SYN ACK especificando um número de seqüência de começo de 6000 e incrementando o número de reconhecimento para 1951, conforme mostrado em 4806B. A unidade de interface 1250 responde com um segmento ACK, conforme mostrado pelo fluxo 8060. A unidade de interface 1250 então encaminha o segmento GET a partir do cliente C1 para o servidor S, após modificar os números de seqüência e de reconhecimento de acordo com as equações de tradução descritas acima, conforme mostrado pela linha de fluxo 4808.

[00614] O servidor S responde com os dados requisitados especificando um número de seqüência de 6001, um número de reconhecimento de 2000 e um parâmetro de comprimento de conteúdo de 999, conforme mostrado pelo fluxo 4810. A unidade de interface 1250 recebe o segmento RESP, traduz os números de seqüência e de reconhecimento e encaminha o segmento RESP para o cliente C1, conforme mostrado pela linha de fluxo 4812A.

[00615] Neste ponto, a unidade de interface 1250 recebe uma requisição pelo cliente C1 para abrir uma conexão. Conforme acima, os fluxos 4816A a 4816C apresentam um método de abertura da conexão entre o cliente C2 e a unidade de interface 1250. De novo, cada fluxo representa um segmento de TCP. No segmento de TCP 4816A, um indicador (flag) SYN no cabeçalho de TCP é regulado, indicando uma nova requisição de conexão a partir do cliente C2. O cliente C2 estabeleceu um número de seqüência de começo de 999 e um número de reconhecimento de 999. A unidade de interface 1250 responde com um segmento SYN ACK especificando um número de seqüência de começo de 4999 e incrementando o número de reconhecimento para 1000, conforme mostrado pelo fluxo 4816B. O cliente C2 responde com um segmento ACK especificando um número de seqüência de 1000 e incrementando o número de reconhecimento para 5000, conforme mostrado pelo fluxo 4816C. O cliente C2 então envia um segmento GET especificando um comprimento de 50 bytes, conforme mostrado pelo fluxo 4818.

[00616] Assuma, neste ponto, que a unidade de interface 1250 não tenha conexões disponíveis para o servidor S que foi usado previamente para o cliente C1, se o cliente C1 tiver terminado com a conexão ou estiver no "tempo para pensar". Ao invés de esperar que o cliente C1 inicie um comando FIN (terminar) ou um comando RST (reinicializar) para liberar a conexão, a unidade de interface 1250 usa o

parâmetro de comprimento de conteúdo para confirmar que todos os dados requisitados foram recebidos pelo cliente C1. Aqui, no fluxo 4812B, a unidade de interface 1250 recebe uma confirmação do cliente C1 que o cliente C1 de fato recebeu todos os dados requisitados. Isto indica para a unidade de interface 1250 que, embora o cliente C1 possa estar em uma pausa por alguma razão, antes de ele enviar um comando FIN ou RST, o cliente C1 terminou a conexão. A unidade de interface 1250 modifica os números de reconhecimento e de seqüência e encaminha o sinal de gatilho RESP ACK para o servidor S, conforme mostrado pelo fluxo 812C.

[00617] Usando a mesma conexão que a usada com o cliente C1, a unidade de interface 1250 então encaminha o segmento GET a partir do cliente C2 para o servidor, após modificar os números de seqüência e de reconhecimento de acordo com as equações de tradução descritas acima, conforme mostrado pela linha de fluxo 4820. O servidor S responde com os dados requisitados especificando um número de seqüência de 7000, um número de reconhecimento de 2050 e um parâmetro de comprimento de conteúdo de 500, conforme mostrado pelo fluxo 822.

[00618] A unidade de interface 1250 recebe o segmento RESP, traduz os números de seqüência e de reconhecimento e encaminha o segmento RESP para o cliente C2, conforme mostrado pela linha de fluxo 4824A. Aqui, no fluxo 4824B, a unidade de interface 1250 obtém uma confirmação a partir do cliente C2 que o cliente C2 de fato recebeu todos os dados requisitados. A unidade de interface 1250 modifica os números de reconhecimento e de seqüência e encaminha o segmento RESP ACK para o servidor S, conforme mostrado pelo fluxo 4824C.

[00619] A conexão entre o cliente C2 e a unidade de interface 1250 então é fechada ou desconectada uma vez que a unidade de interface

1250 receba um comando FIN ou RST do cliente C2, conforme mostrado pelos fluxos 4826A a 4826D. Da mesma forma, a conexão entre o cliente C1 e a unidade de interface 1250 então é fechada ou desconectada uma vez que receba um comando FIN ou RST do cliente C1, conforme mostrado pelos fluxos 4814A a 4814D. É importante notar, contudo, que a unidade de interface 1250 mantém a conexão com o servidor S. Também é importante notar que a seqüência de eventos, conforme eles foram descritos com referência à figura 36 é para fins de ilustração apenas.

[00620] A figura 36 é um fluxograma que descreve a operação do uso do parâmetro de comprimento de conteúdo para aumento da eficiência do agrupamento de conexões entre clientes e servidores, de acordo com uma modalidade. A unidade de interface 1250 mantém as conexões com uma pluralidade de servidores, e roteia as requisições de cliente para estes servidores, com base no nome de caminho especificado na requisição de cliente. Em primeiro lugar, a unidade de interface 1250 abre as conexões com os servidores, conforme mostrado na etapa 4902. Em seguida, em resposta a uma requisição de cliente C1, a unidade de interface 1250 abre uma conexão com o cliente C1 e recebe uma requisição a partir do cliente C1 para a recuperação de dados usando um nome de caminho, conforme mostrado na etapa 4904.

[00621] A unidade de interface 1250 então seleciona o servidor hospedando o conteúdo especificado pelo nome de caminho, conforme mostrado na etapa 4906. Em modalidades alternativas, a unidade de interface 1250 consulta outras políticas predefinidas para a seleção do servidor apropriado, tal como a carga dos servidores e o estado dos servidores. A unidade de interface 1250 gerencia e mantém um banco de dados de servidores e bancos de servidor que ela abriga. Dentre outras coisas, uma informação neste banco de

dados inclui as políticas atualmente ativas e regras que permitem que a unidade de interface 1250 direcione pacotes chegando para o servidor correto. Dependendo das condições de rede e dos serviços desejados, estas políticas e regras podem mudar muito rapidamente.

[00622] A unidade de interface 1250 então traduz a requisição e passa a requisição traduzida para o servidor selecionado, conforme mostrado na etapa 4908. A unidade de interface 1250 recebe a resposta do servidor S, conforme mostrado na etapa 4910. A unidade de interface 1250 então traduz a resposta e passa a resposta traduzida para o cliente C1, conforme mostrado na etapa 4912.

[00623] Assuma para fins de ilustração que, neste ponto, a unidade de interface 1250 receba uma requisição a partir do cliente C2 para a recuperação de dados. A unidade de interface 1250, em resposta à requisição do cliente C2, abre uma conexão com o cliente C2 e recebe uma requisição a partir do cliente C2 para recuperação de dados usando um nome de caminho, conforme mostrado na etapa 4914. A unidade de interface 1250 então seleciona o servidor hospedando o conteúdo especificado pelo nome de caminho, conforme mostrado na etapa 4916.

[00624] Na etapa 4918, a unidade de interface 1250 determina se o cliente C2 selecionou o mesmo servidor que o cliente C1. Se o resultado da etapa 4918 for negativo, então, a unidade de interface 1250 procede de uma forma necessária para satisfazer à requisição do cliente C2 (a qual não é importante para esta modalidade). Neste ponto, o fluxograma na figura 36 termina. Alternativamente, se o resultado da etapa 4918 for positivo, então, a unidade de interface 1250 determina se há quaisquer conexões abertas para o servidor selecionado, conforme mostrado na etapa 920.

[00625] Se o resultado da etapa 4920 for positivo, então a unidade de interface 1250 procede da forma necessária para satisfazer à

requisição do cliente C2 (a qual não é importante para esta modalidade). Neste ponto, o fluxograma na figura 9 termina. Alternativamente, se o resultado para a etapa 4920 for negativo, então a unidade de interface 1250 utilizará o parâmetro de comprimento de conteúdo para confirmar que o cliente C1 recebeu todos os dados que o cliente C1 requisitou, conforme mostrado na etapa 4922. É importante notar que a unidade de interface 1250 não espera que o cliente C1 envie um comando FIN ou RST de modo a determinar que o cliente C1 terminou a conexão ou está no "tempo para pensar". Isto permite um agrupamento de conexão mais eficiente, devido ao fato de a unidade de interface 1250 poder utilizar cada conexão mais rapidamente do que se a unidade de interface 1250 esperasse pelo cliente fechar a conexão, antes de reusar a conexão para um outro cliente.

[00626] Na etapa 4924, a unidade de interface 1250 então traduz a requisição e passa a requisição traduzida para o servidor selecionado usando a mesma conexão que o cliente C1 usou, conforme mostrado na etapa 4924. A unidade de interface 1250 recebe a resposta do servidor S, conforme mostrado na etapa 4926. A unidade de interface 1250 então traduz a resposta e passa a resposta traduzida para o cliente C2, conforme mostrado na etapa 4928. A unidade de interface 1250 utiliza o parâmetro de comprimento de conteúdo para confirmar que o cliente C2 recebeu todos os dados que o cliente C2 requisitou, conforme mostrado na etapa 4930.

[00627] Em seguida a unidade de interface 1250 fecha ou desfaz a conexão com o cliente C2 na etapa 4932. Finalmente, a unidade de interface 1250 fecha ou desfaz a conexão com o cliente C1 na etapa 4934, e o fluxograma na figura 36 termina. Conforme declarado acima com referência à figura 35, a seqüência de eventos conforme eles foram descritos com referência à figura 36 é para fins de ilustração

apenas.

[00628] A figura 37 é um fluxograma detalhado que ilustra as traduções de números de reconhecimento e de seqüência realizadas por uma modalidade, enquanto se incorpora a técnica de campos de tamanho de fragmento. O rótulo para cada fluxo na figura 37 é da forma T:S,A(L), onde T representa um tipo de segmento de TCP, S é o número de seqüência, A é o número de reconhecimento e L é um campo de tamanho de fragmento. Os valores totais dos campos de tamanho de fragmento descrevem o número de bytes de dados no segmento de TCP.

[00629] Por simplicidade, assumiu-se que as conexões com o cliente C1 e com o cliente C2 já foram estabelecidas. O cliente C1 então envia um comando GET especificando um comprimento de 49 bytes, conforme mostrado pelo fluxo 4002. A unidade de interface 1250 determina que não existe nenhuma conexão aberta livre com o servidor S e, portanto, abre uma conexão com o servidor S (não mostrado na figura 37). A unidade de interface 1250 então encaminha o segmento GET a partir do cliente C1 para o servidor S, após modificar os números de seqüência e reconhecimento de acordo com as equações de tradução descritas acima, conforme mostrado pela linha de fluxo 4004.

[00630] Para fins de ilustração, assuma que os dados no segmento de resposta tenham um comprimento de dados de conteúdo total de 999. Ainda, assuma que os dados serão transmitidos em dois fragmentos de dados de 300 e um fragmento de dados de 399. Nota-se que isto é para fins de ilustração apenas e não é pretendido para limitação. Portanto, o servidor S primeiramente responde com um fragmento dos dados requisitados (ou uma mensagem) especificando um número de seqüência de 6001, um número de reconhecimento de 2000 e um campo de tamanho de fragmento de 300, conforme

mostrado pelo fluxo 4008A. A unidade de interface 1250 recebe o segmento RESP, traduz os números de seqüência e de reconhecimento, e encaminha o segmento RESP para o cliente C1, conforme mostrado pela linha de fluxo 4006A. O cliente C1 reconhece o recebimento dos dados para a unidade de interface 1250, conforme mostrado pela linha de fluxo 4006B. A unidade de interface 1250 em resposta passa este reconhecimento para o servidor S, conforme mostrado pela linha de fluxo 4008B.

[00631] O servidor S em seguida responde com o segundo fragmento dos dados requisitados especificando um número de seqüência de 6301, um número de reconhecimento de 2001 e um campo de tamanho de fragmento de 300, conforme mostrado pelo fluxo 4012A. A unidade de interface 1250 recebe o segmento RESP, traduz os números de seqüência e de reconhecimento, e encaminha o segmento RESP para o cliente C1, conforme mostrado pela linha de fluxo 4010A. O cliente C1 reconhece o recebimento dos dados para a unidade de interface 1250, conforme mostrado pela linha de fluxo 4010B. A unidade de interface 1250 em resposta passa este reconhecimento para o servidor S, conforme mostrado pela linha de fluxo 4012B.

[00632] O Servidor S em seguida responde com o terceiro fragmento dos dados requisitados especificando um número de seqüência de 6601, um número de reconhecimento de 2002 e um campo de tamanho de fragmento de 399, conforme mostrado pelo fluxo 4016A. A unidade de interface 1250 recebe o segmento RESP, traduz os números de seqüência e de reconhecimento, e encaminha o segmento RESP para o cliente C1, conforme mostrado pela linha de fluxo 4014A. O cliente C1 reconhece o recebimento dos dados para a unidade de interface 1250, conforme mostrado pela linha de fluxo 4014B. A unidade de interface 1250 em resposta passa este

reconhecimento para o servidor S, conforme mostrado pela linha de fluxo 4016B.

[00633] Finalmente, o servidor S responde com o fragmento final dos dados zero (indicados por um campo de tamanho de fragmento que equivale a zero) especificando um número de seqüência de 7000, um número de reconhecimento de 2003 e um campo de tamanho de fragmento de 9, conforme mostrado pelo fluxo 4020. A unidade de interface 1250 recebe o segmento RESP, traduz os números de seqüência e de reconhecimento e encaminha o segmento RESP para o cliente C1, conforme mostrado pela linha de fluxo 4018. Isto indica para a unidade de interface 1250 e o cliente C1 que todos os dados requisitados foram transmitidos.

[00634] Neste ponto, o cliente C2 então envia um segmento GET especificando um comprimento de 50 bytes, conforme mostrado pelo fluxo 4022. Assuma neste ponto que a unidade de interface 1250 não tem conexões disponíveis com o servidor S. A meta é reusar a mesma conexão para o servidor S que fora previamente usada para o cliente C1, se o cliente C1 tiver terminado a conexão ou estiver no "tempo para pensar". Ao invés de esperar que o cliente C1 inicie um comando FIN (terminar) ou um comando RST (reinicializar) para liberar a conexão, a unidade de interface usa o campo de tamanho de fragmento que equivalia a zero para confirmar que todos os dados requisitados foram recebidos pelo cliente C1. Isto indica para a unidade de interface 1250 que, embora o cliente C1 possa estar pausando por alguma razão antes de enviar um comando FIN ou RST, o cliente C1 terminou a conexão. A unidade de interface 1250 modifica os números de reconhecimento e de seqüência e encaminha o segmento GET para o servidor S, conforme mostrado pelo fluxo 4024.

[00635] Para fins de ilustração, assumo que os dados no segmento de resposta têm um comprimento de dados de conteúdo total de 500.

Ainda, assumo que os dados serão transmitidos em um fragmento de dados de 300 e um fragmento de dados de 200. Note que isto é para fins de ilustração apenas e não é pretendido para limitação. Portanto, o servidor S primeiramente responde com um fragmento dos dados requisitados especificando um número de seqüência de 7000, um número de reconhecimento de 2050 e um campo de tamanho de fragmento de 300, conforme mostrado pelo fluxo 1028A. A unidade de interface 1250 recebe o segmento RESP, traduz os números de seqüência e de reconhecimento e encaminha o segmento RESP para o cliente C2, conforme mostrado pela linha de fluxo 1026A. O cliente C2 reconhece o recebimento dos dados para a unidade de interface 1250, conforme mostrado pela linha de fluxo 4026B. A unidade de interface 1250 em resposta passa este reconhecimento para o servidor S, conforme mostrado pela linha de fluxo 4028B.

[00636] O servidor S em seguida responde com o segundo fragmento dos dados requisitados especificando um número de seqüência de 7300, um número de reconhecimento de 2051 e um campo de tamanho de fragmento de 200, conforme mostrado pelo fluxo 4032A. A unidade de interface 1250 recebe o segmento RESP, traduz os números de seqüência e de reconhecimento, e encaminha o segmento RESP para o cliente C2, conforme mostrado pela linha de fluxo 4030A. O cliente C2 reconhece o recebimento dos dados para a unidade de interface 1250, conforme mostrado pela linha de fluxo 4030B. A unidade de interface 1250 em resposta passa esta informação para o servidor S, conforme mostrado pela linha de fluxo 4032B.

[00637] Finalmente, o servidor S responde com o fragmento final dos dados zero (indicado por um campo de tamanho de fragmento que equivale a zero) especificando um número de seqüência de 7500, um número de reconhecimento de 2052 e um campo de tamanho de

fragmento de 0, conforme mostrado pelo fluxo 4036. A unidade de interface 1250 recebe o segmento RESP, traduz os números de seqüência e de reconhecimento e encaminha o segmento RESP para o cliente C2, conforme mostrado pela linha de fluxo 4034. Isto indica para a unidade de interface 1250 e o cliente C2 que todos os dados requisitados foram transmitidos.

[00638] A conexão entre o cliente C2 e a unidade de interface 1250 então é fechada ou desfeita, uma vez que a unidade de interface 1250 receba um comando FIN ou RST a partir do cliente C2, conforme mostrado pelo fluxo 4038. Da mesma forma, a conexão entre o cliente C1 e a unidade de interface 1250 então é fechada ou desfeita, uma vez que receba um comando FIN ou RST a partir do cliente C1, conforme mostrado pelo fluxo 4040. É importante notar, contudo, que a unidade de interface 1250 mantém a conexão com o servidor S. Também é importante notar que a seqüência de eventos, conforme eles foram descritos com referência à figura 37 é para fins de ilustração apenas e não limite.

[00639] A figura 38 é um fluxograma que descreve a operação do uso dos campos de tamanho de fragmento para aumento da eficiência do agrupamento de conexões entre clientes e servidores, de acordo com uma modalidade. A unidade de interface 1250 mantém as conexões com uma pluralidade de servidores, e roteia as requisições de cliente para estes servidores com base no nome de caminho especificado na requisição de cliente. Em primeiro lugar, a unidade de interface 1250 abre conexões com os servidores, conforme mostrado na etapa 4102. Em seguida, em resposta a uma requisição de cliente C1, a unidade de interface 1250 abre uma conexão com o cliente C1 e recebe uma requisição do cliente C1 para a recuperação dos dados usando um nome de caminho, conforme mostrado na etapa 4104.

[00640] A unidade de interface 1250 então seleciona o servidor

hospedando o conteúdo especificado pelo nome de caminho, conforme mostrado na etapa 4106. A unidade de interface 1250 então traduz a requisição e passa a requisição traduzida para o servidor selecionado, conforme mostrado na etapa 4108. A unidade de interface 1250 recebe a resposta do servidor S, conforme mostrado na etapa 4110. A unidade de interface 1250 então traduz a resposta e passa a resposta traduzida para o cliente C1, até o campo de tamanho de fragmento ser igual a zero, conforme mostrado na etapa 4112.

[00641] Assuma, para fins de ilustração, que neste ponto a unidade de interface 1250 receba uma requisição a partir do cliente C2 para a abertura de uma conexão. A unidade de interface 1250, em resposta a uma requisição de cliente C2, abre uma conexão para o cliente C2, e recebe uma requisição a partir do cliente C2 para recuperar dados usando um nome de caminho, conforme mostrado na etapa 4114. A unidade de interface 1250 então seleciona o servidor hospedando o conteúdo especificado pelo nome de caminho, conforme mostrado na etapa 4116.

[00642] Na etapa 4118, a unidade de interface 1250 determina se o cliente C2 selecionou o mesmo servidor que o cliente C1. Se o resultado da etapa 4118 for negativo, então a unidade de interface 1250 procede de uma forma necessária para satisfação da requisição de cliente C2. Neste ponto, o fluxograma na figura 38 termina. Alternativamente, se o resultado da etapa 4118 for positivo, então, a unidade de interface 1250 determinará se há quaisquer conexões abertas para o servidor selecionado, conforme mostrado na etapa 4120.

[00643] Se o resultado da etapa 1120 for positivo, então, a unidade de interface 1250 procede de uma forma necessária para satisfação da requisição de cliente C2. Neste ponto, o fluxograma na figura 38 termina. Alternativamente, se o resultado da etapa 4120 for negativo,

então a unidade de interface 1250 utilizará o fato de que o campo de tamanho de fragmento equivalia a zero na etapa 4112 para confirmar que o cliente C1 recebeu todos os dados de mensagem que o cliente C1 requisitou. É importante notar que a unidade de interface 1250 não espera que o cliente C1 envie um comando FIN ou RST de modo a determinar que o cliente C1 terminou a conexão ou está no "tempo para pensar".

[00644] Na etapa 4122, a unidade de interface 1250 então traduz a requisição e passa a requisição traduzida para o servidor selecionado usando a mesma conexão que o cliente C1 usou. A unidade de interface 1250 recebe a resposta do servidor S, conforme mostrado na etapa 4124. A unidade de interface 1250 então traduz a resposta e passa a resposta traduzida para o cliente C2, até o campo de tamanho de fragmento equivaler a zero, conforme mostrado na etapa 4126. A unidade de interface 1250 utiliza o campo de tamanho de fragmento para confirmar que o cliente C2 recebeu todos os dados de mensagem que o cliente C2 requisitou.

[00645] Em seguida, a unidade de interface 1250 fecha ou desfaz a conexão com o cliente C2 na etapa 4128. Finalmente, a unidade de interface 1250 fecha ou desfaz a conexão com o cliente C1 na etapa 4130, e o x na figura 38 termina. Conforme declarado acima com referência à figura 37, a seqüência de eventos conforme eles foram descritos com referência à figura 38 é para fins de ilustração apenas e não de limite.

[00646] As modalidades prévias são descritas especificamente quando implementadas em uma unidade de interface, tal como a unidade de interface 1250, que é conectada a servidores em um banco para fins de descarregamento do tempo de processamento de conexão dos servidores. Contudo, elas também podem ser aplicadas em outros tipos de dispositivos que estejam no caminho de conexão

de rede entre o cliente e os servidores. Conforme um tráfego de rede flui através desses dispositivos, todos eles têm a oportunidade de descarregar o processamento de conexão. Alguns exemplos destes dispositivos são:

- Equilibradores de carga os quais distribuem as conexões de rede de cliente entre um conjunto de servidores em um banco de servidores (distribuído local ou geograficamente).

- Gerenciadores de largura de banda os quais monitoram o tráfego de rede e medem o fluxo de pacote.

- Firewalls monitoram pacotes e permitem que apenas os pacotes autorizados fluam através dali.

- Roteadores e comutadores também ficam no caminho do tráfego de rede. A tendência da indústria pode ser integrar uma funcionalidade adicional (tais como equilíbrio de carga, gerenciamento de largura de banda e funcionalidade de firewall) nestes dispositivos.

[00647] As modalidades também podem ser aplicadas em sistemas de computador os quais estão nos pontos finais de conexões de rede. Neste caso, placas de adição podem ser usadas para descarregamento dos elementos de processamento principais no sistema de computador.

3. Armazenamento em Cache Integrado

[00648] A figura 39 ilustra um fluxograma 5300 de uma seqüência de eventos que podem ocorrer em uma aplicação que provê uma funcionalidade de armazenamento em cache integrado de acordo com uma modalidade. Contudo, a modalidade não está limitada à descrição provida pelo fluxograma 5300. Ao invés disso, será evidente para pessoas versadas na(s) técnica(s) relevante(s) a partir dos ensinamentos providos aqui que outros fluxos funcionais estão no escopo e no espírito da modalidade. Estes outros fluxos funcionais poderiam envolver um processamento diferente, um seqüenciamento

diferente e outras variações sobre a integração de armazenamento em cache.

[00649] O método do fluxograma 5300 pode ser implementado em um ou mais dispositivos que sejam acoplados de forma comunicativa a uma rede de comunicação de dados. Por exemplo, o método do fluxograma 5300 pode ser implementado em uma aplicação, tal como a aplicação 1250 descrita acima com referência à figura 1A, tendo uma arquitetura de software 3200 conforme descrito acima com referência à figura 27. O método do fluxograma 5300 será descrito com referência continuada a esta modalidade de exemplo.

[00650] Conforme mostrado na figura 39, o método do fluxograma 5300 começa na etapa 5302, na qual a aplicação 1250 recebe um pacote encriptado a partir de um dos clientes 10. Em uma modalidade, a aplicação 1250 é configurada para atuar como um ponto final de SSL de próxi para servidores 30, descriptando pacotes encriptados recebidos a partir de clientes 10, e, então, enviado para um processamento adicional, conforme necessário, e, finalmente, para um recurso apropriado com base em uma informação de endereço nos pacotes encriptados. O recurso apropriado pode ser, por exemplo, qualquer um dos servidores 30 ou o cache gerenciado pela aplicação 1250. Na etapa 5304, a aplicação 1250 realiza um processamento de descriptação no pacote.

[00651] Na etapa 5306, a aplicação 1250, a qual é configurada de acordo com uma modalidade para a realização de políticas de AAA para controle de acesso, autentica e/ou autoriza o cliente a partir do qual o pacote encriptado foi recebido.

[00652] Na etapa 5308, a aplicação 1250, a qual é configurada de acordo com uma modalidade para a realização de certos tipos de processamento de pacote, realiza um processamento de pacote nos pacotes descriptados para redução das exigências de

processamento de tempo de processamento de conexão geradas pelos protocolos de rede aplicáveis.

[00653] Na etapa 5310, a aplicação 1250, a qual é configurada de acordo com uma modalidade para compressão e descompressão de conteúdo, descomprime uma requisição associada ao pacote. Em uma modalidade, a requisição compreende uma requisição de objeto da web.

[00654] Na etapa 5312, a aplicação 1250 então é capaz de ativar a funcionalidade de cache, a qual recebe uma requisição clara e/ou autorizada e/ou descomprimida e/ou de pacote processado para um objeto. Devido ao fato de o processamento anterior descrito com referência às etapas 5302, 5304, 5306, 5308 e 5310, a lógica de gerenciamento de cache pode tomar uma decisão quanto a se o objeto foi armazenado em cache ou pode ser armazenado em cache com base em uma requisição clara/autorizada/descomprimida/de pacote processado e, portanto, é capaz de processar um arranjo muito mais amplo de requisições do que os caches tradicionais e realizar o armazenamento em cache mais eficientemente do que segundo abordagens tradicionais. Mais ainda, devido ao fato de a lógica de armazenamento em cache estar funcionando em um espaço de kernel juntamente com outros processos, ela se refere ao objeto relevante como uma estrutura de dados com status igual em relação a essa estrutura de dados como cada um dos outros aplicativos e, portanto, a integração é ganha de uma maneira extremamente eficiente.

[00655] Conforme mostrado na etapa 5314, se o objeto já não estiver na memória cache, a aplicação 1250 envia uma requisição para um ou mais servidores 30. Antes de a requisição ser enviada, contudo, várias etapas de processamento adicionais podem ocorrer.

[00656] Por exemplo, na etapa 5216, a aplicação 1250 opcionalmente realiza um processamento de conexão para garantir um

trânsito eficiente da requisição para o(s) servidor(es) e, na etapa 5318, a aplicação 1250 opcionalmente toma uma decisão de equilíbrio de carga para garantir que a requisição seja enviada para o(s) servidor(es) mais apropriado(s). Também, em uma modalidade, a requisição é encriptada antes de ser enviada para o(s) servidor(es) através de um processo de encriptação de back-end, desse modo se provendo uma segurança de rede de extremidade a extremidade. Na etapa 5320, a requisição é transmitida para o(s) servidor(es).

[00657] Na etapa 5322, a aplicação 1250 recebe uma resposta de volta de um dos servidores 30. Se uma encriptação de back-end for suportada, conforme discutido acima, a aplicação 1250 descriptará a resposta a partir do servidor.

[00658] Na etapa 5324, a aplicação 1250 comprime um objeto associado à resposta a partir do servidor. Em uma modalidade, o objeto compreende um objeto da web.

[00659] Na etapa 5326, a lógica de gerenciamento de cache na aplicação 1250 armazena o objeto no cache em uma forma comprimida. A lógica de gerenciamento de cache é capaz de armazenar objetos comprimidos desta forma, devido às capacidades de processamento – uma vez que o objeto esteja armazenado no cache, futuras requisições de cliente pelo objeto podem ser apresentadas a partir do cache, sem a realização das etapas 5316, 5318, 5320, 5322, 5324 e 5326, conforme descrito acima. Isto é indicado pela linha conectando diretamente a etapa de decisão 5314 à etapa 5328 no fluxograma 5300.

[00660] Na etapa 5328, após o objeto ter sido recebido a partir de um servidor ou recuperado a partir do cache, a aplicação 1250 realiza um processamento de pacote na conexão para servir mais eficientemente a requisição de cliente original. Na etapa 5330, o objeto de resposta então é reencryptado e entregue de volta para o cliente.

[00661] Cada uma das etapas de processamento descritas acima ocorre no nível de kernel/OS da aplicação 1250. Pela implementação do cache no meio de e integrado com outras etapas de processamento no espaço de kernel/OS, uma modalidade é capaz de criar uma funcionalidade adicional e melhorar a performance do cache.

[00662] Essa integração permite uma implementação de cache de acordo com uma modalidade para a realização de funções adicionais que tradicionalmente estão além das capacidades funcionais de um cache. Por exemplo, uma modalidade permite que o cache funcione com objetos encriptados e/ou comprimidos.

[00663] Um outro exemplo de funcionalidade adicional que pode ser obtida por uma modalidade envolve o armazenamento em cache de tráfego de HTTPS encriptado de extremidade a extremidade. Tipicamente, os caches armazenam apenas respostas de HTTP não encriptadas de servidores. Certos caches podem suportar, em alguns casos, uma entrega de HTTPS encriptada de SSL a partir do cache para os clientes, mas, em qualquer caso, os caches tradicionais não são capazes de armazenar em cache respostas que tenham sido encriptadas pelo servidor e, então, são incapazes de suportarem uma encriptação de extremidade a extremidade (isto é, de servidor para cliente). Tipicamente, quando uma resposta é encriptada pelo servidor na forma de HTTPS, o cache não é capaz de desencriptar uma resposta como essa e, portanto, é incapaz de armazenar a resposta em sua memória cache. Por esta razão, os caches tradicionais falham em proverem qualquer benefício face ao tráfego encriptado de extremidade a extremidade em uma modalidade, a aplicação de armazenamento em cache integrado servindo como um ponto de terminação de duas vias para o tráfego de HTTPS encriptado de SSL.

[00664] Por exemplo, em uma modalidade, a aplicação de armazenamento em cache integrado atua como um ponto de

terminação para o tráfego encriptado entre o servidor e a aplicação, e entre a aplicação e os clientes. Desta maneira, a aplicação é capaz de descriptar e armazenar em cache respostas encriptadas de SSL recebidas a partir de servidores e, quando apresentando estas respostas para um cliente, recriptar essa resposta e entregá-la com segurança para o cliente requisitante, desse modo permitindo uma encriptação de extremidade a extremidade e, assim, aumentando a aplicabilidade de armazenamento em cache para uma variedade mais ampla de tráfego da web.

[00665] Em uma modalidade, a aplicação também pode servir como um ponto final em uma rede privada virtual de SSL (VPN de SSL). Em particular, a aplicação pode atuar como um ponto final de SSL de próxi para qualquer recurso em uma rede de comunicação de dados privada, descriptando pacotes encriptados recebidos a partir de um cliente e, então, enviando-os para o recurso de servidor de destino apropriado com base em uma informação de endereço nos pacotes encriptados. Uma sessão de comunicação de dados estabelecida entre o cliente e um gateway pode ser encriptada com o gateway servindo como um ponto final de encriptação, conforme descrito nos parágrafos precedentes do presente pedido. Conforme descrito, o cliente pode usar a Camada de Soquetes Seguros (SSL), IPsec, ou algum outro método de encriptação para o estabelecimento da sessão de comunicação de dados encriptados por meio do que um mecanismo de interceptação no cliente dirige o tráfego para o gateway, enquanto mantém o navegador de cliente pensando que está se comunicando diretamente com os servidores de destino ou as redes de destino. Em uma modalidade como essa, a sessão de comunicação de dados encriptados pode ser terminada no gateway, o que também inclui um cache integrado, conforme descrito aqui. Desta forma, uma funcionalidade de armazenamento em cache pode ser

integrada na funcionalidade de VPN de SSL.

[00666] O gateway também pode realizar quaisquer políticas de AAA aplicáveis à requisição e, conseqüentemente, o gateway servirá objetos armazenados em cache apenas para clientes apropriadamente autenticados, bem como permitindo requisições apenas para usuários autorizados a acessarem um objeto armazenado em cache em particular. Isto é possível porque o cache é integrado de forma tal que as políticas de controle de acesso do gateway sejam cumpridas antes de o cache ver qualquer requisição em particular. Assim, os objetos armazenados em cache obtêm o benefício de controle de acesso sem o cache em si precisar realizar a autenticação e a autorização. Através da integração do cache com essas outras funções, o cache em si se torna mais eficiente e mais efetivo na manipulação da variedade de dados que passam através das redes de hoje em dia. Uma modalidade também é capaz de melhorar a eficiência da performance de rede geral pela introdução dos benefícios de funcionalidade de cache para um arranjo mais amplo de tráfego da web.

[00667] Alguns outros resultados únicos do modo de integração descrito acima de acordo com uma modalidade são conforme se segue. Um resultado é a capacidade de armazenar em cache dados pré-comprimidos e apresentá-los para clientes cientes da compressão. Um outro resultado é a capacidade de armazenar em cache dados de acesso controlado. Ainda um outro resultado é a capacidade de trabalhar com caches externos para a provisão de uma capacidade de escalonamento do cache. Devido ao fato de o cache ser integrado com capacidades de redirecionamento e gerenciamento de tráfego no gateway, os caches externos podem ser empregados para a provisão de uma segunda ligação de armazenamento em cache, desse modo se estendendo a capacidade (e os benefícios) de armazenamento em cache significativamente. Através de uma modalidade, esta

capacidade é criada, sem o módulo de cache em si ter que explicitamente realizar políticas de redirecionamento de cache.

[00668] Em termos de performance, pela integração do cache conforme descrito acima, os processadores do cache são liberados da realização da variedade de tarefas de processamento de conexão que tradicionalmente é requerido que os caches, atuando como nós em uma rede, realizem, e, assim, são capazes de realizarem suas funções de armazenamento em cache em seus níveis de performance mais altos. De fato, ao se permitir o armazenamento em cache de dados comprimidos, o cache é capaz de funcionar mesmo mais eficientemente e permitir que os usuários percebam uma performance ainda mais alta.

[00669] Conforme citado previamente neste pedido, a eficiência surge como resultado da forma pela qual o cache é integrado com outros serviços e tecnologias de rede incluindo uma tecnologia de equilíbrio de carga, encriptação, AAA, compressão e outros tipos de aceleração e processamento de pacote. Como resultado, duplicações de processamento e outras ineficiências introduzidas por modos tradicionais de integração são evitadas. Estas ineficiências, causadas por cópia desnecessária e comutação de contexto, surgem porque cada objeto recebido pelo dispositivo deve ser copiado para uma mensagem e, então, em uma memória de processador antes do processamento pelo aplicativo relevante. A requisição então deve ser copiada de volta para o objeto ou nível de pacote para processamento pelo cache, introduzindo-se cópias de memória adicionais. Em contraste, uma modalidade realiza a integração no nível de OS ou de kernel, desse modo permitindo que o cache opere sobre o objeto como uma estrutura de dados em que o cache tem um status igual aos outros aplicativos e/ou processos em relação a e processando essa estrutura de dados, e onde a necessidade dessas cópias de memória

adicionais é eliminada, já que todos os processos estão trabalhando com a mesma estrutura de dados. O resultado é uma integração mais eficiente.

a. Armazenamento em Cache com Validação Pró-ativa em uma Rede de Comunicação de Dados

[00670] Devido ao fato de os objetos da web poderem mudar ao longo do tempo, cada objeto potencialmente armazenável em cache é dito como tendo uma vida útil ou "frescor". O conceito de frescor se refere ao fato de que o servidor de aplicativo que originalmente gerou o conteúdo também determina o período de tempo em que esse objeto pode ser apresentado por um cache, que pode armazenar esse objeto. Os caches devem ser capazes de determinar se a cópia de um objeto armazenado em sua memória ainda é "fresca" ou não, ou se o cache precisa recuperar uma nova cópia do objeto a partir do servidor de origem. Uma modalidade implementa uma nova abordagem para assegurar o frescor do objeto. Muitas implementações de cache convencionais tentam manter o conteúdo armazenado em cache fresco ao buscarem o conteúdo a partir da origem em uma programação predeterminada. A busca de conteúdo a partir da origem ocorre em tempos estabelecidos pelo administrador de cache tipicamente com base em uma ou mais das abordagens a seguir: (i) em intervalos especificados regulares ou (ii) quando o conteúdo está para expirar.

[00671] Há dois problemas tipicamente associados às abordagens comumente empregadas acima. Em primeiro lugar, cargas de processamento desnecessárias são impostas no servidor de origem, porque é requerido que aquele servidor proveja conteúdo para o cache requisitando a atualização (independentemente de essa atualização ocorrer em intervalos especificados ou conforme o conteúdo estiver para expirar), sem consideração quanto a se esse conteúdo finalmente

será apresentado aos clientes. Em segundo lugar, o cache incorre em uma carga de processador adicional com base no tempo de processamento extra gerado porque o cache precisa manter um acompanhamento dos elementos que devem ser atualizados e do tempo em que eles têm que ser atualizados.

[00672] Um cache de acordo com uma modalidade resolve os problemas acima usando uma nova abordagem pré-busca. A pré-busca do conteúdo não é realizada de acordo com uma programação predefinida ou imediatamente antes da expiração do conteúdo. Ao invés disso, uma modalidade realiza uma pré-busca apenas quando ambas as condições a seguir tiverem sido cumpridas: (1) um cliente fez uma requisição pelo conteúdo especificado e (2) aquele conteúdo 'está para expirar'.

[00673] Esta abordagem se dirige a ambos os problemas descritos acima. Uma revalidação pró-ativa tem maior probabilidade de gerar uma requisição para atualização de conteúdo a partir do servidor de origem quando esse conteúdo estiver sendo ativamente acessado. Isto minimiza a quantidade de carga 'desnecessária' no servidor de origem. Conforme discutido acima, quando o cache requisita uma atualização de objetos que não são finalmente apresentados para clientes (e apenas raramente são apresentados, dependendo da sensibilidade do cache), o cache está utilizando de forma ineficiente seus recursos próprios, bem como os recursos do servidor de origem. Uma modalidade evita o uso ineficiente do cache e dos recursos de servidor ao requisitar apenas aquele conteúdo que estiver sendo ativamente acessado. A abordagem também reduz, pela mesma razão, a largura de banda usada para pré-busca e, portanto, torna mais eficiente o uso dos recursos de rede do que as abordagens tradicionais.

[00674] Mais ainda, uma modalidade usa a informação de expiração incluída no objeto armazenado em cache em si para

determinar se é para requisitar uma atualização do objeto a partir do servidor de origem. Essa informação de expiração tipicamente está incluída nos cabeçalhos do objeto relevante. Esta modalidade assim evita as ineficiências de se encarar qualquer informação adicional para busca diferentemente de muitas abordagens tradicionais as quais requerem que o cache mantenha uma tabela acompanhando a programação para atualização. Usar uma técnica de pré-busca 'baseada em demanda' também melhora os benefícios que são inerentes a uma pré-busca. Esta técnica reduz o número de perdas de cache para objetos acessados freqüentemente, uma vez que esses objetos têm muita probabilidade de sofrerem uma revalidação pró-ativa, imediatamente antes de eles expirarem. Esta técnica também pode impedir o surgimento de tráfego para um servidor de origem que pode ocorrer quando uma resposta grande que está em grande demanda expirar. Na abordagem tradicional, todas as requisições desse conteúdo perdem o cache e são enviadas para o servidor de origem, porque o conteúdo de cache expirou. Em contraste, em uma modalidade, o conteúdo da memória cache geralmente será atualizado imediatamente antes da expiração e, portanto, a situação em que perdas de cache ocorrem enquanto o cache está se atualizando é muito menos provável de surgir.

[00675] Em uma modalidade, a agressividade de pré-busca pode ser controlada através do ajuste da extensão da duração antes da expiração em que o conteúdo é determinado como estando para expirar e também o número de requisições de cliente requeridas para disparo de uma atualização pelo cache do objeto relevante.

b. Processamento de Otimização de Respostas Não Armazenáveis em Cache Grandes Usando "Células Negativas"

[00676] De acordo com uma modalidade, o cache reconhece e não armazena objetos que estejam acima de um tamanho especificado, de

modo a melhorar a relação de acerto de objeto. Os caches tipicamente têm um espaço de memória limitado devotado ao armazenamento de objetos armazenados em cache e, portanto, certas respostas que excedam ao espaço de memória alocado são finalmente rejeitadas como não armazenáveis em cache e não armazenadas pelo cache. Com caches tradicionais, o cache tenta o armazenamento da resposta grande em sua memória cache e apenas aborta o armazenamento da resposta uma vez que o cache reconheça que o tamanho de resposta excede a um tamanho máximo predefinido. Os caches tradicionais repetidamente tentarão armazenar em cache a resposta grande a cada vez em que uma requisição por essa resposta for recebida pelo cache a partir do servidor. Em cada caso, o cache precisará determinar que o objeto não é armazenável em cache, já que excede ao espaço de memória. Assim, esta é manifestamente uma abordagem ineficiente.

[00677] De acordo com uma modalidade, o cache emprega uma otimização para se evitar gastar esforço no armazenamento dessas respostas. Sempre que o cache detectar uma resposta que se torna não armazenável em cache devido ao tamanho da resposta, ele armazena uma notação referente à requisição correspondente em uma estrutura de dados denominada uma "célula negativa". A notação indica que a requisição não é armazenável em cache. No futuro, quando um cliente requisita o mesmo objeto, a requisição é combinada com a notação considerada como a primeira requisição armazenada na estrutura de dados. Facilitada a combinação, o cache não tentará armazenar em cache a resposta e, ao invés disso, a requisição será completamente desviada pelo cache.

[00678] Não há uma configuração de usuário requerida para a especificação da duração pela qual uma célula negativa deve permanecer no cache. De fato, os usuários não estão nem mesmo

cientes que este mecanismo em particular está sendo empregado. Em uma modalidade, o cache usa a informação de expiração regular que ele teria empregado para o armazenamento em cache da resposta grande para armazenar em cache a "informação negativa" sobre aquela resposta.

4. Aceleração de Lado de Cliente

[00679] Em uma modalidade, um programa de aceleração de lado de cliente pode realizar uma ou mais técnicas de aceleração para aceleração, melhoria ou aperfeiçoamento de outra forma de comunicações de um cliente com e/ou acesso a um servidor, tal como acessando um aplicativo provido por um servidor. Com referência, agora, à figura 40A, um cliente 6205 tendo um programa de aceleração 6120 é descrito. Em uma visão geral breve, o cliente 6205 opera sobre o dispositivo de computação 6100 que tem um sistema operacional com um modo de kernel 6202 e um modo de usuário 6202, e uma pilha de rede 6210 com uma ou mais camadas 6210a a 6210b. O cliente 6205 pode compreender todos e quaisquer dos clientes 10 discutidos previamente. Embora apenas um cliente 6205 seja mostrado, qualquer número de clientes 10 pode compreender o cliente 6205. O cliente 6205 pode ter instalados e/ou executar um ou mais aplicativos 6220a a 6220n. Em algumas modalidades, um ou mais aplicativos 6220a a 6220n podem se comunicar através da pilha de rede 6210 com uma rede. Um dos aplicativos 6220N também pode incluir um primeiro programa 6222, por exemplo, um programa o qual pode ser usado em algumas modalidades para a instalação e/ou a execução do programa de aceleração 6120.

[00680] A pilha de rede 6210 do cliente 6205 pode compreender qualquer tipo e forma de software ou hardware, ou quaisquer combinações dos mesmos, para a provisão de conectividade para e comunicações com uma rede. Em uma modalidade, a pilha de rede

6210 compreende uma implementação de software para uma suíte de protocolo de rede. A pilha de rede 6210 pode compreender uma ou mais camadas de rede, tais como quaisquer camadas de rede do modelo de comunicações de Interconexão de Sistemas Abertos (OSI), tais como aqueles versados na técnica reconhecem a apreciam. Como tal, a pilha de rede 6210 pode compreender qualquer tipo e forma de protocolos para qualquer uma das camadas a seguir do modelo de OSI: 1) uma camada de enlace física, 2) uma camada de enlace de dados, 3) uma camada de rede, 4) uma camada de transporte, 5) uma camada de sessão, 6) uma camada de apresentação, e 7) uma camada de aplicação. Em uma modalidade, a pilha de rede 310 pode compreender um protocolo de controle de transporte (TCP) pelo protocolo de camada de rede do protocolo de internet (IP), geralmente referido como TCP/IP. Em algumas modalidades, o protocolo de TCP/IP pode ser realizado pelo protocolo de Ethernet, o qual pode compreender qualquer um dos protocolos da família de rede de área ampla (WAN) ou de rede de área local (LAN) do IEEE, tais como aqueles protocolos cobertos pelo IEEE 802.3. Em algumas modalidades, a pilha de rede 6210 compreende qualquer tipo e forma de protocolo sem fio, tal como IEEE 802.11 e/ou um protocolo de internet móvel.

[00681] Tendo em vista uma rede baseada em TCP/IP, qualquer protocolo baseado em TCP/IP pode ser usado, incluindo uma Interface de Programação de Aplicativo de Envio de Mensagem (MAPI) (e-mail), Protocolo de Transferência de Arquivo (FTP), Protocolo de Transferência de Hipertexto (HTTP), protocolo de Sistema de Arquivo de Internet Comum (CIFS) (transferência de arquivo), protocolo de Arquitetura de Computação Independente (ICA), Protocolo de Área de Trabalho Remota (RDP), Protocolo de Aplicativo Sem Fio (WAP), protocolo de IP Móvel, e protocolo de Voz por IP (VoIP). Em uma outra

modalidade, a pilha de rede 210 compreende qualquer tipo e forma de protocolo de controle de transporte, tal como um protocolo de controle de transporte modificado, por exemplo, um TCP de Transação (T/TCP), um TCP com reconhecimentos de seleção (TCP-SACK), TCP com janelas grandes (TCP-LW), um protocolo de predição de congestionamento, tal como o protocolo TCP-Vegas, e um protocolo de enganação de TCP. Em outras modalidades, qualquer tipo e forma de protocolo de datagrama de usuário (UDP), tal como UDP por IP, pode ser usado pela pilha de rede 6210, tal como para comunicações de voz ou para comunicações de dados em tempo real.

[00682] Mais ainda, a pilha de rede 6210 pode incluir um ou mais drivers de rede suportando uma ou mais camadas, tal como um driver de TCP ou um driver de camada de rede. Os drivers de rede podem ser incluídos como parte do sistema operacional do dispositivo de computação 100 ou como parte de quaisquer placas de interface de rede ou outros componentes de acesso de rede do dispositivo de computação 6100. Em algumas modalidades, qualquer um dos drivers de rede da pilha de rede 6210 pode ser personalizado, modificado ou adaptado para prover uma porção personalizada ou modificada da pilha de rede 6210 em suporte de qualquer uma das técnicas descritas aqui. Em outras modalidades, o programa de aceleração 6120 é projetado e construído para operar com ou trabalhar em conjunto com a pilha de rede 6210, instalado ou provido de outra forma pelo sistema operacional do cliente 205.

[00683] A pilha de rede 6210 compreende qualquer tipo e forma de interfaces para recebimento, obtenção, provisão ou acesso de outra forma a qualquer informação e dados relativos a comunicações de rede do cliente 6205. Em uma modalidade, uma interface com a pilha de rede 6210 compreende uma interface de programação de aplicativo (API). A interface também pode compreender qualquer chamada

funcional, mecanismo de enganchamento ou filtração, mecanismo de evento ou de retorno de chamada, ou qualquer tipo de técnica de criação de interface. A pilha de rede 6210 através da interface pode receber ou prover qualquer tipo e forma de estrutura de dados, tal como um objeto, relacionado à funcionalidade ou operação da pilha de rede 6210. Por exemplo, a estrutura de dados pode compreender uma informação e dados relacionados a um pacote de rede ou um ou mais pacotes de rede. Em algumas modalidades, a estrutura de dados compreende uma porção do pacote de rede processado em uma camada de protocolo da pilha de rede 6210, tal como um pacote de rede da camada de transporte. Em algumas modalidades, a estrutura de dados 6225 compreende uma estrutura de dados de nível de kernel, enquanto em outras modalidades a estrutura de dados 6225, enquanto em outras modalidades a estrutura de dados 6225 compreende uma estrutura de dados de modo de usuário. Uma estrutura de dados de nível de kernel pode compreender uma estrutura de dados obtida ou relacionada a uma porção da pilha de rede 6210 operando em modo de kernel 6202, ou um driver de rede ou algum software rodando em modo de kernel 6202, ou qualquer estrutura de dados obtida ou recebida por um serviço, processo, tarefa, linha ou outras instruções executáveis rodando ou operando em modo de kernel do sistema operacional.

[00684] Adicionalmente, algumas porções da pilha de rede 6210 podem se executar ou operar no modo de kernel 6202, por exemplo, o enlace de dados ou câmara camada de rede, enquanto outras porções se executam ou operam no modo de usuário 6203, tal como uma camada de aplicativo da pilha de rede 6210. Por exemplo, uma primeira porção 6210a da pilha de rede pode prover um acesso de modo de usuário à pilha de rede 6210 a um aplicativo 6220a a 6220n, enquanto uma segunda porção 6210a da pilha de rede 6210 provê

acesso a uma rede. Em algumas modalidades, uma primeira porção 6210a da pilha de rede pode compreender uma ou mais camadas superiores da pilha de rede 6210, tal como qualquer uma das camadas 5 a 7. Em outras modalidades, uma segunda porção 6210b da pilha de rede 6210 compreende uma ou mais camadas inferiores, tal como qualquer uma das camadas 1 a 4. Cada uma dentre a primeira porção 6210a e a segunda porção 6210b da pilha de rede 6210 pode compreender qualquer porção da pilha de rede 6210, em qualquer uma de uma ou mais camadas de rede, em modo de usuário 6203, em modo de kernel 6202, ou combinações dos mesmos, ou em qualquer porção de uma camada de rede ou ponto de interface para uma camada de rede ou qualquer porção de ou ponto de interface para o modo de usuário 6203 e o modo de kernel 6203.

[00685] O programa de aceleração 6120 da presente pode compreender um software, um hardware, ou qualquer combinação de software e de hardware. Em algumas modalidades, o programa de aceleração 6120 compreende qualquer tipo e forma de instruções executáveis construídas e projetadas para executarem ou proverem uma funcionalidade e operações conforme descrito aqui. Em algumas modalidades, o programa de aceleração 6120 compreende qualquer tipo e forma de aplicativo, programa, serviço, processo, tarefa ou linha. Em uma modalidade, o programa de aceleração 6120 compreende um driver, tal como um driver de rede construído e projetado para ter uma interface e trabalhar com a pilha de rede 6210. A lógica, as funções e/ou operações das instruções executáveis do programa de aceleração 6120 podem realizar uma ou mais das técnicas de aceleração a seguir: 1) compressão de protocolo múltiplo 6238, 2) agrupamento de protocolo de controle de transporte 6224, 3) multiplexação de protocolo de controle de transporte 6226, 4) armazenamento em buffer de protocolo de controle de transporte

6228, e 5) armazenamento em cache através de um gerenciador de cache 6232, as quais serão descritas em maiores detalhes abaixo. Adicionalmente, o programa de aceleração 6120 pode realizar uma encriptação 6234 e/ou uma desencriptação de quaisquer comunicações recebidas e/ou transmitidas pelo cliente 6205. Em algumas modalidades, o programa de aceleração 6120 também realiza um tunelamento entre o cliente 6205 e um outro dispositivo de computação 6100, tal como um servidor 30. Em outras modalidades, o programa de aceleração 6120 provê uma conexão de rede privada virtual com um servidor 30.

[00686] Em algumas modalidades, o programa de aceleração 6120 opera em uma ou mais camadas da pilha de rede 6210, tal como na camada de transporte. Em uma modalidade, o programa de aceleração 6120 compreende um driver de filtro, um mecanismo de enganchamento ou qualquer forma e tipo de interface de driver de rede adequada que tenha uma interface com a camada de transporte da pilha de rede, tal como através da interface de driver de transporte (TDI). Em algumas modalidades, o programa de aceleração 6120 tem uma interface com uma primeira camada de protocolo, tal como a camada de transporte ou uma outra camada de protocolo, tal como qualquer camada acima da camada de protocolo de transporte, por exemplo, uma camada de protocolo de aplicativo. Em uma modalidade, o programa de aceleração 6120 pode compreender um driver em conformidade com a Especificação de Interface de Driver de Rede (NDIS), ou um driver de NDIS. Em uma outra modalidade, o programa de aceleração 6120 pode compreender um filtro de mínimo ou um driver de miniporta. Em uma modalidade, o programa de aceleração 6120 ou uma porção do mesmo opera no modo de kernel 6202. Em uma outra modalidade, o programa de aceleração 6120 ou uma porção do mesmo opera no modo de usuário 6203. Em algumas

modalidades, uma porção do programa de aceleração 6120 opera no modo de kernel 6202, enquanto uma outra porção do programa de aceleração 6120 opera no modo de usuário 6203. Em outras modalidades, o programa de aceleração 6120 opera no modo de usuário 6203, mas tem uma interface com um driver de modo de kernel, processo, serviço, tarefa ou porção do sistema operacional, de modo a obter uma estrutura de dados de nível de kernel 6225. Em modalidades adicionais, o programa de aceleração 6120 é um aplicativo ou programa de modo de usuário, tal como um aplicativo 6220a a 6220n.

[00687] O programa de aceleração 6120 pode operar na ou ter uma interface com uma camada de protocolo de uma maneira transparente para qualquer outra camada de protocolo da pilha de rede 6210. Por exemplo, em uma modalidade, o programa de aceleração 6120 opera ou tem uma interface com a camada de transporte da pilha de rede 6210 de forma transparente para qualquer camada de protocolo abaixo da camada de transporte, tal como a camada de rede, e qualquer camada de protocolo acima da camada de transporte, tais como os protocolos de camada de sessão, apresentação ou aplicativo. Isto permite que outras camadas de protocolo da pilha de rede 6210 operem conforme desejado e sem modificação, para uso do programa de aceleração 6120. Como tal, o programa de aceleração 6120 pode ter uma interface com a camada de transporte para aceleração de quaisquer comunicações providas através de qualquer protocolo portado pela camada de transporte, tal como qualquer protocolo de camada de aplicativo por TCP/IP.

[00688] Mais ainda, o programa de aceleração 6120 pode operar na ou ter uma interface com a pilha de rede 6210 de uma maneira transparente para qualquer aplicativo 6220a a 6220n, um usuário do cliente 6205, ou qualquer outro dispositivo de computação, tal como

um servidor, em comunicações com o cliente 6205. O programa de aceleração 6120 pode ser instalado e/ou executado no cliente 6205 de uma maneira tal que o programa de aceleração 6120 possa acelerar quaisquer comunicações de um aplicativo 6220a a 6220n, sem modificação do aplicativo 6220a a 6220n. Em algumas modalidades, o usuário do cliente 6205 ou um dispositivo de computação em comunicações com o cliente 6205 não está ciente da existência, execução ou operação do programa de aceleração 6120. Como tal, em algumas modalidades, o programa de aceleração 6120 é instalado, executado e/ou operado de forma transparente para um aplicativo 6220a a 6220n, um usuário do cliente 6205, um outro dispositivo de computação, tal como um servidor, ou qualquer uma das camadas de protocolo acima e/ou abaixo da camada de protocolo com a interface criada pelo programa de aceleração 6120.

[00689] Em algumas modalidades, o programa de aceleração 6120 realiza uma ou mais das técnicas de aceleração 6224, 6226, 6228, 6232 de uma maneira ou forma integrada. Em uma modalidade, o programa de aceleração 6128 compreende qualquer tipo e forma de mecanismo para a interceptação, o enganchamento, a filtração ou a recepção de comunicações na camada de protocolo de transporte da pilha de rede 6210. Pela interceptação de um pacote de rede do cliente 6205 na camada de transporte e a criação de uma interface com a pilha de rede 6210 na camada de transporte através de uma estrutura de dados, tal como uma estrutura de dados de nível de kernel 6225, o programa de aceleração 6120 pode realizar técnicas de aceleração relacionadas à camada de transporte no pacote de rede, tal como um armazenamento em buffer de protocolo de controle de transporte (TCP), um agrupamento de TCP e uma multiplexação de TCP. Adicionalmente, o programa de aceleração 6120 pode realizar uma compressão 6225 ou qualquer um dos protocolos, ou múltiplos

protocolos, portados como carga útil de pacote de rede do protocolo de camada de transporte.

[00690] Em uma modalidade, o programa de aceleração 6120 usa uma estrutura de dados de nível de kernel 6225 provendo acesso a qualquer porção de um ou mais pacotes de rede, por exemplo, um pacote de rede compreendendo uma requisição a partir de um cliente 6205 ou uma resposta a partir de um servidor. Em uma modalidade, a estrutura de dados de nível de kernel pode ser usada pelo programa de aceleração 6120 para a realização da técnica de aceleração desejada. Em uma modalidade, o programa de aceleração 6120 está rodando no modo de kernel 6202, quando se usa a estrutura de dados de nível de kernel 6225, enquanto em uma outra modalidade o programa de aceleração 6120 está rodando no modo de usuário 6203, quando se usa a estrutura de dados de nível de kernel 6225. Em algumas modalidades, a estrutura de dados de nível de kernel pode ser copiada ou passada para uma segunda estrutura de dados de nível de kernel, ou qualquer estrutura de dados de nível de usuário desejada. Embora o programa de aceleração 6120 geralmente seja descrito na figura 40A como tendo uma primeira porção operando em modo de usuário 6203 e uma segunda porção operando em modo de kernel 6202, em algumas modalidades, qualquer porção do programa de aceleração 6120 pode rodar em modo de usuário 6203 ou em modo de kernel 6202. Em algumas modalidades, o programa de aceleração 6120 pode operar apenas em modo de usuário 6203, enquanto em outras modalidades, o programa de aceleração 6120 pode operar apenas em modo de kernel 6202.

[00691] Mais ainda, pela interceptação na camada de transporte da pilha de rede 6210 ou pela obtenção de acesso ao pacote de rede através de uma estrutura de dados de nível de kernel 6225, o programa de aceleração 6120 pode realizar ou aplicar uma pluralidade

de técnicas de aceleração em um ponto de interface único ou em um ponto de execução único ou tempo de execução de quaisquer instruções executáveis do programa de aceleração 6120. Por exemplo, em uma modalidade, em uma função ou um conjunto de instruções do programa de aceleração 6120, uma pluralidade de técnicas de aceleração pode ser executada, tal como ao se chamar um conjunto de instruções executáveis construídas e projetadas para a realização da técnica de aceleração. Em algumas modalidades, o programa de aceleração 6120 em um ponto de interface, local de execução ou em um conjunto de instruções chama um ou mais interfaces de programação de aplicativo (APIs) para qualquer programa, serviço, processo, tarefa, linha, ou instruções executáveis projetadas e construídas para proverem: 1) compressão de protocolo múltiplo 6238, 2) agrupamento de protocolo de controle de transporte 6224, 3) multiplexação de protocolo de controle de transporte 6226, 4) armazenamento em buffer de protocolo de controle de transporte 6228, e 5) armazenamento em cache através de um gerenciador de cache 6232 e, em algumas modalidades, uma encriptação 6234.

[00692] Pela execução da pluralidade de técnicas de aceleração em um local ou uma localização nas instruções executáveis do programa de aceleração 6120 ou em uma camada de protocolo da pilha de rede 6210, tal como a camada de transporte, a integração destas técnicas de aceleração é realizada de forma mais eficiente e efetiva. Em um aspecto, o número de comutações de contexto entre processos pode ser reduzido, bem como se reduzindo o número de estruturas de dados usadas ou cópias de estruturas de dados em memória necessárias ou usadas de outra forma. Adicionalmente, uma sincronização de e comunicações entre qualquer uma das técnicas de aceleração pode ser realizada mais eficientemente, tal como de uma maneira acoplada firmemente, em um conjunto de instruções

executáveis do programa de aceleração 6120. Como tal, qualquer lógica, regras, funcionalidade ou operações referentes à ordem de técnicas de aceleração, quais técnicas realizar, e dados e informação a serem compartilhados ou passados entre técnicas podem ser realizados mais eficientemente. O programa de aceleração 6120 pode interceptar um pacote de TCP na camada de transporte, obter a carga útil do pacote de TCP através de uma estrutura de dados de nível de kernel 6225, e, então, realizar as técnicas de aceleração desejadas em uma ordem desejada. Por exemplo, o pacote de rede pode ser primeiramente comprimido e, então, armazenado em cache. Em um outro exemplo, os dados armazenados em cache comprimidos podem ser comunicados através de uma conexão de TCP armazenada em buffer, agrupada e/ou multiplexada para um servidor.

[00693] Em algumas modalidades, e ainda com referência à figura 40A, um primeiro programa 6222 pode ser usado para a instalação e a execução do programa de aceleração 6120, de forma automática, silenciosa, transparente ou de outra forma. Em uma modalidade, o primeiro programa 6222 compreende um componente de plug-in, tal como um controle ActiveX ou um controle Java ou um script que é carregado e executado por um aplicativo 6220a a 6220n. Por exemplo, o primeiro programa compreende um controle ActiveX carregado e rodado por um aplicativo navegador da web 6220, tal como no espaço de memória ou contexto do aplicativo 6220. Em uma outra modalidade, o primeiro programa 6222 compreende um conjunto de instruções executáveis carregadas em e rodadas pelo aplicativo 6220a a 6220n, tal como um navegador. Em uma modalidade, o primeiro programa 6222 compreende um programa projetado e construído para a instalação do programa de aceleração 6120. Em algumas modalidades, o primeiro programa 6222 obtém, transfere (via download) ou recebe o programa de aceleração 6120 através da rede

a partir de um outro dispositivo de computação. Em uma outra modalidade, o primeiro programa 6222 é um programa instalador ou um gerenciador de plug and play para instalação de programas, tais como drivers de rede ou o sistema operacional do cliente 6205.

[00694] Em outras modalidades, o primeiro programa 6222 pode compreender todas e quaisquer das funcionalidades descritas aqui na Seção B. Em uma outra modalidade, o primeiro programa 6222 pode compreender um agente de coleta 404. Em uma outra modalidade, o primeiro programa pode compreender um programa para instalação de um agente de coleta. Em uma outra modalidade, o primeiro programa 6222 também pode compreender um ambiente de computação 15. Em uma modalidade, o primeiro programa 6222 também pode compreender um meio para a instalação de um ambiente de computação, tal como um ambiente de execução ou um ambiente de execução virtual. Em uma modalidade, o primeiro programa 6222 pode compreender um cliente de transmissão de aplicativo 442, conforme discutido previamente. Em uma outra modalidade, o primeiro programa 6222 pode compreender um aplicativo a ser executado em um cliente 10.

[00695] Em outras modalidades, o primeiro programa 6222 pode compreender uma porção da funcionalidade, operações e lógica do programa de aceleração 6120 para facilitação ou realização de qualquer outra funcionalidade, operações e lógica do programa de aceleração 6120 descrito aqui, tal como qualquer uma das técnicas de aceleração. Em algumas modalidades, o primeiro programa 6222 é usado para o estabelecimento de uma conexão, tal como uma conexão de camada de transporte, ou uma sessão de comunicação com uma aplicação ou um servidor, tal como uma sessão de comunicação de Camada de Soquete Seguro (SSL). Em uma modalidade, um primeiro programa 6222 é usado para o

estabelecimento ou a facilitação do estabelecimento de uma conexão de rede privada virtual e uma sessão de comunicação.

[00696] O gerenciador de cache 6232 do programa de aceleração 6120 ou o cliente 6205, conforme descrito na figura 40A, pode compreender um software, um hardware ou qualquer combinação de software e hardware para a provisão de acesso a cache, controle e gerenciamento de qualquer tipo e forma de conteúdo, tais como objetos ou objetos gerados dinamicamente apresentados pelos servidores 30. Os dados, objetos ou conteúdo processados e armazenados pelo gerenciador de cache 6232 podem compreender dados em qualquer formato, tal como uma linguagem de marcação, ou comunicados através de qualquer protocolo. Em algumas modalidades, o gerenciador de cache 6232 duplica os dados originais armazenados em algum lugar ou dados previamente computados, gerados ou transmitidos, em que os dados originais podem requerer um tempo de acesso mais longo para serem buscados, computados ou obtidos de outra forma em relação a um elemento de memória cache. Uma vez que os dados estejam armazenados no elemento de dados e memória cache, um uso futuro pode ser feito pelo acesso à cópia armazenada em cache, ao invés de se buscarem de novo ou recomputarem os dados originais, desse modo se reduzindo o tempo de acesso. Em algumas modalidades, o elemento de memória cache pode compreender um objeto de dados na memória do cliente 6205. Em outras modalidades, o elemento de memória cache pode compreender uma memória tendo um tempo de acesso mais rápido do que a memória usada de outra forma pelo cliente 6205. Em uma outra modalidade, o elemento de mecanismo pode compreender qualquer tipo e forma de elemento de armazenamento do cliente 6205, tal como uma porção de um disco rígido. Ainda em uma outra modalidade, o gerenciador de cache 6232 pode usar qualquer porção e combinação

de memória, armazenamento, ou a unidade de processamento para armazenamento em cache de dados, objetos e outro conteúdo.

[00697] Mais ainda, o gerenciador de cache 6232 pode incluir qualquer lógica, funções, regras ou operações para a realização de quaisquer modalidades das técnicas descritas aqui. Por exemplo, o gerenciador de cache 6232 inclui uma lógica ou funcionalidade para invalidação de objetos com base na expiração de um período de tempo de invalidação ou mediante o recebimento de um comando de invalidação de um cliente 6205a a 6205n ou servidor 30. Em algumas modalidades, o gerenciador de cache 6232 pode operar como um programa, serviço, processo ou tarefa em execução no espaço de kernel 6202 e, em outras modalidades, no espaço de usuário 6203. Em uma modalidade, uma primeira porção do gerenciador de cache 6232 se executa no espaço de usuário 6203, enquanto uma segunda porção se executa no espaço de kernel 6202. Em algumas modalidades, o gerenciador de cache 6232 pode compreender qualquer tipo de processador de finalidade geral (GPP), ou qualquer outro tipo de circuito integrado, tal como um Arranjo de Porta Programável de Campo (FGPA), um Dispositivo Lógico Programável (PLD) ou um Circuito Integrado Específico de Aplicação (ASIC).

[00698] O agente de encriptação 6234 do programa de aceleração 6120 ou do cliente 6205 compreende qualquer lógica, regras de negócio, funções ou operações para manipulação do processamento de qualquer protocolo relacionado à segurança, tal como SSL ou TLS, ou qualquer função relacionada a isso. Por exemplo, o agente de encriptação 6234 encripta e desencripta pacotes de rede, ou qualquer porção dos mesmos, comunicados pelo cliente 6205. O agente de encriptação 6234 também pode configurar ou estabelecer conexões de SSL ou TLS em nome do cliente 6205. Como tal, o agente de encriptação 6234 provê um descarregamento e uma aceleração do

processamento de SSL. Em uma modalidade, o agente de encriptação 6234 usa um protocolo de tunelamento para prover uma rede privada virtual entre um cliente 6205 e um outro dispositivo de computação, tal como um servidor.

[00699] Ainda com referência à figura 40A, o agente de compressão de protocolo múltiplo 6238 do programa de aceleração 6120 ou do cliente 6205 compreende qualquer lógica, regras de negócios, função ou operações para compressão de um ou mais protocolos de um pacote de rede, tal como qualquer um dos protocolos usados pela pilha de rede 6210 do cliente 6205. Por exemplo, a compressão de protocolo múltiplo 6238 pode incluir utilitários de compressão e descompressão compreende uma compressão e uma descompressão de GZip, compressão diferencial e UnCompression, ou qualquer outro utilitário proprietário ou publicamente disponível para compressão e descompressão de dados a serem transmitidos por uma rede. Em uma modalidade, o agente de compressão de protocolo múltiplo 6238 comprime bidirecionalmente entre o cliente 6205 e um outro dispositivo de computação, tais como servidores, qualquer protocolo baseado em TCP/IP, incluindo uma Interface de Programação de Aplicativo de Envio de Mensagem (MAPI) (e-mail), um Protocolo de Transferência de Arquivo (FTP), um Protocolo de Transferência de Hipertexto (HTTP), protocolo de Sistema de Arquivo de Internet Comum (CIFS) (transferência de arquivo), um protocolo de Arquitetura de Computação Independente (ICA), um Protocolo de Área de Trabalho Remota (RDP), um Protocolo de Aplicativo Sem fio (WAP), um protocolo de IP móvel, e um protocolo de Voz por IP (VoIP). Em outras modalidades, o agente de compressão de protocolo múltiplo 6238 provê uma compressão de protocolos baseados em Linguagem de Marcação de Hipertexto (HTML) e, em algumas modalidades, provê a compressão de quaisquer linguagens de marcação, tal como a

Linguagem de Marcação Extensível (XML). Como tal, o agente de compressão de protocolo múltiplo 6238 acelera a performance para usuários acessando aplicativos através de clientes de área de trabalho, por exemplo, os clientes de Microsoft Outlook e leves não da web, tal como qualquer cliente aberto por aplicativos de empresa populares, tais como Oracle, SAP e Siebel, e mesmo clientes móveis, tais como Pocket PC.

[00700] O programa de aceleração 6120 também realiza técnicas de aceleração de camada de protocolo de transporte de armazenamento em buffer, agrupamento e multiplexação, conforme será descrito em maiores detalhes abaixo. Como tal, o programa de aceleração 6120 compreende qualquer tipo e forma de instruções executáveis tendo lógica, regras, funções e operações para a realização de qualquer uma destas técnicas, conforme descrito aqui. O programa de aceleração 6120 intercepta, controla e gerencia na camada de transporte da pilha de rede 6210 quaisquer chamadas de interface de programação de aplicativo (API) de camada de transporte feitas por um aplicativo 6220a a 6220n através da pilha de rede 6210. O programa de aceleração 6120 responde a quaisquer requisições do cliente 6205 de uma maneira transparente, de modo que o cliente 6205 receba uma resposta conforme esperado a partir da camada de protocolo de transporte da pilha de rede 6210. Por exemplo, em uma modalidade, o programa de aceleração 6120 intercepta na pilha de rede 6210 do cliente 6205 uma requisição para o estabelecimento de uma conexão de camada de transporte com um outro dispositivo de computação, tal como um servidor, e pode usar um grupo de uma ou mais conexões de camada de transporte estabelecidas pelo programa de aceleração 6120 para responder à requisição. Em uma outra modalidade, o programa de aceleração 6120 multiplexa uma requisição a partir de um primeiro aplicativo 6220a através de uma

conexão de camada de transporte estabelecida usada por um segundo aplicativo 6220b.

[00701] Em algumas modalidades, o programa de aceleração 6120 compreende um mecanismo para armazenamento em buffer ou manutenção de comunicações do cliente 6205 no cliente 6205, antes da transmissão em uma rede. Por exemplo, a taxa de consumo pelo cliente 6205 das comunicações recebidas a partir de uma rede, tal como a partir de um servidor, pode ser menor do que a taxa de produção de comunicações transmitida pelo cliente 6205 na rede. Como tal, o cliente 6205 pode estar enviando mais requisições para um servidor 30 a uma taxa maior do que pela qual o cliente 6205 pode consumir e processar respostas a partir dessas requisições. O programa de aceleração 6120 pode interceptar uma comunicação, e determinar se uma taxa de consumo e/ou uma taxa de produção do cliente 6205 está abaixo de um limite predeterminado, tal com um limite configurado por um usuário, pelo cliente 6205 ou por um outro dispositivo de computação. Se a taxa determinada estiver abaixo do limite desejado, o programa de aceleração 6120 armazenará a comunicação interceptada em um elemento de memória do cliente, até a performance do cliente 6205 aumentar a taxa de consumo e/ou a produção para uma taxa igual a ou mais alta do que o limite predeterminado ou desejado. Nesse ponto, um mecanismo de lado de cliente é provido para estrangulamento das comunicações do cliente 6205, com base na performance de consumo e/ou na produção de comunicações pelo cliente 6205.

[00702] O aplicativo 6220a a 6220n descrito na figura 40A pode ser qualquer tipo e/ou forma de aplicativo, tal como qualquer tipo e/ou forma de navegador da web, cliente baseado na web, aplicativo de cliente – servidor, um cliente de computação de cliente leve, um controle ActiveX, ou um miniaplicativo Java, ou qualquer outro tipo

e/ou forma de instruções executáveis capazes de execução no cliente 6205 ou de comunicação através de uma rede 6204. O aplicativo 6220a a 6220n pode usar qualquer tipo de protocolo e ele pode ser, por exemplo, um cliente de HTTP, um cliente de FTP, um cilindro de Oscar ou um cliente de Telnet. Em algumas modalidades, o aplicativo 6220a a 6220n usa uma exibição remota ou um protocolo de nível de apresentação. Em uma modalidade, o aplicativo 6220a a 6220n é um cilindro ICA, desenvolvido pela Citrix Systems, Inc. de Fort Lauderdale, Flórida. Em outras modalidades, o aplicativo 6220a a 6220n inclui um cliente de Área de Trabalho Remota (RDP), desenvolvido pela Microsoft Corporation de Redmond, Washington. Em outras modalidades, o aplicativo 6220a a 6220n compreende qualquer tipo de software relacionado a comunicações de VoIP, tal como um softfone de IP. Em outras modalidades, o aplicativo 6220a a 6220n compreende qualquer aplicativo relacionado a comunicações de dados em tempo real, tais como aplicativos para transmissão de vídeo e/ou áudio.

[00703] A figura 40B ilustra uma arquitetura de exemplo de uma aplicação 1250 similar à arquitetura de aplicação descrita na figura 27. Em uma visão geral breve, a aplicação 1250 compreende uma camada de hardware 6206 e uma camada de software dividida em um espaço de usuário 6203 e um espaço de kernel 6202. A camada de hardware 6206 também provê as estruturas e os elementos os quais permitem que programas e serviços no espaço de kernel 6202 e no espaço de usuário 6203 para comunicação de dados interna e externamente com respeito à aplicação 1250. A camada de software compreende programas, serviços, processos, tarefas, linhas e outras instruções executáveis para a provisão de lógica, funções e operações da aplicação 1250.

[00704] A aplicação 1250 compreende um mecanismo de

determinação de aceleração de aplicativo 6275 e um programa de aceleração de lado de cliente 6120. O mecanismo de determinação de aceleração de aplicativo 6275 compreende um software, um hardware ou qualquer combinação de hardware e software. Em algumas modalidades, o mecanismo de determinação de aceleração de aplicativo 6275 compreende qualquer tipo e forma de instruções executáveis, tais como um programa, serviços, processo, tarefa ou linha tendo lógica, função, regras ou operações para determinar se um aplicativo 6220a a 6220n em execução em um cliente 6205 e/ou servidor 30 pode ser acelerado ou se acesso ou comunicações entre um cliente 6205 e um servidor 30 podem ser acelerados. Em uma modalidade, um banco de dados é usado pelo mecanismo de determinação de aceleração de aplicativo 6275 para determinar se um aplicativo 6220a a 6220n pode ser acelerado. Por exemplo, o banco de dados pode associar um aplicativo 6220a a 6220n com uma ou mais técnicas de aceleração capazes de acelerarem o aplicativo 6220a a 6220n, e ainda pode ser com base em usuário, tipo, forma, localização, capacidade de processamento e outras características do cliente 6205 e/ou servidor 30. Em algumas modalidades, o mecanismo de determinação de aceleração de aplicativo 6275 usa uma tabela de consulta, um arquivo, uma estrutura de dados ou um objeto na memória compreendendo uma informação identificando se um aplicativo 6220a a 6220n por nome, tipo ou categoria pode ser acelerado por uma técnica de aceleração. Em outras modalidades, a aplicação 1250 e/ou o mecanismo de determinação de aceleração de aplicativo 6275 inclui um mecanismo de configuração, tal como interface de usuário, gráfico, linha de comando ou de outra forma, para o recebimento de uma entrada de usuário para a identificação, especificação ou configuração quanto a se um aplicativo 6220a a 6220n ou acesso a um servidor 30 pode ser acelerado.

[00705] Em algumas modalidades, o mecanismo de determinação de aceleração de aplicativo 6275 requisita a partir do servidor 30 uma informação identificando se um aplicativo 6220a a 6220n pode ser acelerado e, em outras modalidades, por que técnica(s) de aceleração e para que tipo e forma de clientes 6205. Ainda em uma outra modalidade, o mecanismo de determinação de aceleração de aplicativo 6275 compreende um banco de dados de informação histórica referente à performance de um aplicativo 6220a a 6220n entre um cliente 6205 e um servidor 30, com e sem uma ou mais técnicas de aceleração de lado de cliente, para a provisão de um banco de dados de informação comparativa e heurística sobre onde o aplicativo 6220a a 6220n é acelerado ou capaz de ser acelerado, usando-se quaisquer técnicas de aceleração de lado de cliente. Por exemplo, a aplicação 1250 pode capturar uma informação de performance relacionada à rede relacionada à performance do aplicativo 6220a a 6220n a partir do cliente 6205. Como tal, a determinação quanto a se um aplicativo 6220a a 6220n é capaz de ser acelerado pode ser adaptada para ser com base em ou influenciada por uma mudança de características operacionais e de performance da rede 6204.

[00706] Em um aspecto, o aplicativo 6220a a 6220n pode não ser capaz de ser acelerado ou pode ser capaz de ser acelerado, mas a aceleração não seria efetiva ou de outra forma seria mínima. Em uma modalidade, o tipo e a forma de aplicativo 6220a a 6220n podem não usar um protocolo ou podem não se comunicar de uma maneira adequada para uso com uma técnica de aceleração. Em uma outra modalidade, o protocolo ou a maneira pela qual o aplicativo 6220a a 6220n se comunica pode permitir a realização de uma técnica de aceleração, mas, com base em qualquer uma das características operacionais ou de performance do cliente 6205, da aplicação 1250 ou

do servidor 30, a técnica de aceleração não seria efetiva ou de outra forma proveria uma aceleração mínima. Como tal, o mecanismo de determinação de aceleração de aplicativo 6275 pode determinar que não é desejado que o aplicativo 6220a a 6220n seja acelerado, com base em se o aplicativo 6220a a 6220n é capaz de ser acelerado ou se a aceleração se adequaria a um limite predeterminado desejado de melhoramento de performance.

[00707] Em um outro aspecto, a aplicação 6250 armazena um programa de aceleração de lado de cliente 6120 em um elemento de armazenamento ou de memória da aplicação 1250, tal como um armazenamento ou uma memória provido pela camada de hardware 6206 da aplicação. Em uma modalidade, a aplicação 1250 dinamicamente determina através do mecanismo de determinação de aceleração de aplicativo 6275 um aplicativo 6220a a 6220n a ser usado ou sendo usado pelo cliente 6205 que pode ser acelerado pelo programa de aceleração 6120 em execução no cliente 6205 e transmite ou de outra forma comunica o programa de aceleração 6120 a partir do armazenamento ou da memória da aplicação 1250 para o cliente 6205. Em uma outra modalidade, a aplicação 1250 determina que comunicações entre o cliente 6205 e um servidor 30 podem ser aceleradas pelo programa de aceleração 6120 em execução no cliente 6205 e comunica o programa de aceleração 6120 para o cliente 6205. Em algumas modalidades, a aplicação 1250 recebe, transfere (via download) ou obtém o programa de aceleração 6120 a partir de um outro dispositivo de computação 6100, tal como um servidor 30.

[00708] Em algumas modalidades, o programa de aceleração 6120 recebe, transfere (via download) ou obtém uma informação de política a partir do agente de política 3236 da aplicação 1250. Em outras modalidades, o programa de aceleração 6120 executa ou opera um agente de política, independentemente de ou em conjunto com o

agente de política 3236 da aplicação 1250. Em outras modalidades, o agente de pacote 3240 ou uma porção do mesmo pode ser operado no cliente 6205, tal como uma parte do programa de aceleração 6120. Como tal, o programa de aceleração 6120 pode operar no cliente 6205 de acordo com o temporizador de processamento de pacote 3242, conforme descrito acima. Em uma modalidade, o programa de aceleração 6120 pode realizar técnicas de aceleração integradas em um ponto em execução e em resposta a intervalos de tempo granulares providos pelo temporizador de processamento de pacote 3242.

[00709] Em algumas modalidades, o programa de monitoração de saúde 3216 pode checar e determinar o status, erro ou histórico de qualquer programa de aceleração de lado de cliente 6120 em qualquer cliente 6205 em comunicação com a aplicação 1250 ou para o qual a aplicação 1250 transmitiu o programa de aceleração 6120. Em algumas modalidades, o programa de monitoração de saúde 3216 ou uma porção do mesmo se executa no cliente 6205.

[00710] Com referência, agora, à figura 41A, uma modalidade de um método 6300 para provisão dinâmica pela aplicação 1250 de um programa de aceleração 6120 e para instalação automática e execução do programa de aceleração 6120 pelo cliente 6205 é descrita. Em uma visão geral breve, na etapa 6310, a aplicação 1250 intercepta uma requisição a partir de um cliente 6205 para o estabelecimento de uma sessão de comunicação com o servidor. Na etapa 6315, a aplicação 1250 transmite o programa de aceleração 6120 para o cliente 6205 para o cliente 6205 automaticamente instalar e executar. Na etapa 6320, mediante o recebimento do programa de aceleração 6120, o cliente 6205 automaticamente executa o programa de aceleração 6120 na pilha de rede 6210 para interceptação de comunicações entre o cliente 6205 e o servidor 30. Na etapa 6330, o

programa de aceleração 6120 realiza qualquer uma da pluralidade de técnicas de aceleração e pode encriptar e/ou desencriptar comunicações.

[00711] Em maiores detalhes, na etapa 6310, a aplicação 1250 pode interceptar ou receber de outra forma por quaisquer meios adequados e mecanismos uma requisição a partir do cliente 6205 para o estabelecimento de uma sessão de comunicação com o servidor 30. Em uma modalidade, o agente de pacote 6240 da aplicação 1250 intercepta comunicações a partir do cliente 6205. Em outras modalidades, a aplicação 1250 estabelece uma primeira conexão de camada de transporte com o cliente 6205, por exemplo, com o programa de aceleração 6120, e uma segunda conexão de camada de transporte com o servidor 6205 em nome do cliente 6205. Como tal, a aplicação 1250 pode receber, interceptar ou obter de outra forma qualquer uma das comunicações de cliente transmitidas para o servidor 30. Em algumas modalidades, a aplicação 1250 intercepta uma requisição para o cliente 6205 para o estabelecimento de uma conexão de camada de transporte com o servidor 30. Em outras modalidades, a aplicação 1250 intercepta uma requisição para o estabelecimento de uma sessão de comunicação através de qualquer camada de protocolo acima da conexão de camada de transporte, tal como um protocolo de camada de aplicativo de HTTP. Esta modalidade do método pode ser praticada com uma requisição para o estabelecimento de uma sessão de comunicação em qualquer camada de protocolo da pilha de rede 6210 do cliente 6205.

[00712] Na etapa 6315, a aplicação 1250 transmite o programa de aceleração 6120 para o cliente 6205. A aplicação 1250 pode transmitir o programa de aceleração 6120 em qualquer ponto antes, durante ou depois do estabelecimento da sessão de comunicação requisitada pelo cliente 6205. Em uma modalidade, a aplicação 1250 transmite o

programa de aceleração 6120 para o cliente 6205 em resposta à interceptação da requisição de cliente. Em uma outra modalidade, a aplicação 1250 encaminha a requisição para o servidor 30 e transmite o programa de aceleração 6120 para o cliente 6205. Em algumas modalidades, a aplicação 1250 estabelece a sessão de comunicação com o servidor 30 e, mediante o estabelecimento da sessão de comunicação, a aplicação 1250 transmite o programa de aceleração 6120. Em ainda uma outra modalidade, a aplicação 1250 realiza a autenticação e/ou a autorização do cliente 6205, ou o usuário do cliente 6205, e se o usuário ou cliente 6205 autenticado for autorizado dessa forma, a aplicação 1250 transmite o programa de aceleração 6120 para o 6205. Em uma modalidade, a aplicação 1250 encaminha a requisição de cliente para o servidor 30 para autenticação e/ou autorização e, se o servidor 30 autenticar e/ou autorizar a requisição de cliente, a aplicação 1250 transmitirá o programa de aceleração 6120 para o cliente 6205.

[00713] Em algumas modalidades, a aplicação 1250 transmite o programa de aceleração 6120 a partir do armazenamento ou da memória da aplicação 1250. Em outras modalidades, a aplicação 1250 requisita o programa de aceleração 6120 a partir do servidor 30 e encaminha o programa de aceleração recebido 1620 para o cliente 6205. Em uma outra modalidade, o servidor 30 transmite o programa de aceleração 6120 para o cliente 6205. Em uma modalidade, a aplicação 1250 transmite um Localizador de Recurso Uniforme (URL) para o cliente 6205 para o cliente 6205 obter, transferir (via download) ou receber o programa de aceleração. Em algumas modalidades, o URL identifica uma localização do programa de aceleração 6120 em um armazenamento ou uma em da aplicação 1250, enquanto em outras modalidades o URL identifica o programa de aceleração 6120 em um servidor 30, tal como um servidor da web provendo o programa

de aceleração 6120 para transferência (via download). Em uma modalidade, o programa de aceleração 6120 é armazenado no cliente 6205, e a aplicação 1250 transmite uma chave, tal como uma chave de encriptação ou de licença, para o cliente 6205 para o cliente 6205 instalar e fazer uso do programa de aceleração 6120 armazenado no cliente 6205. Em algumas modalidades, a aplicação 1250 transmite ao cliente 6205 quaisquer arquivos, configuração, dados ou outra informação a ser usada para instalação e execução do programa de aceleração 6120 no cliente 6205.

[00714] Em uma modalidade, o programa de aceleração 6120 é projetado e construído para ser automaticamente instalado e executado pelo cliente 6205. O programa de aceleração 6120 pode incluir quaisquer arquivos, entradas, configuração, dados ou instruções para se fazer com que o programa de aceleração 6120 seja registrado ou reconhecido pelo sistema operacional do cliente 6205, de acordo com o tipo e a forma de sistema operacional. Em uma modalidade, um outro dispositivo de computação, tal como um servidor ou uma aplicação, transmite o programa de aceleração para o cliente 6205, e o cliente 6205 automaticamente instala e executa o programa de aceleração 6120. Em uma modalidade, o programa de aceleração 6120 é projetado e construído para ser um dispositivo de *plug-and-play* (PnP) a ser adicionado a um dispositivo de computação rodando 6100. Em algumas modalidades, o programa de aceleração 6120 é um executável auto-instalado, tal como um executável incluindo um programa instalador e o programa de aceleração 6120. Em outras modalidades, o programa de aceleração 6120 pode incluir uma pluralidade de arquivos, por exemplo, um pacote de instalação ou uma transferência (via download) de instalação, tais como arquivos necessários para registro e instalação do programa de aceleração 6120 no sistema operacional do cliente 6205. Por exemplo, o

programa de aceleração 6120 pode compreender um arquivo .inf e um arquivo .sys. Um arquivo .inf provê um Windows Setup na família Microsoft Windows de sistemas operacionais com a informação requerida para a configuração de um dispositivo, tal como uma lista de configurações lógicas válidas para o dispositivo e os nomes de arquivos de driver associados ao dispositivo. Em algumas modalidades, o arquivo .inf pode compreender um arquivo autorun .inf, o qual é um arquivo de configuração que conta ou informa ao sistema operacional qual executável iniciar e qualquer informação de configuração relacionada ao início do executável. Em uma modalidade, o arquivo .sys é o arquivo de driver compreendendo o programa de aceleração 6120 ou uma porção do mesmo.

[00715] Na etapa 6320, o cliente 6205 automaticamente instala o programa de aceleração 6120. O programa de aceleração 6120 pode ser instalado de qualquer maneira adequada de acordo com o sistema operacional do cliente 6205. Em uma modalidade, o cliente 6205 instala o programa de aceleração 6120 mediante o recebimento do programa de aceleração 6120. Em algumas modalidades, o cliente 6205 automaticamente realiza ou executa uma instalação silenciosa do programa de aceleração 6120. Em uma modalidade, a instalação silenciosa é realizada de forma transparente para um usuário ou aplicativo do cliente 6205. Em outras modalidades, a instalação silenciosa do programa de aceleração 6120 não requer um reboot ou um recomeço do cliente 6205. Em uma outra modalidade, a instalação silenciosa não requer uma interação pelo usuário para se começar e/ou completar a instalação. Em outras modalidades, a instalação silenciosa do programa de aceleração 6120 ocorre, enquanto o cliente 6205 está rodando e de forma transparente para uma camada de rede, camada de sessão e/ou camada de aplicativo da pilha de rede 6210. Em algumas modalidades, o programa de aceleração 6120 é um

executável auto-instalado que é executado pelo cliente 6205. Em outras modalidades, o cliente 6205 usa um gerenciador do tipo *plug-and-play* para a instalação do programa de aceleração 6120. Em uma modalidade, o cliente 6205 compreende um gerenciador de instalação o qual recebe e instala o programa de aceleração 6120. Em uma outra modalidade, o programa de aceleração 6120 transmitido pela aplicação 1250 também inclui um programa de instalação que instala o programa de aceleração 6120.

[00716] Em uma outra modalidade, o programa de aceleração 6120 é automaticamente instalado através de uma instalação silenciosa. Em uma modalidade, uma instalação silenciosa compreende uma instalação não atendida por um usuário. Em uma outra modalidade, uma instalação silenciosa compreende uma instalação não requerendo ou tendo uma interação pelo usuário para começar e/ou completar a instalação. Em algumas modalidades, a instalação é silenciosa pelo fato de o processo de instalação não exibir uma informação referente a um status ou progresso da instalação. Em uma modalidade, a instalação é silenciosa pelo fato de o processo de instalação não exibir uma informação referente ao status ou progresso da instalação. Em uma modalidade, a instalação é silenciosa pelo fato de ser transparente para o usuário. Em outras modalidades, a instalação é silenciosa pelo fato de a instalação do programa de aceleração 6120 não requerer um reboot ou um reinício do cliente 6205. Em uma outra modalidade, a instalação é silenciosa pelo fato de a instalação ocorrer sem emendas durante uma operação do cliente 6205, sem interrupção ou perturbação da operação do cliente. Como tal, o programa de aceleração 6120 pode ser instalado de uma maneira que seja transparente para o usuário ou um aplicativo do cliente 6205 por não requerer um reboot e não exibindo qualquer informação para o usuário relacionado à instalação.

[00717] De modo a prevenir ou evitar um reboot ou um reinício do cliente 6205, em algumas modalidades, o cliente 6205, tal como o sistema operacional do cliente 6205, tem um gerenciador de plug and play para instalação e configuração de drivers, tal como um driver de rede em uma modalidade do programa de aceleração 6120, para dispositivos de Plug and Play, enquanto o sistema operacional estiver rodando. Em uma modalidade, o gerenciador de plug and play não é instruído para rebutar ou reiniciar o cliente 6205, com base na configuração do pacote de instalação do programa de aceleração 6120. Em uma outra modalidade, o arquivo .inf não compreende uma instrução para rebutar ou reiniciar o computador. Em uma modalidade, o programa de aceleração 6120 pode ser implementado como um componente lado a lado, ao invés da substituição de bibliotecas de link dinâmico (DLLs) em uso compartilhadas. Em outras modalidades específicas, para um driver de rede do programa de aceleração 6120, o programa de aceleração 6120 usa a API de driver de rede INetCfgPnpReconfigCallback, de modo que não seja requerido que o usuário rebute o sistema operacional para fazer com que mudanças de configuração tenham efeito no driver. Adicionalmente, o programa de aceleração 6120 pode ter um objeto de notificação que chama a API SendPnpReconfig em sua implementação do método ApplyPnpChanges do INetCfgComponentControl para envio de uma informação de configuração para o driver do componente de rede que possui o objeto. A API SendPnpReconfig provê um objeto de notificação com um mecanismo para o envio de dados para o driver e, em algumas modalidades, é usada para se evitar requerer que um usuário rebute o sistema operacional antes de as mudanças de configuração terem efeito.

[00718] Na etapa 6325, mediante a conclusão da instalação do programa de aceleração 6120 de forma automática, silenciosa,

transparente ou de outra forma, o programa de aceleração 6120 é automaticamente executado no cliente 6205. Em algumas modalidades, o programa de instalação que instala o programa de aceleração 6120 inicia ou executa o programa de aceleração 6120. Em algumas modalidades, o programa instalador para o programa de aceleração 6120 faz uma chamada de sistema para carregar ou executar o programa de aceleração 6120 na memória do cliente 6205. Em uma modalidade, a instalação do programa de aceleração 6120 compreende uma instrução, um comando ou uma diretiva para se iniciar o programa de aceleração 6120. Em uma modalidade, o programa de aceleração 6120 inclui uma configuração de rodada automática, tal como um arquivo autorun.inf, que notifica o cliente 6205 para automaticamente rodar o programa de aceleração 6120. Em outras modalidades, um gerenciador de plug and play ou o sistema operacional do cliente 6205 executa automaticamente o programa de aceleração 6120 quando da instalação. Em uma modalidade, o programa de aceleração 6120 compreende um serviço, processo, uma linha ou tarefa, que é iniciado pelo cliente 6205. Em algumas modalidades, o programa de aceleração 6120 é um serviço do sistema operacional que é configurado para iniciar automaticamente. Em uma modalidade, o programa de aceleração 6120 compreende um driver de rede carregado na memória da pilha de rede do sistema operacional do cliente.

[00719] Em uma outra modalidade, o programa de aceleração 6120 compreende um driver de rede que é carregado na memória do cliente 6205. Em algumas modalidades, o programa de aceleração 6120 é carregado em uma memória alocada à pilha de rede 6210. Em alguns casos, o programa de aceleração 6120 é carregado e executado em uma área ou um espaço de memória que permite que o programa de aceleração 6120 acesse uma camada de protocolo da pilha de rede,

tal como a camada de transporte. Em outros casos, o programa de aceleração é carregado e executado em uma memória que permite que o programa de aceleração 6120 acesse uma estrutura de dados de nível de kernel 6225. Em outras modalidades, o programa de aceleração 6120 é carregado na memória de um aplicativo 6220a a 6220n. Em uma outra modalidade, o programa de aceleração 6120 se executa independentemente em seu próprio espaço de memória ou contexto. Em uma modalidade, o programa de aceleração 6120 roda no espaço de memória ou contexto de um aplicativo 6220a a 6220n. Em algumas modalidades, o programa de aceleração 6120 é carregado em uma memória de modo de usuário ou uma memória alocada ao modo de usuário 6203, enquanto em outras modalidades o programa de aceleração 6120 é carregado em uma memória de modo de kernel ou uma memória alocada ao modo de kernel 6202.

[00720] Em algumas modalidades, o programa de aceleração 6120 é carregado em uma memória e/ou executado no cliente 6205 de forma transparente para um usuário do cliente, um aplicativo do cliente 6205, a aplicação 1250 ou o servidor 30. Em outras modalidades, o programa de aceleração 6120 se executa para ter uma interface com a camada de transporte da pilha de rede 6210, e se executa de forma transparente para qualquer camada de protocolo acima da camada de transporte, tal como uma camada de sessão ou de aplicativo, e qualquer camada de protocolo abaixo da camada de transporte, tal como a camada de rede. Em uma modalidade, o programa de aceleração 6120 se executa de forma transparente para qualquer conexão de camada de transporte do cliente 6205, ou a camada de transporte em si.

[00721] Na etapa 6330, o programa de aceleração 6120 carregado, iniciado ou em execução de outra forma realiza qualquer uma da pluralidade de técnicas de aceleração do programa de aceleração

6120, tais como quaisquer técnicas providas por: 1) compressão de protocolo múltiplo 6238, 2) agrupamento de protocolo de controle de transporte 6224, 3) multiplexação de protocolo de controle de transporte 6226, 4) armazenamento em buffer de protocolo de controle de transporte 6228, e 5) armazenamento em cache através de um gerenciador de cache 6232. O programa de aceleração 6120 também pode realizar qualquer encriptação e/ou desencriptação de comunicações entre o cliente 6205 e o servidor 30. Em uma modalidade, o programa de aceleração 6120 realiza uma compressão de protocolo múltiplo. Em uma outra modalidade, o programa de aceleração 6120 realiza um agrupamento de protocolo de controle de transporte e em uma modalidade adicional, o programa de aceleração 6120 realiza uma multiplexação através da conexão de camada de transporte agrupada. Em uma modalidade, o programa de aceleração 6120 realiza um armazenamento em buffer de protocolo de controle de transporte. Em algumas modalidades, o programa de aceleração 6120 realiza um armazenamento em cache. Em outras modalidades, o programa de aceleração 6120 realiza um armazenamento em cache e uma compressão. Em uma modalidade, o programa de aceleração 6120 realiza um armazenamento em cache com um agrupamento de camada de transporte e multiplexação. Em uma outra modalidade, o programa de aceleração 6120 realiza uma compressão de protocolo múltiplo com um agrupamento de camada de transporte e uma multiplexação. Em uma outra modalidade, o programa de aceleração 6120 realiza um armazenamento em cache e/ou uma compressão com armazenamento em buffer de TCP, e em uma modalidade adicional, com agrupamento de TCP e multiplexação.

[00722] Como tal, o programa de aceleração 6120 é provido dinamicamente pela aplicação 1250 e automaticamente instalado e executado no cliente 6205 de uma maneira silenciosa ou transparente

para o usuário ou aplicativo do cliente 6205 para a realização de uma ou mais técnicas de aceleração de lado de cliente para comunicações entre o cliente 6205 e um servidor 30. O programa de aceleração 6120 pode realizar estas técnicas de aceleração de forma transparente para qualquer camada de protocolo da pilha de rede e de forma transparente para um usuário do cliente, aplicativo do cliente, aplicação ou servidor.

[00723] Em um outro aspecto, a aplicação 1250 pode determinar se um aplicativo requisitado para ser acessado pelo cliente 6205 pode ser acelerado, e prover o programa de aceleração 6120 para o cliente 6205, se o aplicativo puder ser acelerado. Com referência, agora, à figura 41B, uma outra modalidade de um método é descrita. O método pode ser praticado mediante requisições para o estabelecimento de uma conexão ou uma sessão de comunicação bem como requisições para acesso a um aplicativo em um servidor. Em uma visão geral breve do método 6350, na etapa 6355, a aplicação 1250 intercepta uma requisição de um cliente 6205 requisitando acesso a um aplicativo 6220a a 6220n em um servidor 30. Na etapa 6260, a aplicação 1250 determina se o aplicativo 6220 é capaz de ser acelerado. Na etapa 6265, se o aplicativo 6220 não puder ser acelerado, então, o aplicativo encaminha a requisição para o servidor na etapa 6267. Na etapa 6365, se o aplicativo 6220 puder ser acelerado, então a aplicação 1250 determina se o programa de aceleração 6120 está instalado no cliente 6205 ou foi transmitido previamente para o cliente 6205. Se o programa de aceleração 6120 não tiver sido provido ainda para o cliente 6205, então, o método 6250 continua na etapa 6315 do método 6300 descrito acima para a transmissão, instalação e execução do programa de aceleração. Se o programa de aceleração 6120 tiver sido instalado e estiver em execução no cliente 6205, então a aplicação 1250, na etapa 6375,

envia uma mensagem para o programa de aceleração 6120 no cliente 6205 para acelerar o aplicativo 6220. Na etapa 6330 do método 6350, o programa de aceleração 6120 realiza uma pluralidade de técnicas de aceleração nas comunicações para o aplicativo 6220, e pode encriptar e/ou desencriptar essas comunicações.

[00724] Em maiores detalhes, na etapa 6355, a aplicação 1250 pode interceptar por quaisquer meios e mecanismos adequados uma requisição a partir do cliente 6205 para acessar um aplicativo provido pelo servidor 30. Em uma modalidade, o agente de pacote 6240 da aplicação 1250 intercepta comunicações a partir do cliente 6205. Em outras modalidades, a aplicação 1250 estabelece uma primeira conexão de camada de transporte com o cliente 6205, por exemplo, com o programa de aceleração 6120, e uma segunda conexão de camada de transporte com o servidor 6205 em nome do cliente 6205. Como tal, a aplicação 1250 pode receber, interceptar ou obter de outra forma qualquer uma das comunicações de cliente transmitidas para o servidor 30. Em algumas modalidades, a aplicação 1250 intercepta uma requisição para o cliente 6205 acessar um aplicativo 6220 através de uma conexão de camada de transporte estabelecida com o servidor 30. Em outras modalidades, a aplicação 6205 intercepta uma requisição para o estabelecimento de uma sessão de comunicação através de qualquer camada de potência de transmissão acima da conexão de camada de transporte, tal como um protocolo de camada de aplicativo de HTTP. Em uma modalidade, a aplicação 6205 intercepta uma requisição a partir do cliente 205 para exibir e prover um aplicativo 6220 a partir do servidor 30 através de um protocolo de exibição remota, tal como ICA ou RDP.

[00725] Na etapa 6360, a aplicação 1250 determina se o aplicativo 6220 requisitado pelo cliente 6205 pode ser acelerado. Em algumas modalidades, a aplicação 1250 identifica, extrai ou processa de outra

forma um identificador de aplicativo a partir da requisição de cliente interceptada que identifica o aplicativo por nome, tipo ou categoria. Em uma modalidade, o mecanismo de determinação de aceleração de aplicativo 6275 é usado pela aplicação 1250 para se determinar se o aplicativo 6220 pode ser acelerado. Em algumas modalidades, o mecanismo de determinação de aceleração de aplicativo 6275 realiza uma pesquisa ou consulta em um banco de dados, uma tabela de consulta ou outra fonte estruturada de dados em uma memória ou um armazenamento, tal como uma estrutura de dados ou um objeto, para determinar se o aplicativo 6220 pode ser acelerado. Em uma outra modalidade, a aplicação 1250 envia uma comunicação, tal como uma requisição, para um servidor 30 para determinar se o aplicativo 6220 pode ser acelerado.

[00726] Em outras modalidades, a aplicação 1250 tem um registro ou histórico de performance para determinar se o aplicativo 6220 foi acelerado antes e se a aceleração teve um melhoramento na performance e na operação do aplicativo 6220. Como tal, a aplicação 1250 pode determinar que um aplicativo 6220 pode ser acelerado, se essa aceleração se adequar a um limite predeterminado de melhoramento na performance ou nas operações do aplicativo 6220. Ainda em uma outra modalidade, a aplicação 1250 provê regras heurísticas com base na operação atual e na performance da rede 6204, do cliente 6205 ou do servidor 30. Em uma modalidade, o aplicativo 6220 pode ser determinado como sendo capaz de ser acelerado, se o cliente 6205 tiver certas características ou capacidades de performance e operacionais, por exemplo, um processador de certa velocidade ou uma quantidade mínima de memória. Em algumas modalidades, o aplicativo 6220 pode ser determinado como sendo capaz de ser acelerado com base em uma política ou regra configurada, tal como no gerenciador de política da

aplicação 1250. Por exemplo, um aplicativo 6220 a ser comunicado entre um usuário remoto com um certo tipo de cliente 6205 acessando um certo tipo de aplicativo 6220 e/ou servidor 30 pode ser acelerado. Em outras modalidades, o aplicativo 6220 pode ser determinado como sendo capaz de aceleração com base em uma autenticação e uma autorização do usuário ou do cliente 6205. Ainda em uma outra modalidade, o aplicativo 6220 pode ser determinado como não sendo desejado que seja acelerado. Por exemplo, o aplicativo 6220 é de um tipo que não é usado freqüentemente.

[00727] Na etapa 6365, se o aplicativo 6220 for determinado como não sendo capaz de ser acelerado ou de outra forma for desejado não aplicar técnicas de aceleração ao aplicativo 6220 no cliente 6205, a aplicação 1250 encaminha a requisição de cliente interceptada para o servidor 30 na etapa 6368 e não transmite ou provê o programa de aceleração 6120 para o cliente 6205. Em uma modalidade, a aplicação 1250 pode realizar ou prover uma aceleração baseada em aplicação da aplicação 6220. Em outras modalidades, a aplicação 1250 não realiza uma aceleração do aplicativo 6220 na aplicação 1250. Ainda em uma outra modalidade, a aplicação 1250 pode realizar algumas técnicas de aceleração e não outras para o aplicativo 6220, se a aplicação 1250 determinar que o aplicativo 6220 não é capaz de ser ou de outra forma desejado que seja acelerado.

[00728] Na etapa 6365, se o aplicativo 6220 for determinado como sendo capaz de ser acelerado ou de outra forma for desejado aplicar técnicas de aceleração ao aplicativo no cliente 6205, a aplicação 1250 determinará se o programa de aceleração 6120 foi provido para o cliente 6205. Em uma modalidade, a aplicação 1250 determina se o programa de aceleração 6120 foi instalado no cliente 6205 ou está em execução no cliente 6205. Em algumas modalidades, a aplicação 1250 envia uma comunicação para o programa de aceleração 6120 em um

cliente 6205 para determinar se o programa de aceleração 6120 está rodando no cliente 6205. Em outras modalidades, a aplicação 1250 checa um arquivo de registro ou um arquivo de histórico para determinar se o programa de aceleração 6120 foi transmitido para o cliente 6205. Em uma outra modalidade, a aplicação 1250 checa com um programa de monitoração de saúde 6216 da aplicação 1250 ou do cliente 6205, para determinar se o programa de aceleração 6120 está em execução no cliente 6205.

[00729] Se a aplicação 1250 determinar que o programa de aceleração 6120 não foi transmitido, instalado e/ou executado no cliente 6205, a aplicação 1250 proverá o programa de aceleração 6120 de acordo com as etapas de método 6300 descritas em conjunto com a figura 41A. Por exemplo, a aplicação 1250 transmite o programa de aceleração 6120 para o cliente 6205, o qual o cliente 6205 mediante o recebimento instala e executa. Em uma modalidade, mediante a realização das etapas adequadas da modalidade de método 6300, a aplicação 1250 pode comunicar na etapa 6275 uma mensagem para o programa de aceleração para aplicação de uma ou mais das técnicas de aceleração ao aplicativo 6220. Em outras modalidades, se o programa de aceleração 6120 já estiver instalado e em execução, então, na etapa 6375, a aplicação 1250 comunica uma mensagem para o programa de aceleração 6120 para aplicar uma ou mais das técnicas de aceleração ao aplicativo 6220.

[00730] Em algumas modalidades, o programa de aceleração 6120 realiza qualquer uma das técnicas de aceleração disponíveis pelo programa de aceleração 6120 para o aplicativo identificado 6120. Em outra modalidade, a aplicação 1250 indica para o programa de aceleração 6120 qual das técnicas de aceleração realizar para o aplicativo 6220. Em uma modalidade, o programa de aceleração 6120

pode aplicar as técnicas de aceleração desejadas para o aplicativo 6120 em uma base por sessão. Isto é, a mensagem a partir da aplicação 1250 para o programa de aceleração 6120 apenas informa o programa de aceleração 6120 para realizar técnicas de aceleração para esta instância ou sessão do aplicativo 6220. Em outras modalidades, uma vez que o programa de aceleração 6120 receba uma mensagem a partir da aplicação 1250 para aplicação de técnicas de aceleração para o aplicativo identificado 6220, o programa de aceleração 6120 aplica as técnicas de aceleração para quaisquer instâncias ou sessões do aplicativo 6220, ou até o cliente 6205 ser rebutado ou reiniciado, ou a aplicação 6205 ser rebutada ou reiniciada.

[00731] Em uma modalidade, a mensagem a partir da aplicação 1250 na etapa 6375 não é específica de aplicativo. Por exemplo, a mensagem informa o programa de aceleração 6120 para executar uma ou mais das técnicas de aceleração para qualquer aplicativo do cliente 6205. Em algumas modalidades, a mensagem enviada para o cliente 6205 informa o programa de aceleração 6120 para parar de usar qualquer uma ou mais das técnicas de aceleração para o aplicativo 6220, ou para todos os aplicativos 6220a a 6220n. Em uma outra modalidade, a aplicação 1250 comunica uma mensagem para o programa de aceleração 6120 para ignorar certos aplicativos 6220. Ainda em uma outra modalidade, a aplicação 1250 comunica uma mensagem para o programa de aceleração 6120 para prover dados de configuração ou uma informação para o programa de aceleração 6120, tal como uma atualização para uma técnica de aceleração ou uma aplicação de uma nova técnica de aceleração.

[00732] Na etapa 6330, o programa de aceleração 6120 realiza qualquer uma da pluralidade de técnicas de aceleração do programa de aceleração 6120 para o aplicativo 6220, tais como quaisquer técnicas providas por 1) compressão de protocolo múltiplo 6238, 2)

agrupamento de protocolo de controle de transporte 6224, 3) multiplexação de protocolo de controle de transporte 6226, 4) armazenamento em buffer de protocolo de controle de transporte 6228, e 5) armazenamento em cache através de um gerenciador de cache 6232. O programa de aceleração 6120 também pode realizar qualquer encriptação e/ou desencriptação de comunicações entre o cliente 6205 e o servidor 30. Em uma modalidade, o programa de aceleração 6120 realiza uma compressão de protocolo múltiplo de dados relacionados a aplicativo. Em uma outra modalidade, o programa de aceleração 6120 realiza um agrupamento de protocolo de controle de transporte e, em uma modalidade adicional, o programa de aceleração 6120 realiza uma multiplexação através da conexão de camada de transporte agrupada. Em uma modalidade, o programa de aceleração 6120 realiza um armazenamento em buffer de protocolo de controle de transporte. Em algumas modalidades, o programa de aceleração 6120 realiza um armazenamento em cache. Em outras modalidades, o programa de aceleração 6120 realiza um armazenamento em cache e uma compressão. Em uma modalidade, o programa de aceleração 6120 realiza um armazenamento em cache com um agrupamento de camada de transporte e em uma modalidade adicional também com multiplexação. Em uma outra modalidade, o programa de aceleração 6120 realiza uma compressão de protocolo múltiplo com um armazenamento em buffer de TCP, e em uma modalidade adicional com agrupamento de camada de transporte e, ainda em uma modalidade adicional também com multiplexação. Em uma outra modalidade, o programa de aceleração 6120 realiza um armazenamento em cache com uma compressão e, em uma modalidade adicional, com agrupamento de TCP e ainda em uma modalidade adicional com multiplexação.

[00733] Como tal, uma aplicação 1250 dinamicamente determina se

é para acelerar um aplicativo ou se o aplicativo pode ser acelerado, e se comunica com o programa de aceleração de lado de cliente 6120 para realização no cliente 6205 de qualquer uma ou mais das técnicas de aceleração para o aplicativo 6220. Mais ainda, em algumas modalidades, uma pluralidade de programas de aceleração 6120 pode ser dinamicamente entregue para o cliente 6205 pela aplicação e automaticamente instalada e executada pelo cliente 6205. Por exemplo, um programa de aceleração pode ser provido de acordo com as técnicas e os métodos para cada conexão com um servidor 6205, ou cada sessão de comunicação com um aplicativo 6220. Como tal, o cliente 6205 pode instalar automaticamente e executar uma pluralidade de programas de aceleração 6120 para manipulação e realização de uma aceleração de cada servidor 630 ou cada aplicativo 6220a a 6220n.

[00734] Com referência, agora, à figura 41C, uma modalidade de um método 6380 para a realização de uma pluralidade de técnicas de aceleração de uma maneira integrada é descrita. Em uma breve visão geral, na etapa 6280, o programa de aceleração 6120 intercepta na camada de transporte um pacote de rede de uma comunicação entre o cliente 6205 e o servidor 30 através de uma conexão de camada de transporte. Na etapa 6390, o programa de aceleração 6120 acessa na camada de transporte o pacote de rede através de uma estrutura de dados de nível de kernel, por exemplo, uma estrutura de dados provida com uma API para a pilha de rede 6210 do cliente 6205. Na etapa 6395, o programa de aceleração 6120 realiza uma pluralidade de técnicas de aceleração de uma maneira integrada, usando a estrutura de dados de nível de kernel em um ponto de interface ou em um ponto de execução no programa de aceleração 6120.

[00735] Em maiores detalhes, na etapa 6385, o programa de aceleração 6120 intercepta por quaisquer meios e mecanismos

adequados um pacote de rede de uma comunicação entre o cliente 6205 e o servidor 30 através de uma conexão de camada de transporte. Em uma modalidade, o programa de aceleração 6120 intercepta um pacote de rede de ou relacionado a uma requisição pelo cliente ou uma resposta a isso, para o estabelecimento de uma conexão de camada de transporte entre o cliente 6205 e o servidor 30. Em uma outra modalidade, o programa de aceleração 6120 intercepta um pacote de rede de ou relacionado a uma requisição ou uma resposta a isso, para acesso ou uso de um aplicativo 6220 através da conexão de camada de transporte entre o cliente 6205 e o servidor 30. Em uma modalidade, o programa de aceleração 6120 intercepta o pacote de rede na camada de protocolo de transporte através de uma interface de driver de transporte ou de outra forma um driver de rede interfaceado em uma camada de protocolo de transporte da pilha de rede 6210. Em uma outra modalidade, programa de aceleração 6120 intercepta o pacote de rede na camada de protocolo de transporte ou qualquer outra camada de protocolo da pilha de rede 6210 através de um driver de Especificação de Interface de Driver de Rede (NDIS), ou um driver de miniporta, ou um driver de minifiltro. Em algumas modalidades, o programa de aceleração 6120 intercepta o pacote de rede na camada de transporte através de um mecanismo de enganchamento ou filtração.

[00736] Na etapa 6390, o programa de aceleração 6120 acessa ou obtém de outra forma informação e dados do pacote de rede interceptado na camada de transporte através de uma estrutura de dados de nível de kernel 6225. Pelo uso da estrutura de dados de nível de kernel 6225, o programa de aceleração 6120 pode obter uma informação e dados sobre a(s) carga(s) útil(eis) ou um ou mais protocolos portados ou transportados pelo pacote de rede na camada de transporte. Em algumas modalidades, usar uma estrutura de dados

de nível de kernel para representação do pacote de rede nas camadas da pilha de rede na e/ou acima da camada de transporte permite que o programa de aceleração 6120 realize ou opere a pluralidade de técnicas de aceleração na camada de transporte e para camadas de protocolo portadas pelo pacote de rede de camada de transporte. Em uma modalidade, o uso de uma única estrutura de dados de nível de kernel 6225 impede ou evita uma cópia e alocação de memória juntamente com uma comutação de contexto usando-se múltiplas estruturas de dados em várias camadas de protocolo da pilha de rede 6210. Em uma modalidade, o programa de aceleração 6120 copia a estrutura de dados de nível de kernel 6225 para uma segunda estrutura de dados, a qual pode compreender uma outra estrutura de dados de nível de kernel ou uma estrutura de dados de nível de usuário.

[00737] Na etapa 6395, o programa de aceleração 6120 realiza, executa ou opera a pluralidade de técnicas de aceleração em um ponto de interface único ou um ponto de execução no programa 6210 ou em um conjunto de instruções executáveis ou um ponto de execução do programa 6210. O programa de aceleração 6120 realiza qualquer uma da pluralidade de técnicas de aceleração do programa de aceleração 6120, tais como quaisquer técnicas providas por: 1) compressão de protocolo múltiplo 6238, 2) agrupamento de protocolo de controle de transporte 6224, 3) multiplexação de protocolo de controle de transporte 6226, 4) armazenamento em buffer de protocolo de controle de transporte 6228, e 5) armazenamento em cache através de um gerenciador de cache 6232. O programa de aceleração 6120 também pode realizar qualquer encriptação e/ou desencriptação de comunicações do aplicativo 6220 entre o cliente 6205 e o servidor 30 no mesmo ponto de execução das técnicas de aceleração do programa de aceleração 6120.

[00738] Em uma modalidade, o programa de aceleração 6120 realiza em um conjunto de instruções executáveis, tal como uma chamada de função ou um local ou uma localização, qualquer pluralidade desejada das técnicas de aceleração subseqüentemente umas às outras. Por exemplo, o programa de aceleração 6120 obtém o pacote de rede interceptado através de uma estrutura de dados de nível de kernel e, então, executa as instruções representando a lógica, a função, as regras ou a operação das técnicas de aceleração subseqüentemente umas às outras. Como tal, uma informação e dados do pacote de rede podem ser extraídos ou obtidos uma vez através da estrutura de dados de nível de kernel 6225 e usados como entrada, parâmetros, argumentos e condições para qualquer uma das instruções do programa de aceleração 6120 representando as técnicas de aceleração. Embora o pacote de rede porte dados de protocolo de nível mais alto e uma informação, o programa de aceleração 6120 em algumas modalidades processa o pacote de rede e os dados de protocolo de nível mais alto e uma informação em um ponto e em um tempo durante uma execução. Adicionalmente, o programa de aceleração 6120 pode realizar cada uma de uma pluralidade de técnicas de aceleração em qualquer ordem desejada de uma maneira integrada, tal como uma compressão de dados armazenados no gerenciador de cache 6232 ou uma compressão/descompressão de dados recuperados a partir do cache.

[00739] Em uma modalidade, o programa de aceleração 6120 realiza uma compressão de protocolo múltiplo e um armazenamento em cache subseqüentemente a cada outro. Em uma outra modalidade, o programa de aceleração 6120 realiza, subseqüentemente a cada outra, operações relacionadas a um agrupamento de protocolo de controle de transporte e multiplexação através da conexão de camada de transporte agrupada. Em uma modalidade, o programa de

aceleração 6120 realiza um armazenamento em buffer de protocolo de controle de transporte subseqüentemente a uma compressão e um armazenamento em cache, ou a um agrupamento de TCP e/ou multiplexação. Em algumas modalidades, o programa de aceleração 6120 realiza um armazenamento em cache. Em uma modalidade, o programa de aceleração 6120 realiza um armazenamento em cache subseqüentemente a um agrupamento de camada de transporte e multiplexação. Em uma outra modalidade, o programa de aceleração 6120 realiza uma compressão de protocolo múltiplo subseqüentemente a um agrupamento de camada de transporte e uma multiplexação. Em uma modalidade, o programa de aceleração 6120 realiza um armazenamento em cache e uma compressão subseqüentemente com um armazenamento em buffer de TCP e em uma modalidade adicional subseqüentemente com agrupamento de TCP e multiplexação.

[00740] Embora o programa de aceleração seja geralmente descrito como subseqüentemente realizando as técnicas de aceleração, uma execução subseqüente também pode incluir outra lógica, funções e operações não relacionadas a uma aceleração, mas integradas e executadas entre cada técnica de aceleração. O programa de aceleração ainda obtém uma eficiência operacional e de performance com essa integração conforme as instruções executáveis para as técnicas de aceleração e quaisquer outras operações ou funções são executadas em um ponto de interface único ou ponto de execução no programa de aceleração. Mais ainda, as técnicas de aceleração para camadas de protocolo portadas ou acima da camada de protocolo de transporte são processadas em um tempo e/ou em uma localização na camada de transporte. Como tal, as técnicas de aceleração para estes protocolos de nível mais alto não precisam ser aplicadas de novo conforme o pacote de rede atravessar e for processado nestes níveis

mais altos da pilha de rede 6210, ou em um ponto posterior na pilha de rede 6210.

[00741] Em outros aspectos, um primeiro programa 6222 e o programa de aceleração 6120 (ou também referido como o segundo programa nesta modalidade) podem ser usados. Em uma modalidade, o primeiro programa 6222 juntamente com o segundo programa 6120 podem ser usados para facilitação e estabelecimento de uma conexão de rede privada virtual com um servidor 30, tal como através da aplicação 1250, pela qual as técnicas de aceleração de lado de cliente podem ser aplicadas. Em uma outra modalidade, o primeiro programa 6222 é usado para instalação e execução do segundo programa ou do programa de aceleração 6120.

[00742] Com referência, agora, à figura 42A, uma modalidade de um método 6400 para a prática deste aspecto é descrita. Em uma visão geral breve, na etapa 6402, o cliente 6205 faz um login e estabelece uma sessão de comunicação com a aplicação 6205. Na etapa 6404, a aplicação 1250 envia o primeiro programa 6222 para o cliente 6205. Na etapa 6406, o cliente 6205 instala e executa o primeiro programa 6222, o qual, por sua vez, instala e executa o programa de aceleração 6120, isto é, o segundo programa. Na etapa 6407, o cliente 6205 se comunica com e acessa recursos em uma rede privada através de uma sessão de comunicação de dados encriptados estabelecida. Na etapa 6410, o cliente 6205 faz um logout da aplicação 1250 e termina a sessão de comunicação com a aplicação 1250.

[00743] Na etapa 6402 de método 6400, o cliente 6205 realiza um procedimento de login e estabelece uma sessão de comunicação de dados encriptados com a aplicação 1250 através da rede 6204. Em uma modalidade, a sessão de comunicação de dados encriptados é usada como um túnel para formar uma ponte de tráfego do cliente

6205 para qualquer um dos servidores 30, os quais residem atrás da aplicação 1250 em uma rede de comunicação de dados privada. Em uma modalidade, o cliente 6205 faz um login e estabelece uma sessão de comunicação de dados com a aplicação 1250 usando a Camada de Soquetes Seguros (SSL), ou outros métodos de encriptação, tal como IPSec, e Segurança de Nível de Transporte (TLS). Em uma outra modalidade, um protocolo tal como Protocolo de Transferência de Hipertexto por Camada de Soquetes Segura (HTTPS) pode ser usado para iniciação da sessão de comunicação de dados encriptados.

[00744] Na etapa 6404, em resposta a um login e ao estabelecimento da sessão de comunicação de dados encriptados, a aplicação 1250 envia um primeiro programa para o cliente 6205 pela rede 6204. O primeiro programa é projetado e construído, ou configurado de outra forma, para atuar como um ponto final de túnel para comunicação pela sessão de comunicação de dados encriptados. Em uma modalidade, o primeiro programa compreende um aplicativo de plug-in que é automaticamente instalado e executado pelo navegador do cliente 6204. Por exemplo, o primeiro programa pode compreender um controle ActiveX que é provido como um plug-in a ser executado pelo navegador da web Microsoft Internet Explorer®. Em uma outra modalidade, o primeiro programa pode compreender um miniaplicativo em Java que é provido como um plug-in a ser executado por um navegador da web Netscape Navigator® ou um outro componente de controle ou de programação que trabalhe através de ambientes de rede.

[00745] Na etapa 6406, o cliente 6205 instala e executa o primeiro programa 6222, onde a execução do primeiro programa compreende a instalação de um segundo programa no cliente 6205. Em uma modalidade, o primeiro programa 6222 pode ser automaticamente instalado e executado, tal como usando qualquer uma das técnicas

discutidas em conjunto com o método 6300 e a figura 41A. Em algumas modalidades, o primeiro programa 6222 obtém, transfere (via download) ou recebe o segundo programa, ou o programa de aceleração 6120, a partir da aplicação 1250. Em uma outra modalidade, o primeiro programa 6222 compreende um instalador ou um gerenciador de instalação para o segundo programa, tal como o programa de aceleração 6120 para instalar e executar automaticamente o segundo programa, tal como por meio de uma instalação silenciosa ou uma instalação transparente para um usuário do cliente 6205, do aplicativo 6220 do cliente 6205, da aplicação 1250 ou do servidor 30.

[00746] Em uma modalidade, o segundo programa é configurado, em parte, para interceptar comunicações a partir de aplicativos 6220 rodando no cliente 6205 que são destinados para recursos na rede 6204 e para a provisão das comunicações interceptadas para o primeiro programa 6222 para envio para a aplicação 1250 através da sessão de comunicação de dados encriptados. O segundo programa também pode ser configurado para prover um serviço de resolução de nome de rede de intranet e opcionalmente dividir o tráfego de rede. Pela divisão do tráfego, uma modalidade é capaz de determinar que tráfego é canalizado para um túnel de SSL ou um túnel de encriptação do primeiro programa 6222 e que tráfego é permitido ou permite continuar adiante para processamento pela camada de transporte da pilha de rede 6210 sob operações normais, de rotina ou típicas do cliente 6205. Em uma modalidade, o segundo programa compreende um interceptador dinâmico (por exemplo, um driver de dispositivo de filtro) que é inserido como um "gancho" em um sistema operacional do cliente 6205. Por exemplo, o segundo programa pode compreender um driver de dispositivo de filtro que é afixado à pilha de camada de transporte do sistema operacional de cliente, tal como a pilha de

camada de transporte de um sistema operacional Microsoft Windows®.

[00747] Na etapa 6408, uma vez que os primeiro e segundo programas tenham sido instalados, os aplicativos rodando no cliente 6205 podem se comunicar com e acessar recursos, tais como aplicativos e dados, em uma rede de comunicação de dados privada 6204 através da sessão de comunicação de dados encriptados estabelecida. A maneira pela qual esta comunicação ocorre será discutida em maiores detalhes abaixo com respeito à figura 42B. Note que, em uma modalidade, as funções do primeiro programa e do segundo programa conforme descrito acima são realizadas por um único componente de controle ou de programação que é automaticamente instalado e executado pelo cliente 6205, tal como o programa de aceleração 6120. Além disso, para a provisão de uma conexão de rede privada virtual e comunicações, o primeiro programa 6222 e/ou o segundo programa, tal como o programa de aceleração 6120, pode realizar qualquer uma das técnicas de aceleração descritas aqui em comunicações do cliente através da conexão de rede privada virtual, por exemplo, o túnel encriptado ou ponte para a aplicação 1250.

[00748] Na etapa 6410, o cliente 6205 realiza um procedimento de logout para desconexão da rede 6204, o que termina a sessão de comunicação de dados encriptados com a aplicação 1250. Em uma modalidade, no momento de feitura do logout, o primeiro programa 6222 automaticamente limpa as modificações feitas no sistema operacional do cliente 6205 para se retornar o sistema operacional para um estado antes da instalação do primeiro programa 6222 e/ou do segundo programa. Em uma modalidade, o primeiro programa 6222 e/ou o segundo programa também incluem um desinstalador ou instruções de desinstalação para remoção dos primeiro e segundo

programas do sistema operacional do cliente 6205 ou a partir de uma operação adicional no cliente 6205 de uma maneira não intrusiva para as operações continuadas do cliente 6205. Em ainda uma outra modalidade, o primeiro programa 6222 e/ou o programa de aceleração 6120 remove quaisquer arquivos, tais como arquivos temporários ou cookies, usados por aplicativos do cliente 6205 durante quaisquer conexões de comunicação ou sessões providas.

[00749] A figura 42B descreve uma modalidade de um outro método 6450 por meio do qual um cliente 6205 se comunica com e acessa recursos em uma rede de comunicação de dados privados 6204. Por exemplo, o método 6450 representa um método pelo qual a etapa 6408 de método 6400 pode ser realizada. Em uma visão geral breve, na etapa 6452, o cliente 6205 faz uma nova conexão ou resolve um nome de domínio, tal como uma resolução de nome de domínio de TCP/IP, através do primeiro programa e/ou do segundo programa. Na etapa 6454, o segundo programa é executado. Na etapa 6456, o segundo programa intercepta comunicações a partir do cliente 6205 destinadas à rede privada e redireciona ou envia as comunicações para o primeiro programa 6222. Na etapa 6458, o primeiro programa 6222 termina ou delega a conexão, separa a carga útil e encapsula a carga útil para entrega através da sessão de comunicação encriptada estabelecida. Na etapa 6460, o primeiro programa 6222 envia comunicações interceptadas pela rede pública para a aplicação 1250 na rede privada através da sessão de comunicação encriptada preestabelecida. Na etapa 6462, a aplicação 1250 descripta as comunicações recebidas a partir do primeiro programa e encaminha as comunicações descriptadas para o recurso de destino apropriado, tal como um servidor 30. Na etapa 6464, o recurso de destino processa as comunicações descriptadas, e na etapa 6464 o recurso de destino envia uma comunicação de resposta, se houver, para a

aplicação 1250. Na etapa 6468, a aplicação 1250 encripta as comunicações de resposta e envia as comunicações encriptadas pela rede pública para o primeiro programa 6222 de cliente 6205 através da sessão de comunicação encriptada pré-estabelecida. Na etapa 6470, o primeiro programa 6222 desencripta comunicações de resposta e encaminha as comunicações desencriptadas para o aplicativo de cliente apropriado através do segundo programa.

[00750] Na etapa 6452, um aplicativo 6220 de um cliente 6205 faz uma nova conexão ou resolve um nome de domínio através da camada de protocolo de transporte da pilha de rede 6210 do cliente 6205. Em uma modalidade, o aplicativo 6220 pode requisitar o estabelecimento de uma conexão de camada de transporte entre o cliente 6205 e um servidor 30, ou entre o cliente 6205 e a aplicação 1250. Em uma outra modalidade, o aplicativo 6220 ou o cliente 6205 pode requisitar acesso a um aplicativo 6220 provido pelo servidor 30. Por exemplo, o servidor 30 pode prover uma computação baseada em servidor ou uma computação de cliente leve pela transmissão de um protocolo de exibição remota de ICA ou RDP representando uma saída de um aplicativo 6220 em execução no servidor 30. Em uma outra modalidade, o cliente 6205 pode requisitar acesso a recursos de um servidor 30, tais como arquivos ou diretórios, ou serviços de e-mail. Em algumas modalidades, o cliente 6205 pode estar em uma rede pública 40 e o servidor 30 em uma rede privada 40'. Em outras modalidades, o cliente 6205 e o servidor 30 podem estar em redes privadas diferentes.

[00751] Na etapa 6454, o segundo programa executa uma ou mais funções automaticamente ou de outra forma, antes de quaisquer funções de camada de transporte serem iniciadas. Em algumas modalidades, o segundo programa é ou compreende de outra forma o programa de aceleração 6120. Em uma modalidade, o segundo

programa intercepta ou recebe de outra forma a requisição de cliente da etapa 6452. Em algumas modalidades, o aplicativo 6220 do cliente 6205 faz chamadas de API para a pilha de rede 6210, as quais são interceptadas pelo segundo programa. Antes de quaisquer chamadas de PAI serem processadas pela camada de transporte da pilha de rede 6210, o segundo programa é enganchado em ou interfaceado de outra forma com a pilha de rede 6210 para execução de lógica, regras, funções ou operações, antes de a comunicação ser transmitida ou processada para transmissão através de uma conexão de camada de transporte.

[00752] Na etapa 6456, o segundo programa intercepta comunicações a partir do cliente 205, tal como por qualquer aplicativo 6220a a 6220n no cliente 6205 que são destinadas para recursos em uma rede 40' e as redireciona para o primeiro programa 6222, o qual em uma modalidade compreende um plug-in de controle ActiveX, um miniaplicativo em Java ou outro componente de controle ou de programação que funciona através de ambientes de rede. O segundo programa pode acessar, ler ou obter de outra forma uma informação de destino a partir do pacote ou dos pacotes de rede provendo as comunicações interceptadas para a determinação que a comunicação está destinada a uma rede, tal como uma rede privada 40' atrás da aplicação 1250. Por exemplo, o segundo programa pode extrair ou interpretar o endereço de IP de destino e/ou a porta a partir do pacote de rede. Mediante a determinação que uma comunicação interceptada é destinada à rede 40', o segundo programa comunica a comunicação interceptada para o primeiro programa 6222 através de quaisquer meios e mecanismo de interface adequados, tal como através de qualquer interface de comunicação interprocesso ou uma chamada de API. Em uma modalidade, a comunicação interceptada é enviada para o primeiro programa 6222 como está, ou, em outras modalidades, a

comunicação interceptada é pré-processada pelo segundo programa, antes do envio para o primeiro programa 6222. Por exemplo, o segundo programa pode remover a carga útil da comunicação interceptada e encaminhar a carga útil para o primeiro programa 6222.

[00753] Na etapa 6458, cada comunicação interceptada é terminada ou delegada pelo primeiro programa 6222, e o primeiro programa 6222 prepara a comunicação interceptada para transmissão através da sessão de comunicação de dados encriptados estabelecida. Em uma modalidade, o primeiro programa 6222 separa a carga útil e encapsula a carga útil para entrega através da sessão de comunicação de dados encriptados. Em uma outra modalidade, o primeiro programa 6222 encapsula a comunicação interceptada conforme recebida pelo segundo programa. Em algumas modalidades, a carga útil é uma carga útil de TCP e é encapsulada em uma nova conexão de TCP entre o cliente 6205 e o servidor 30, tal como através da aplicação 1250.

[00754] Na etapa 6460, o primeiro programa 6222 envia as comunicações interceptadas pela rede 6204 para a aplicação 1250 através da sessão de comunicação de dados encriptados préestabelecida. Em algumas modalidades, o primeiro programa 6222 encripta as comunicações interceptadas e envia as comunicações interceptadas encriptadas para a aplicação 1250. Em uma modalidade, uma encriptação é realizada de acordo com protocolos de SSL. Em uma outra modalidade, uma encriptação é baseada em TLS. Qualquer tipo e forma de encriptação e/ou de desencriptação podem ser usados pelo primeiro programa 6222 ou pelo programa de aceleração 6120.

[00755] Na etapa 6462, a aplicação 1250 atua como um próximo terminando a conexão enviada pelo primeiro programa 6222. A aplicação 1250 desencripta as comunicações recebidas a partir do primeiro programa 6222, e encaminha as comunicações

desencriptadas para o recurso de destino apropriado na rede 40' através de uma segunda conexão que a aplicação 1250 estabeleceu com o recurso de destino na rede 40'. Em uma modalidade, uma desencriptação é realizada de acordo com protocolos de SSL ou outros protocolos aplicáveis de encriptação e desencriptação. Em algumas modalidades, a aplicação 1250 realiza uma ou mais técnicas de aceleração na comunicação encaminhada para o recurso de destino, tal como uma ou mais das seguintes: técnicas providas por 1) compressão de protocolo múltiplo 6238', 2) agrupamento de protocolo de controle de transporte 6224', 3) multiplexação de protocolo de controle de transporte 6226', 4) armazenamento em buffer de protocolo de controle de transporte 6228', e 5) armazenamento em cache através de um gerenciador de cache 6232'.

[00756] Na etapa 6464, o recurso de destino processa as comunicações desencriptadas. Em uma modalidade, as comunicações desencriptadas são uma requisição para o estabelecimento de uma conexão ou sessão de comunicação. Em uma outra modalidade, as comunicações desencriptadas são uma requisição para se começar ou acessar um aplicativo 6220 em nome do cliente 6205. Em outras modalidades, as comunicações desencriptadas são uma requisição por uma página da web, tal como uma requisição de HTTP para receber uma página da web a partir de um servidor da web 30.

[00757] Na etapa 6466, se as comunicações desencriptadas incluírem uma requisição para a qual há uma resposta, então, o recurso de destino enviará comunicações de resposta para a aplicação 1250. Em algumas modalidades, a resposta inclui um reconhecimento de estabelecimento de uma conexão ou sessão de comunicação conforme requisitado pelo cliente 6205. Em outras modalidades, a resposta inclui uma mensagem de erro. Em uma modalidade, a resposta inclui uma requisição de autenticação ou um mecanismo de

resposta a desafio. Em outras modalidades, a resposta inclui um programa de aceleração 6120 a ser usado pelo cliente 6205. Em uma outra modalidade, a resposta inclui HTML, tal como uma página da web a ser exibida pelo cliente 6205. Em outras modalidades, a resposta inclui um objeto, tal como um objeto gerado dinamicamente.

[00758] Na etapa 6468, a aplicação 1250 envia as comunicações de resposta pela rede 40 para o primeiro programa 6220 no cliente 6205 através da sessão de comunicação de dados encriptados pré-estabelecida. Em uma modalidade, a aplicação 1250 encripta as comunicações de resposta e envia as comunicações de resposta encriptadas para o primeiro programa 6222. Em algumas modalidades, a encriptação é realizada de acordo com protocolos de SSL ou outros protocolos aplicáveis de encriptação e desencriptação. Mais ainda, a aplicação 1250 pode realizar qualquer uma das técnicas de aceleração em comunicações para o cliente 6205, tal como compressão de protocolo múltiplo 6238', armazenamento em cache 6232' ou armazenamento em buffer de TCP 6228'.

[00759] Na etapa 6470, o primeiro programa 6222 desencripta as comunicações de resposta e encaminha a comunicação para o aplicativo apropriado 6222 através do segundo programa. O primeiro programa 6222 pode usar qualquer meio de interface adequado ou mecanismo para comunicação para o segundo programa, tal como através de qualquer tipo e forma de mecanismo de comunicação interprocesso ou uma chamada de API. O segundo programa provê a comunicação de resposta através da pilha de rede 6210 do cliente 6205 para o aplicativo 6220. Como tal, o aplicativo 6220 recebe de forma transparente a comunicação de resposta sem quaisquer mudanças ou modificação para o aplicativo 6220.

[00760] De acordo com uma outra modalidade, o cliente 6205 realiza um processamento adicional das comunicações interceptadas

antes do envio de comunicações pela rede 40 na etapa 6458. Devido ao fato de uma modalidade prover uma solução de VPN que atua como um proxy terminando conexões no cliente, antes da encriptação desses dados, o processamento adicional pode ser realizado mais efetivamente. Esse processamento pode incluir uma resolução de nome de Serviço de Nome de Domínio (DNS) das comunicações interceptadas de modo a se permitir que aplicativos de cliente usem quaisquer endereços de IP que eles escolherem, bem como mudem dinamicamente aqueles endereços no tempo de execução. Esse processamento adicional permite que modalidades sejam integradas efetivamente com outras tecnologias, tal como equilíbrio de carga de serviço global para a obtenção de maior disponibilidade e maior eficiência dentre gateways ou servidores distribuídos. O processamento de conexão adicional também pode permitir a manutenção de registros detalhados e estatísticas referentes às comunicações interceptadas.

[00761] Em uma outra modalidade, uma aplicação 1250 termina as comunicações recebidas a partir do primeiro programa no cliente 6205 e processa ainda uma ou mais requisições incluídas ali, ao invés de encaminhar as comunicações para um destino na rede 40', conforme mostrado na etapa 6462. Este processamento adicional pode incluir uma encriptação de back-end em que comunicações são reencriptadas pela aplicação 1250, antes da entrega para o destino apropriado na rede 40', desse modo provendo uma segurança de rede de extremidade a extremidade. O destino desencriptará, depois disso, o tráfego e responderá apropriadamente. Ainda, esse processamento pode permitir que a aplicação 1250 apresente respostas a partir de um cache, ao invés de requerer um trabalho adicional por um servidor de destino, realizar um equilíbrio de carga de rede local, um equilíbrio de carga de serviço global e/ou compressão nas comunicações para

melhoria da eficiência e capacidade de resposta da rede 40.

[00762] De acordo com os métodos descritos acima, uma VPN com base em uma sessão de comunicação de dados encriptados é estabelecida entre um cliente 205 e a rede 40. Por exemplo, em uma modalidade, uma VPN segura é estabelecida através de HTTPS. Após isso, todas as comunicações a partir do cliente 6205 para a rede 40 são roteadas através do primeiro programa para a aplicação 1250 e vice-versa, através desta sessão de comunicação de dados encriptados. Deve ser notado que, embora a sessão de comunicação de dados encriptados possa ser estabelecida usando-se HTTPS, as comunicações que são passadas através da sessão de comunicação de dados encriptados não precisam ser dados de pacote de HTTPS ou mesmo dados de pacote de HTTP. Por exemplo, as comunicações também podem compreender um Protocolo de Controle de Transmissão/Protocolo de Datagrama de Usuário (TCP/UDP) ou dados de pacote de protocolo de Mensagem de Controle de Internet (ICMP), embora não se pretenda que estes exemplos sejam limitantes. Mais ainda, embora o método descrito com referência à figura 42B descreve uma comunicação do tipo de requisição – resposta entre um aplicativo no cliente 6205 e um recurso na rede 40, as comunicações encriptadas não precisam ser baseadas em requisição – resposta. Ao invés disso, as comunicações podem ser de qualquer tipo. Assim, qualquer aplicativo de cliente que possa estabelecer uma conexão ou sessão de comunicação, tal como uma sessão de UDP, pode enviar e receber comunicações encriptadas.

[00763] Em um outro aspecto, o programa de aceleração 6120 pode desviar dinamicamente do cliente qualquer dispositivo intermediário para se conectar ou comunicar com um servidor 30. Por exemplo, um cliente 6205 pode ter uma conexão com um servidor através de um ou mais intermediários, tal como a aplicação 1250. Por

uma razão ou outro, um intermediário pode não estar mais disponível para uso pelo cliente 6205 para comunicação com o servidor 30, por exemplo, a aplicação 1250 pode estar parada para manutenção ou pode estar no processo de reboot ou reinício. O programa de aceleração 6120 determina que o intermediário não está mais disponível e automaticamente estabelece um percurso de conexão ou sessão de comunicação diferente com o servidor 30. Isto pode ocorrer de forma transparente para o usuário ou aplicativo do cliente 6205, de modo que a conexão e/ou sessão de comunicação não pareça ter mudado ou de outra forma ter sido interrompida.

[00764] Com referência, agora, à figura 43, uma modalidade de um método 6500 para o desvio automático de um intermediário é descrito. Em uma visão geral breve, na etapa 6505, o programa de aceleração 6120 estabelece uma conexão de camada de transporte entre o cliente 6205 e o servidor 30 através de um intermediário, tal como a aplicação 1250. Na etapa 6510, o programa de aceleração 6120 determina que o intermediário não é usável para comunicação pelo cliente 6205 para o servidor 30 através da conexão de camada de transporte estabelecida. Na etapa 6515, o programa de aceleração 6120 intercepta no cliente 6205 uma comunicação a partir do cliente 6205 para o servidor 30. Na etapa 6520, o programa de aceleração 6120 estabelece uma segunda conexão de camada de transporte entre o cliente 6205 e o servidor 30, e, como resultado, desvia-se do intermediário determinado como não usável para as comunicações do cliente para o servidor 30. Na etapa 6525, o programa de aceleração 6120 transmite a comunicação interceptada do cliente 6205 para o servidor 30 através da segunda conexão de camada de transporte.

[00765] Em maiores detalhes, na etapa 6505, o programa de aceleração 6120 estabelece uma conexão de camada de transporte entre o cliente 6205 e o servidor 30 através de um intermediário. Em

uma modalidade, o intermediário compreende uma aplicação 6205. Em outras modalidades, o intermediário compreende um ou mais dos seguintes: um cache, um servidor, um gateway, um firewall, uma ponte, um roteador, um comutador, um centro de conexão, um próxi ou qualquer aplicativo ou programa de software atuando como ou provendo a funcionalidade e as operações de qualquer um destes tipos e formas de intermediários. Em uma modalidade, o intermediário pode operar no servidor 30. Em algumas modalidades, a conexão de camada de transporte é estabelecida através de uma pluralidade de intermediários do mesmo tipo e forma ou de tipos e formas diferentes. Em uma outra modalidade, a conexão de camada de transporte compreende a conexão de um grupo de conexão de camada de transporte estabelecida como o cliente 6205 ou a aplicação 1250.

[00766] Na etapa 6510, o programa de aceleração 6120 determina que o intermediário não está disponível ou de outra forma não é usável para comunicação pelo cliente 6205 para o servidor 30 através da conexão de camada de transporte estabelecida. O programa de aceleração 6120 pode determinar o status ou a disponibilidade do intermediário por quaisquer meios adequados e/ou mecanismo. Em uma modalidade, o programa de aceleração 6120 determina que o intermediário não está disponível pelo recebimento de uma mensagem de erro ou uma resposta de falha associada a uma transmissão para o intermediário. Por exemplo, o programa de aceleração 6120 pode receber uma resposta de comunicação de camada de transporte com falha quando transmitindo uma comunicação a partir do cliente 6205 através da conexão de camada de transporte estabelecida. Em uma outra modalidade, o programa de aceleração 6120 pode transmitir um comando de ping para o intermediário em uma frequência predeterminada para monitoração do status e da disponibilidade do intermediário. Se o programa de aceleração 6120 não receber uma

resposta do intermediário ou, em algumas modalidades, receber uma resposta atrasada ou uma resposta com uma latência mais longa do que o desejado, o programa de aceleração 6120 poderá determinar que o intermediário não está disponível ou é usável pelo cliente 6205. Em outras modalidades, um servidor 30, uma aplicação 1250 ou o intermediário pode enviar uma mensagem para o cliente 6205 ou o programa de aceleração 6120 provendo uma informação identificando que o intermediário não está disponível ou não é usável de outra forma pelo cliente 6205. Em algumas modalidades, a conexão de camada de transporte estabelecida é perturbada ou interrompida ou, em outras modalidades, é fechada.

[00767] Na etapa 6515, o programa de aceleração 6120 intercepta uma comunicação a partir do cliente 6205 para o servidor 30 destinada através do intermediário através da conexão de camada de transporte estabelecida. O programa de aceleração 6120 pode interceptar a comunicação em qualquer ponto e em qualquer camada de potência de transmissão na pilha de rede 6210. Em uma modalidade, o programa de aceleração 6120 intercepta a comunicação na camada de protocolo de transporte antes da transmissão na conexão de camada de transporte estabelecida. Por exemplo, em algumas modalidades, o programa de aceleração 6120 compreende um driver de rede que tem uma interface de driver de transporte ou de outra forma é interfaceado para a camada de protocolo de transporte. Outras modalidades podem incluir um primeiro programa 6222 e o programa de aceleração 6120 como um segundo programa, conforme discutido em conjunto com as figuras 42A a 42B, em que o primeiro programa 6222 ou o programa de aceleração 6120 intercepta a comunicação.

[00768] Na etapa 6520, o programa de aceleração 6120 estabelece uma segunda conexão de camada de transporte para o servidor 6205

para o cliente 6205, de modo a desviar do intermediário determinado como estando indisponível ou não usável pelo cliente na etapa 6510. Em uma modalidade, o programa de aceleração 6120 estabelece uma segunda conexão de camada de transporte diretamente para o servidor 30, por exemplo, quando o cliente 6205 e o servidor estão na mesma rede 6205 ou em redes diferentes roteáveis entre o cliente 6205 e o servidor 30. Em uma outra modalidade, o programa de aceleração 6120 estabelece a segunda conexão de camada de transporte com um segundo intermediário, tal como uma segunda aplicação 1250'. Em algumas modalidades, o programa de aceleração 6120 requisita que a aplicação 1250 estabeleça uma outra conexão de camada de transporte com o servidor 30. Em uma modalidade, a aplicação 1250 usa uma segunda conexão de camada de transporte de um grupo de conexões de camada de transporte para o servidor 30. Em uma outra modalidade, o programa de aceleração 6120 requisita que o servidor 30 estabeleça a segunda conexão de camada de transporte. Em algumas modalidades, o programa de aceleração 6120 usa uma segunda conexão de camada de transporte de um grupo de conexões de camada de transporte estabelecidas pelo programa de aceleração 6120 com o servidor 30.

[00769] Em uma modalidade, o programa de aceleração 6120 estabelece a segunda conexão de camada de transporte na etapa 6520 de forma transparente para um usuário ou aplicativo 6220 do cliente 6205 ou, em algumas modalidades, de forma transparente para qualquer camada de protocolo acima ou abaixo da camada de transporte. Em alguns aspectos, a segunda conexão de camada de transporte é estabelecida automaticamente para o cliente 6205 mediante uma determinação na etapa 6510 que o intermediário não está disponível ou não deve ser usado pelo cliente 6205. Em outras modalidades, a segunda conexão de camada de transporte é

estabelecida automaticamente mediante uma falha de transmissão da comunicação interceptada para o servidor 30, por exemplo, a primeira tentativa de transmissão da comunicação. Em algumas modalidades, a segunda conexão de camada de transporte é estabelecida automaticamente mediante uma falha de uma ou mais transmissões tentadas de novo da comunicação, ou mediante uma exaustão de um número predeterminado de novas tentativas. Em uma outra modalidade, a segunda conexão de camada de transporte é estabelecida mediante uma determinação que o intermediário está atrasando a taxa de transmissão ou recepção de pacotes de rede, causando uma latência ou afetando de outra forma o uso da conexão de camada de transporte de uma maneira indesejada. Em uma modalidade, o programa de aceleração 6120 realiza um equilíbrio de carga e estabelece uma segunda conexão de camada de transporte desviando do intermediário para descarregar qualquer processamento ou operações do intermediário para o cliente 6205 e/ou um segundo intermediário.

[00770] Na etapa 6525, o programa de aceleração 6120 transmite a comunicação interceptada do cliente 6205 para o servidor 30 através da segunda conexão de camada de transporte. Em uma modalidade, o programa de aceleração 6120 transmite a comunicação interceptada diretamente para o servidor 30. Em outras modalidades, o programa de aceleração 6120 transmite a comunicação interceptada através de um segundo intermediário, tal como uma segunda aplicação 1250. Pelo uso da segunda conexão de camada de transporte, o programa de aceleração 6120 desvia do intermediário e continua as operações de um aplicativo 6220 do cliente 6205 com o servidor 30. Em uma modalidade, um aplicativo 6220 do cliente 6205 continua com as operações e comunicações com o servidor 30 como se o aplicativo 6220 estivesse continuando a usar a conexão de camada de

transporte prévia ou primeiramente estabelecida. Como tal, o programa de aceleração 6120 impede, evita ou contorna qualquer interrupção de comunicação, perturbação, latências, atrasos ou outras questões operacionais ou de performance que podem ocorrer se o intermediário não for desviado pelo programa de aceleração 6120. Em um outro aspecto, esta técnica automaticamente provê ao cliente 6205 acesso contínuo a um servidor 30 ou aplicativo acessado remotamente, mesmo se houver um problema ou uma perturbação no acesso a partir de um dispositivo intermediário.

[00771] Mais ainda, as técnicas de redirecionamento e desvio descritas acima podem ser usadas para a realização de um equilíbrio de carga e um gerenciamento de tráfego no cliente 6205 para acesso a um ou mais servidores 30 provendo os aplicativos 6220a a 6220n, ou outro conteúdo e uma funcionalidade para o cliente 6205. Por exemplo, em uma modalidade, um intermediário ou uma aplicação usada pelo cliente para acesso a um servidor pode estar sobrecarregado com conexões de camada de transporte crescentes e diminuindo uma taxa de respostas, performance ou outras operações. Mediante uma determinação de uma performance decrescente do intermediário ou da aplicação, o programa de aceleração 6120 pode redirecionar o cliente para um outro intermediário ou aplicação, ou servidor, para se desviar de quaisquer estrangulamentos de performance na conectividade de extremidade a extremidade de cliente para o servidor.

[00772] Em outros aspectos, técnicas de aceleração de lado de cliente podem estar relacionadas a ou ser realizadas na camada de protocolo de transporte da pilha de rede do cliente. O programa de aceleração 6120 pode compreender instruções executáveis para a realização de qualquer um de um ou mais dentre 1) armazenamento em buffer de protocolo de controle de transporte (TCP) 6228, 2)

agrupamento de conexão de TCP 6224 e 3) multiplexação TCP 6226. Em algumas modalidades, como o programa de aceleração 6120 processa de forma transparente as comunicações interceptadas na camada de protocolo de transporte da pilha de rede de cliente, o programa de aceleração 6120 pode controlar e gerenciar as conexões de TCP do cliente, e o uso e a transmissão pelas conexões pelos aplicativos 6220a a 6220n do cliente 6205. A figura 44 descreve uma modalidade de método 6600 de prática de técnicas de armazenamento em buffer de TCP, enquanto as figuras 45A a 45B descrevem uma modalidade da técnica de agrupamento de conexão de TCP e as figuras 46, 47 e 48 a técnica de multiplexação de TCP.

[00773] Em uma visão geral breve de uma modalidade de método 6600 descrita na figura 44, na etapa 6605, o programa de aceleração 6120 intercepta uma comunicação a partir do cliente 6205 para o servidor 30, tal como uma requisição para acesso ao servidor 30 pelo cliente 6205. Na etapa 610, o programa de aceleração 6120 determina se uma diferença entre uma taxa de consumo de respostas de servidor recebidas e uma taxa de produção de requisições transmitidas pelo cliente cai abaixo de um limite predeterminado. Se na etapa 6615 a diferença de taxas de produto e de consumo não cair abaixo do limite predeterminado, o programa de aceleração 6120 encaminhará a comunicação para o servidor 30 na etapa 6617. Se, na etapa 6615, a diferença de taxas estiver abaixo do limite predeterminado, então, na etapa 6620, o programa de aceleração 6120 armazenará a comunicação na memória do cliente 6205. Na etapa 6625, o programa de aceleração 6120 determina se a diferença nas taxas mudou acima do limite predeterminado e, se assim for, encaminha a comunicação armazenada para o servidor 30. Caso contrário, o programa de aceleração 6120 mantém a comunicação na memória do cliente 6205, até um ponto no tempo em que a diferença nas taxas muda na etapa

6625 para acima do limite predeterminado. Por exemplo, se o cliente 6205 estiver transmitindo requisições para o servidor 30 a uma taxa maior do que aquela pela qual o cliente 6205 pode consumir as respostas geradas, o programa de aceleração 6120 manterá uma transmissão adicional até um ponto futuro no tempo no qual a diferença nas taxas tiver mudado.

[00774] Em maiores detalhes, na etapa 6605, o programa de aceleração intercepta uma comunicação a partir do cliente 6205 para o servidor 30. O programa de aceleração 6120 pode interceptar a comunicação em qualquer ponto e em qualquer camada de protocolo na pilha de rede 6210. Em uma modalidade, o programa de aceleração 6120 intercepta a comunicação na camada de protocolo de transporte antes da transmissão na conexão de camada de transporte estabelecida. Por exemplo, em algumas modalidades, o programa de aceleração 6120 compreende um driver de rede que tem uma interface de driver de transporte ou é interfaceado de outra forma com a camada de protocolo de transporte. Outras modalidades podem incluir um primeiro programa 6222 e o programa de aceleração 6120 como um segundo programa, conforme discutido em conjunto com as figuras 42A a 42B, nas quais o primeiro programa 6222 ou o programa de aceleração 6120 intercepta a comunicação. Em uma modalidade, a comunicação compreende uma requisição pelo cliente 6205 para usar ou acessar de outra forma um recurso do servidor 30, tal como um aplicativo 6220.

[00775] Na etapa 6610, o programa de aceleração 6120 determina se uma diferença entre uma taxa de consumo e uma taxa de produção do cliente 6205 cai abaixo de um limite predeterminado. Em uma modalidade, o programa de aceleração 6120 conta e acompanha o número de requisições transmitidas pelo cliente 6205 para o servidor 30, e, em uma outra modalidade, o programa de aceleração 6120

conta e acompanha o número de respostas recebidas pelo cliente 6205 a partir do servidor 30. Em algumas modalidades, o cliente 6205 acompanha as respostas transmitidas e as requisições recebidas em uma base de por aplicativo 6220. As respostas e as requisições podem ser acompanhadas em qualquer camada de protocolo da pilha de rede 6210. Em uma modalidade, o número de requisições transmitidas pelo cliente 6205 ou pelo aplicativo 6220 é contado e acompanhado a partir do ponto de submissão até a camada de transporte ou até uma conexão de camada de transporte entre o cliente 6205 e o servidor 30. Da mesma forma, em uma outra modalidade, o número de respostas recebidas pelo cliente 6205 ou pelo aplicativo 6220 a partir do servidor 30 é contado e acompanhado a partir do ponto de recepção até a camada de transporte ou a partir da conexão de camada de transporte entre o cliente 6205 e o servidor 30 e/ou no ponto em que a resposta é provida para uma camada de protocolo, tal como uma camada de aplicativo, acima da camada de transporte da pilha de rede 6210.

[00776] Em algumas modalidades, o programa de aceleração 6120 acessa inspeciona ou obtém de outra forma informação e dados sobre o envio e a recepção de buffers de TCP da conexão de camada de transporte estabelecida pelo programa de aceleração 6120 entre o cliente 6205 e o servidor 30. Por exemplo, o programa de aceleração 6120 pode determinar o tamanho default e o máximo de qualquer buffer de TCP/IP e as porções atualmente usadas do buffer para a determinação de uma diferença nas taxas entre envio e recepção de pacotes de rede a partir do cliente 6205 para o servidor 30. Em outras modalidades, o programa de aceleração 6120 usa qualquer tipo e forma de algoritmo de congestionamento para determinar se há um congestionamento causado por uma diferença de consumo e produto de pacotes de rede a partir do cliente 6205 para o servidor 30. Em

uma outra modalidade, o programa de aceleração 6120 tem uma interface com ou obtém uma informação ou dados a partir de um algoritmo de congestionamento usado pela conexão de camada de transporte, tal como por um driver de rede ou um provedor de serviços de TCP. Por exemplo, em uma modalidade, o programa de aceleração 6120 determina uma informação e dados referentes à janela de congestionamento usada pela conexão.

[00777] O limite predeterminado pode ser configurado, especificado, definido ou identificado por qualquer meio adequado e mecanismo do programa de aceleração 6120. Em uma modalidade, o limite pode ser especificado como uma percentagem, relativa, absoluta ou de outra forma, entre a taxa de produção e a taxa de consumo do cliente 6205 e/ou do aplicativo 6220. As taxas para consumo e/ou produto podem ser identificadas por um número de recepções consumidas e transmissões produzidas, respectivamente, ao longo de qualquer período de tempo em qualquer granularidade. Em algumas modalidades, o limite pode ser especificado como uma diferença de quantidade entre a taxa de produção e o consumo do cliente 6205 e/ou do aplicativo 6220, e, em algumas modalidades, uma diferença de quantidade por um período de tempo. Por exemplo, o limite pode ser especificado como o ponto no tempo em que o cliente 6205 produziu 6100 requisições a mais do que o cliente 6205 consumiu. Em um outro exemplo, o limite pode ser especificado como o ponto no tempo em que o cliente 6205 está produzindo 610 requisições por período de tempo para o servidor 30 a mais do que as requisições consumidas pelo cliente 6205 durante o mesmo período de tempo.

[00778] Na etapa 6615, se a diferença de taxa de produto e consumo do cliente 6205 e/ou aplicativo 6220 estiver abaixo do limite predeterminado, o programa de aceleração 6120 encaminhará a comunicação para o servidor 6260 na etapa 6617. Em algumas

modalidades, o programa de aceleração realiza quaisquer técnicas de aceleração para a comunicação. Por exemplo, a comunicação pode ser encaminhada para o servidor através de uma conexão de camada de transporte multiplexada agrupada e, adicionalmente, pode ser comprimida. Em outras modalidades, o cliente 6205 pode encaminhar a comunicação para uma aplicação 1250 provendo uma conexão para o cliente 6205 para o servidor 30.

[00779] Na etapa 6615, se a diferença de taxa de produto e de consumo do cliente 6205 e/ou do aplicativo 6220 estiver abaixo do limite predeterminado, o programa de aceleração 6120, na etapa 6620, armazenará a comunicação na memória do cliente 6205. Em algumas modalidades, a memória pode ser uma memória do modo de kernel 6202 do cliente 6205. Em uma modalidade, o programa de aceleração 6120 pode armazenar a comunicação no cache através do gerenciador de cache 6232. Em outras modalidades, o programa de aceleração 6120 pode usar um objeto, uma estrutura de dados ou um outro elemento de dados acessível pelo programa de aceleração 6120 para armazenar em buffer, manter ou armazenar de outra forma a comunicação interceptada. Em uma modalidade, a comunicação interceptada pode ser armazenada de uma maneira comprimida na memória. Em uma outra modalidade, o programa de aceleração 6120 envia a comunicação interceptada para um primeiro programa 6222 para armazenamento ou manutenção na memória para transmissão em um ponto posterior no tempo.

[00780] Na etapa 6625, o programa de aceleração 6120 determina quando transmitir a comunicação armazenada para o servidor 30. Em uma modalidade, o programa de aceleração 6120 realiza as etapas 6610 e 6615 para determinar se a diferença de taxas de produção e de consumo do cliente 6205 está acima do limite, mediante o que o programa de aceleração 6120 encaminha a comunicação armazenada

para o servidor 30 na etapa 6617. Em algumas modalidades, o programa de aceleração 6120 compara a diferença de taxas de produção e de consumo em uma frequência regular ou predeterminada ou em uma base de interrogação ou de evento, e quando a diferença subir acima do limite predeterminado, o programa de aceleração 6120 encaminha a comunicação para o servidor 30. Em outras modalidades, o programa de aceleração 6120 regula ou configura um temporizador para determinar por quanto tempo armazenar a comunicação interceptada. Mediante a expiração do temporizador, o programa de aceleração 6120 transmite a comunicação armazenada para o servidor 30. Em uma outra modalidade, o programa de aceleração 6120 checa o número de respostas de servidor consumidas pelo cliente 6205 desde o armazenamento da comunicação interceptada. Se o número de respostas consumidas for maior do que um número predeterminado, o programa de aceleração 6120 liberará a comunicação interceptada do buffer de memória ou do armazenamento e submeterá a comunicação para transmissão para o servidor 30.

[00781] Se, na etapa 6625, o programa de aceleração 6120 determinar que as taxas de produção e de consumo não mudaram de uma maneira adequada, o programa de aceleração 6120 segurar ou manterá a comunicação interceptada na memória até um ponto de tempo adequado ser atingido. Em uma modalidade, o programa de aceleração 6120 encaminha a comunicação para o servidor na etapa 6617, mesmo se as taxas de produção e/ou de consumo não mudarem. Por exemplo, após um período de tempo esperando que a taxa de produção e/ou de consumo mude e as taxas não mudarem, o programa de aceleração 6120 encaminha a comunicação para o servidor 30.

[00782] Embora a técnica de armazenamento em buffer de TCP

seja geralmente discutida em relação a uma comunicação ou requisição interceptada, as modalidades do método 6600 podem ser praticadas seqüência, de forma quase simultânea ou concorrentemente para múltiplas comunicações interceptadas do cliente 6205 para o servidor 30. Adicionalmente, em uma outra modalidade, o método 6600 pode ser praticado no cliente com respeito a comunicações a partir do cliente para múltiplos servidores 30. Por exemplo, uma primeira instância de método 6600 pode ser praticada entre o cliente 6205 e um primeiro servidor 30', e uma segunda instância de método 6600 pode ser praticada entre o cliente 6205 e um segundo servidor 30''. Mais ainda, em algumas modalidades, o método 6600 pode ser praticado por um primeiro aplicativo 6200a e também por um segundo aplicativo 6200b, usando as respectivas taxas de produção e de consumo de cada aplicativo. Em outras modalidades, o método 6600 pode ser praticado para um primeiro aplicativo 6200a, mas não por um segundo aplicativo 6200b.

[00783] De acordo com um outro aspecto, o programa de aceleração de lado de cliente 6120 reduz a carga de processamento de servidores 30 e/ou aplicação 1250 causada pela abertura e pelo fechamento repetidamente de conexões dos clientes pela abertura de uma ou mais conexões com cada servidor e mantendo-se estas conexões para se permitirem acessos repetidos a dados pelos aplicativos do cliente 6205 ao servidor 30. Esta técnica geralmente é referida aqui como um "agrupamento de conexão". Com referência, agora, à figura 45A, em uma visão geral breve do método 6700, na etapa 6702, o programa de aceleração 6120 intercepta uma requisição de aplicativo para acesso a um servidor e, na etapa 6704, determina a identidade do servidor associado à requisição. Na etapa 6706, o programa de aceleração 6120 determina se o programa de aceleração 6120 tem uma conexão de camada de transporte estabelecida com o

servidor 30 livre para uso pelo aplicativo 6220. Se não houver uma conexão de camada de transporte para o servidor 30 livre para uso pelo aplicativo 6220, o programa de aceleração 6220 estabelecerá, na etapa 6708, uma conexão de camada de transporte com o servidor 30 para uso pelo cliente 6205. Na etapa 6706, se houver uma conexão de camada de transporte disponível para uso pelo aplicativo 6220, na etapa 6710, o programa de aceleração 6120 traduzirá a requisição de aplicativo para transmissão ou comunicação através da conexão de camada de transporte disponível.

[00784] Em uma visão geral adicional, na etapa 6712, o programa de aceleração 6120 recebe a resposta à requisição a partir do servidor 30 e, na etapa 6714, traduz a resposta em uma resposta para o aplicativo 6220. Na etapa 6716, o programa de aceleração 6120 pode manter ou deixar a conexão de camada de transporte aberta para uso por qualquer um dos aplicativos 6220a a 6220n do cliente 6205. Pela manutenção no cliente 6205 de conexões de camada de transporte abertas com os servidores 30 e pela abertura e pelo fechamento de conexões com os aplicativos, conforme necessário, o programa de aceleração 6120 libera os servidores de problemas de carregamento de conexão de TCP associados ao serviço do cliente 6205 pela rede 40, tal como a Internet. Na etapa 6718, o programa de aceleração 6120 em algum ponto fecha a conexão de camada de transporte, se a conexão for determinada como não sendo mais usada por um ou mais aplicativos 6220 do cliente 6205 para acesso ao servidor 30.

[00785] Em maiores detalhes, na etapa 6702, o programa de aceleração 6120 intercepta uma requisição por qualquer aplicativo 6220a a 6220n do cliente 6205 para acesso a um servidor 30. Em algumas modalidades, a requisição é interceptada na camada de protocolo de transporte antes do estabelecimento ou da transmissão da requisição através de uma conexão de camada de transporte. Em

outras modalidades, a requisição é interceptada em qualquer camada de protocolo acima da camada de transporte ou de uma conexão de camada de transporte. Em uma modalidade, a requisição do aplicativo 6220 é uma requisição para abertura ou estabelecimento de uma conexão de camada de transporte com o servidor 30. Em algumas modalidades, em resposta à requisição, o programa de aceleração 6120 estabelece uma primeira conexão de camada de transporte de um grupo de conexões de camada de transporte para uso pelos aplicativos 6220a a 6220n do cliente 6205. Em uma outra modalidade, a requisição de aquecimento é uma requisição para acesso ao servidor através de uma conexão de camada de transporte estabelecida do cliente 6205.

[00786] Na etapa 6704, o programa de aceleração 6120 determina a identidade do servidor 30 a partir da requisição por qualquer meio adequado e mecanismo. Em algumas modalidades, o nome de domínio ou o endereço de protocolo de internet do servidor 30 é identificado ou referenciado de outra forma pelo conteúdo da requisição, por exemplo, uma string de texto da requisição pode identificar o nome de domínio de um servidor 30. Em uma modalidade, a identidade do servidor 30 é determinada pela informação de cabeçalho de um pacote de TCP, tais como o endereço de protocolo de internet de destino e o número de porta. Em uma outra modalidade, o servidor 30 é associado ao aplicativo 6220, e o programa de aceleração 6120 procura ou consulta a associação em um banco de dados ou um outro armazenamento de informação estruturado.

[00787] Na etapa 6706, o programa de aceleração 6120 determina se há uma conexão de camada de transporte disponível para uso ou que esteja livre de outra forma para uso pelo aplicativo 6220. Em uma modalidade, o programa de aceleração 6120 pode ainda não ter

estabelecido uma conexão de camada de transporte com o servidor 30, e, como tal, não há uma conexão de camada de transporte disponível para o aplicativo 6220 usar. Em uma outra modalidade, o programa de aceleração 6120 pode ter uma conexão de camada de transporte previamente estabelecida com o servidor 30, mas determina que um outro aplicativo 6220 atualmente está ativamente usando a conexão. Conforme será discutido em maiores detalhes abaixo, o programa de aceleração 6120 determina se uma conexão de camada de transporte estabelecida está disponível para uso por um outro aplicativo ou pode ser compartilhada pelos aplicativos 6220a a 6220n, com base no comprimento de uma mensagem sendo recebida a partir do servidor 30 para o aplicativo 6220, tal como uma resposta a uma requisição e/ou se as comunicações entre o servidor 30 e o aplicativo 6220 estiverem atualmente inativas.

[00788] Na etapa 6708, se o programa de aceleração 6120 determinar que uma conexão de camada de transporte não está disponível para uso pelo aplicativo 6220, o programa de aceleração 6120 estabelecerá uma conexão de camada de transporte com o servidor 30. Em algumas modalidades, a conexão de camada de transporte estabelecida na etapa 6708 é a primeira conexão de camada de transporte com o servidor 30 e, em outras modalidades, a conexão de camada de transporte é uma segunda conexão de camada de transporte de uma pluralidade de conexões de camada de transporte com o servidor 30. Em ainda uma outra modalidade, o programa de aceleração 6120 espera que uma camada de transporte já estabelecida se torne disponível ou livre para comunicação da requisição de aplicativo para o servidor 30. Por exemplo, o programa de aceleração 6120 pode determinar que um primeiro aplicativo 6220a pode estar brevemente completando uma transação com o servidor 30 através de uma conexão estabelecida.

[00789] Na etapa 6710, o programa de aceleração 6120 traduz a requisição de aplicativo a ser transmitida através da conexão de camada de transporte para o servidor 6106. Em algumas modalidades, o programa de aceleração 6120 usa um número de porta para a comunicação de conexão de camada de transporte para todos os aplicativos 6220a a 6220n do cliente 6205 compartilhando a conexão. Em alguns casos, o programa de aceleração 6120 acompanha as requisições e as respostas destacadas para requisições em uma base de aplicativo por aplicativo. Como tal, o programa de aceleração 6120 reconhece qual aplicativo 6220 está transmitindo e recebendo pacotes de rede através da conexão de camada de transporte para o servidor 30 em qualquer dado ponto no tempo. Em uma modalidade, apenas um aplicativo 6220 em um tempo está enviando e recebendo na conexão de camada de transporte e, assim, o programa de aceleração 6220 entende qual aplicativo 6220 está usando a conexão. Em algumas modalidades, o programa de aceleração 6120 associa um id de processo do aplicativo 6220 à requisição. Em outras modalidades, o programa de aceleração 6120 provê e associa um número de porta ao aplicativo 6220, e modifica o número de porta no pacote de rede de TCP a ser transmitido para o número de porta atribuído de aplicativo. Em uma outra modalidade, o número de porta é provido pelo aplicativo 6220 e o programa de aceleração 6120 muda ou provê de outra forma o número de porta de modo conforme no pacote de rede de TCP.

[00790] Na etapa 6712, o programa de aceleração 6120 recebe uma resposta para a requisição do aplicativo a partir do servidor 30. Em uma modalidade, o servidor 30 não responde à requisição. Em uma outra modalidade, o servidor 30 responde com um erro ou uma mensagem de falha. Em algumas modalidades, o servidor 30 responde com múltiplas respostas. Em outras modalidades, o servidor 30 responde com uma resposta compreendendo múltiplos pacotes de

rede ou múltiplos segmentos de TCP. Em uma outra modalidade, o servidor 30 responde com um ou mais pacotes de rede identificando o número de porta de fonte associado ao ou atribuído ao aplicativo 6220. Em uma modalidade, o servidor 30 responde com um ou mais pacotes de rede identificando um número de porta de fonte da conexão de camada de transporte e usado para múltiplos aplicativos do cliente 6205.

[00791] Na etapa 6714, o programa de aceleração 6120 traduz ou processa de outra forma a resposta a partir do servidor 30 de uma maneira em resposta ao aplicativo 6220. Em uma modalidade, o programa de aceleração 6120 substitui o número de porta de fonte do pacote ou dos pacotes de rede recebidos pelo número de porta do aplicativo 6220. Em uma outra modalidade, o programa de aceleração 6120 determina através de um mecanismo de acompanhamento o aplicativo 6220 atualmente usando a conexão de camada de transporte e passa a resposta para o aplicativo 6220 através da pilha de rede 6210. Em uma modalidade, a resposta não é alterada e passada para processamento através das camadas de protocolo da pilha de rede 6210 acima da camada de transporte da conexão. Em algumas modalidades, o programa de aceleração 6120 espera que múltiplas porções, tais como segmentos de TCP, da resposta sejam recebidas, antes do processamento e do encaminhamento da resposta para o aplicativo 6220. Em uma modalidade, o programa de aceleração 6120 passa a resposta para um primeiro programa 6222, o qual tem uma interface com e provê a resposta para o aplicativo 6220.

[00792] Na etapa 6716, o programa de aceleração 6120 mantém ou deixa a conexão de camada de transporte aberta em um grupo de uma ou mais conexões de camada de transporte a partir do cliente 6205 para o servidor 30. Em uma modalidade, o programa de aceleração 6120 ou um driver de camada de transporte da pilha de rede 6210

inclui um mecanismo de manter vigente que periodicamente sonda a outra extremidade de uma conexão, quando a conexão estiver de outra forma inativa, por exemplo, quando não há dados a enviar. O mecanismo de manter vigente pode enviar esta mensagem, de modo a receber uma resposta para confirmar que a conexão ainda está ativa, embora a conexão possa estar inativa. A mensagem de manter vigente e a resposta correspondente podem incluir qualquer tipo e forma de formato, comando, diretiva ou comunicação. Como tal, em algumas modalidades, o programa de aceleração 6120 transmite ou faz com se transmita através de um driver de camada de transporte uma mensagem de manter vigente para a conexão de camada de transporte. Em algumas modalidades, o programa de aceleração 6120 regula uma frequência para mensagens de manter vigente, e, em outras modalidades, muda a frequência de mensagens de manter vigente com base no compartimento ou na atividade dos aplicativos 6220a a 6220n usando a conexão.

[00793] Em algumas modalidades, o programa de aceleração 6120 intercepta quaisquer comandos de RST e/ou FIN, isto é, comandos de TCP/IP para reinicialização e/ou terminação da conexão de TCP, recebidos pela conexão de camada de transporte. Em uma modalidade, o programa de aceleração 6120 ignora, não faz nenhuma ação, ou de outra forma abandona, apaga ou limpa o comando de RST e/ou FIN interceptado. Em uma outra modalidade, o programa de aceleração 6120 intercepta e recebe os comandos de RST e/ou FIN, mas envia uma mensagem para a outra extremidade da conexão para deixar ou manter a conexão aberta. Em outras modalidades, o programa de aceleração 6120 estabelece uma nova conexão de camada de transporte, em resposta a um fechamento de uma conexão de camada de transporte estabelecida devido ao processamento de um comando de RST e/ou FIN.

[00794] Em outras modalidades, o programa de aceleração 6120 insere uma instrução, um comando ou uma diretiva em uma comunicação interceptada do cliente 6205 para direcionamento do servidor 30 para manter a conexão aberta ou de outra forma não fechar a conexão, a menos que o cliente 6205 envie um comando para fazê-lo. Por exemplo, em uma modalidade, o programa de aceleração 6120 intercepta uma comunicação de uma requisição GET do protocolo HTTP, tal como a versão de protocolo 1.0, e insere um cabeçalho de manter vigente, por exemplo, "Connection: Keep-Alive" na comunicação com o servidor 30. Em outras modalidades, uma requisição GET ou outro comando de HTTP pode incluir o cabeçalho de manter vigente. Nestas modalidades, o programa de aceleração 6120 pode interceptar a comunicação e checar quanto ao cabeçalho de manter vigente e, então, encaminhar a comunicação para o servidor 30. Em algumas modalidades, uma versão 1.1 ou posterior de HTTP é usada, por meio da qual o mecanismo de manter vigente é implícito, de modo que o servidor 30 mantenha a conexão aberta até o cliente 6205 requisitar o fechamento da conexão. Em outras modalidades, o programa de aceleração 6120 mantém a conexão de camada de transporte aberta com o servidor 30 até o cliente 6205 ser rebutado ou reiniciado, a rede 40 se tornar indisponível ou o cliente 6205 ser desconectado da rede 40, ou o servidor 30 ser rebutado ou reiniciado.

[00795] Na etapa 6718, o programa de aceleração 6120 pode fechar qualquer uma ou mais das conexões de camada de transporte entre um cliente 6205 e um servidor 30 em qualquer ponto desejado no tempo. Em algumas modalidades, o programa de aceleração 6120 fecha uma conexão de camada de transporte mediante a terminação de um ou mais aplicativos 6220a a 6220n no cliente 6205 usando a conexão. Em outras modalidades, o programa de aceleração 6120

fecha uma conexão de camada de transporte mediante uma expiração de um período de expiração para qualquer aplicativo 6220a a 6220n usar a conexão. Por exemplo, o programa de aceleração 6120 pode configurar, regular ou prover um temporizador para expirar quando de um período de tempo predeterminado e, se a conexão estiver ou permanecer inativa durante o período de tempo. O programa de aceleração 6120 fechará a conexão. Em algumas modalidades, o servidor 30 pode ser rebutado, reiniciado ou a conexão desfeita ou interrompida e o programa de aceleração 6120 fecha a conexão. Em algumas modalidades, o programa de aceleração 6120 transmite ou faz com que seja transmitido um comando de RST e/ou FIN para fechamento da conexão mediante a conclusão do envio de requisições para e do recebimento de todos os dados de respostas a partir do servidor 30. Em outras modalidades, a conexão de camada de transporte ou o grupo de conexões de camada de transporte é fechado mediante um reinício ou um reboot do cliente 6205, uma desconexão da rede 40 ou uma indisponibilidade da rede 40, ou um reinício ou um reboot do servidor 30.

[00796] Em algumas modalidades, uma primeira conexão de camada de transporte com o servidor 30 é mantida aberta, enquanto uma segunda conexão de camada de transporte com o servidor é fechada, conforme o programa de aceleração 6120 determinar que apenas a primeira conexão de camada de transporte é necessária para compartimento de uma conexão com o servidor 30 por um ou mais aplicativos 6220a a 6220n do cliente 6205. Em outras modalidades, o programa de aceleração 6120 mantém um grupo de uma conexão de camada de transporte com qualquer servidor 30 e estabelece uma segunda ou uma pluralidade de conexões para um dado servidor 30, com base em requisições aumentadas, comunicações ou uso de conexão de camada de transporte dos

aplicativos 6220a a 6220n no cliente 6205.

[00797] Embora uma modalidade de método 6700 seja geralmente discutida em relação a um grupo de uma ou mais conexões de camada de transporte a partir do cliente 6205 com um servidor 30, o programa de aceleração 6120 pode estabelecer subsequente, quase de forma simultânea ou concorrentemente um grupo de conexões de camada de transporte entre o cliente e cada um de uma pluralidade de servidores 30. Como tal, um primeiro aplicativo 6220a e um segundo aplicativo 6220b pode usar um primeiro grupo de uma ou mais conexões de camada de transporte para o servidor 30a, e um terceiro aplicativo 6220c e um quarto aplicativo 6220d usando um segundo grupo de uma ou mais conexões de camada de transporte com o servidor 30b. Mais ainda, cada uma das etapas de uma modalidade do método 6700 pode ser realizada em instâncias diferentes e em frequências diferentes. Em algumas modalidades, múltiplas instâncias do programa de aceleração 6120 podem ser usadas para se lidar com cada grupo de uma ou mais conexões de camada de transporte para cada servidor 30.

[00798] Agora, com referência à figura 45B, um fluxograma é descrito de um programa de aceleração 6120 provendo uma conexão de camada de transporte para uso por dois aplicativos 6220a e 6220b de um cliente 6205, para um servidor 30 em uma modalidade, ou para uma aplicação 1250, em uma outra modalidade. O programa de aceleração 6120 no cliente 6205 abre uma primeira conexão de camada de transporte entre o cliente 6205 e o servidor 30, ou a aplicação 1250, usando um endereço de rede 1 provido pelo aplicativo 6220, conforme descrito pela etapa 6752. A etapa 6752 é mostrada como uma etapa de duas vias, porque o protocolo de TCP/IP emprega um cumprimento de estágio múltiplo para abertura de conexões.

[00799] Uma vez que a conexão de camada de transporte seja

estabelecida, o programa de aceleração 6120 intercepta uma requisição GET a partir do aplicativo 6220a especificando um nome de caminho de /sales/forecast.html, conforme mostrado pela etapa 6754. Devido ao fato de nenhuma conexão de camada de transporte livre estar aberta entre o programa de aceleração 6120 e o servidor 30, ou a aplicação 6205, o programa de aceleração 6120 abre uma conexão de camada de transporte. Em uma modalidade, o programa de aceleração 6120 mapeia a requisição do aplicativo 6220a para um segundo endereço de rede de endereço de rede 2, o qual especifica o servidor 30, conforme mostrado pela etapa 6756. Por exemplo, o programa de aceleração 6120 realiza uma tradução de endereço de rede para modificação do endereço de IP de destino e/ou da porta de destino para um servidor 30' requisitado pelo aplicativo 6220a ou um outro servidor 30'' que também pode lidar com ou responder a requisição. Em uma outra modalidade, o programa de aceleração 6120 envia a requisição para o servidor 30 ou a aplicação 1250, conforme recebida ou conforme gerada pelo aplicativo 6220s.

[00800] O programa de aceleração 6120 também passa a requisição GET para aquele servidor 30 ou a aplicação 1250, conforme mostrado pela etapa 6758. Em uma modalidade, a aplicação 1250 encaminha a requisição para o servidor 30 e, em uma modalidade adicional, a aplicação 1250 encaminha a requisição através de conexões de camada de transporte agrupadas ou agrupadas e multiplexadas entre a aplicação 1250 e o servidor 30. Em algumas modalidades, o servidor 30 responde com a página da web requisitada, conforme mostrado pela etapa 6760. O programa de aceleração 6120 encaminha a página da web para o aplicativo 6220a, conforme mostrado pela etapa 6762. Em uma modalidade, a conexão de camada de transporte entre o programa de aceleração 6120 e o servidor 30 ou a aplicação 1250 está fechada, conforme mostrado pela

etapa 6764. Em outras modalidades, o programa de aceleração 6120 intercepta a requisição de fechar e ignora a requisição, deixando a conexão de camada de transporte aberta. De acordo com o protocolo de TCP/IP, o fechamento de uma conexão de rede pode envolver um processo de estágio múltiplo. Portanto, a linha de fluxo da etapa 6764 é mostrada como bidirecional. Em outras modalidades e de acordo com as técnicas do aspecto de agrupamento, a conexão de camada de transporte estabelecida e usada pelo primeiro aplicativo 6220 é mantida aberta ou mantida de outra forma para a acomodação de etapas de dados adicionais para o mesmo aplicativo 6220a ou um aplicativo diferente, tal como o segundo aplicativo 6220b.

[00801] Na etapa 6766, o programa de aceleração 6120 intercepta uma requisição a partir do segundo aplicativo 6220a para o servidor 30, ou aplicação 1250. Se houver uma conexão de camada de transporte livre aberta e/ou usável pelo segundo aplicativo 6220b, tal como a conexão de camada de transporte estabelecida na etapa 6756 para o primeiro aplicativo 6220a, o programa de aceleração 6120 usa esta conexão de camada de transporte previamente estabelecida. Como tal, uma segunda conexão de camada de transporte não precisa ser aberta na etapa 6766. Caso contrário, o programa de aceleração 6120 estabelece uma segunda conexão de camada de transporte com o servidor 30 ou a aplicação 1250. Na etapa 6768, o programa de aceleração intercepta uma requisição a partir do segundo aplicativo 6220b, por exemplo, requisitando a página da web /sales/forecast.html, e transmite a requisição para o servidor 30 ou a aplicação 1250, na etapa 6770. Devido ao fato de uma conexão livre já estar aberta entre o programa de aceleração 6120 e o servidor 6120, é desnecessário que o programa de aceleração 6120 sobrecarregue o servidor 6120 com a carga de processamento de abertura de uma conexão adicional. Na etapa 6772, o programa de aceleração 6120

intercepta ou recebe uma resposta a partir do servidor 30, tal como através da aplicação 1250 a partir da conexão de camada de transporte, e encaminha a resposta para o segundo aplicativo 6220b. Na etapa 6776, o programa de aceleração 6120 intercepta uma requisição de fechar a partir do segundo aplicativo 6220b, e, em algumas modalidades, fecha a conexão, enquanto em outras modalidades ignora a requisição, e mantém a conexão para a acomodação de outras requisições de dados do primeiro aplicativo 6220a, do segundo aplicativo 6220b ou ainda de um outro aplicativo 6220c a 6220n do cliente 6205.

[00802] Há vários cenários que resultam no programa de aceleração 6120 fechando a conexão com o servidor 30 ou a aplicação 1250, na etapa 6776. Por exemplo, o cliente 6205 ou o programa de aceleração 6120 pode iniciar um comando FIN (terminar) mediante uma determinação que o cliente 6205 recuperou todos os dados requisitados para os aplicativos 6220a e 6220b, ou mediante uma terminação, um fechamento ou uma saída dos aplicativos 6220a e 6220b. Em algumas modalidades, o cliente 6205 ou o programa de aceleração 6120 também pode iniciar um comando RST (reinicializar) sob condições similares. Além do fechamento da conexão entre o programa de aceleração 6120 e o servidor 30 ou a aplicação 1250, o comando RST resulta em várias operações de preparo sendo realizadas para se manter a conexão de lado de servidor em boa ordem. Em particular, o protocolo de TCP garante que o comando RST tenha o número de SEQ (seqüência) correto, de modo que o servidor aceite o segmento. Contudo, não é garantido que o comando RST tenha o número de ACK (reconhecimento) correto. Para se lidar com este cenário, o programa de aceleração 6120 mantém um acompanhamento dos bytes de dados enviados pelo servidor 30 ou pela aplicação 1250, e os bytes reconhecidos pelo cliente 6205. Se o

cliente 6205 ainda não tiver reconhecido todos os dados pelo servidor 30, a programa de aceleração 6120 calcula os bytes não reconhecidos, e envia um ACK para o servidor 6205.

[00803] Mais ainda, embora não mostrado na figura 45B, o servidor 30 ou a aplicação 1250 também pode fechar uma conexão entre si mesmo e o cliente 6205. O servidor 30 ou a aplicação 1250 enviaria um comando FIN para o cliente 6205. Em resposta, em algumas modalidades, o programa de aceleração 6120 fecha a conexão e, em uma modalidade adicional, restabelece uma outra conexão com o servidor 30 ou a aplicação 1250.

[00804] Mais ainda, embora uma modalidade de método 6700 da figura 45A e o fluxograma de exemplo da figura 45B sejam geralmente discutidos como um agrupamento de uma ou mais conexões de camada de transporte para uso por uma pluralidade de aplicativos, a técnicas de agrupamento pode ser aplicada a um aplicativo único 6220 que requisiite ou inicie uma pluralidade de conexões de camada de transporte e requisições através destas conexões. Por exemplo, em uma modalidade de protocolo HTTP, uma conexão de camada de transporte pode ser estabelecida ou cada requisição de HTTP a partir de um aplicativo. Usando-se as técnicas, um grupo de uma ou mais conexões de camada de transporte pode ser usado pelo aplicativo 6220, sem abertura e fechamento de conexões de camada de transporte para cada requisição.

[00805] Em um outro aspecto, técnicas para a multiplexação de requisições de aplicativo através da mesma conexão de camada de transporte ou de uma compartilhada podem ser usadas, tal como uma conexão de camada de transporte estabelecida através das técnicas de agrupamento descritas em conjunto com as figuras 45A a 45B. Em algumas modalidades, a disponibilidade de uma conexão de camada de transporte estabelecida é determinada e requisições podem ser

multiplexadas a partir de uma pluralidade de aplicativos através da conexão ao se checar se o conteúdo de uma resposta a partir do servidor 30 para requisições de um aplicativo foi recebido completamente. Conforme será discutido em maiores detalhes abaixo, em uma modalidade, o parâmetro de comprimento de conteúdo de uma resposta é usado e, em uma outra modalidade, um cabeçalho de codificação de transferência fragmentada de uma resposta é usado para se checar se todos os dados de uma resposta foram recebidos. Em um aspecto, se todos os dados de uma resposta foram recebidos é checado para se determinar se uma conexão agrupada atualmente está livre para uso por um aplicativo e/ou se é para estabelecer uma outra conexão de camada de transporte para o grupo de conexões com o servidor, tal como nas etapas 6706 e 6708 de método 6700 descritas na figura 45. Em uma outra modalidade, a técnica de checar o comprimento de conteúdo para uma resposta é usada como uma técnica para a multiplexação de requisições a partir de uma pluralidade de aplicativos através da mesma conexão de camada de transporte.

[00806] Com referência, agora, à figura 46, uma modalidade de um método 6800 para multiplexação de requisições através de uma conexão de camada de transporte única a partir do cliente 6205 com o servidor 30 é descrita. Em uma visão geral breve, na etapa 6805, o programa de aceleração 6120 estabelece uma conexão de camada de transporte entre o cliente 6205 e o servidor 30. Na etapa 6810, o programa de aceleração 6120 intercepta uma primeira requisição de um primeiro aplicativo 6220a para o servidor 30. Na etapa 6815, o programa de aceleração 6120 determina se a conexão de camada de transporte atualmente está sendo usada por um outro aplicativo ou de outra forma está inativa. Na etapa 6817, se a conexão de camada de transporte estiver disponível para uso pelo aplicativo 6220a, então, na etapa 6820, o programa de aceleração 6120 transmite a requisição

para o servidor. Caso contrário, na etapa 6817, se a conexão de camada de transporte não estiver disponível para uso pelo aplicativo 6220a, então, o programa de aceleração 6120 na etapa 6819 esperará por um período de tempo e retornará para a etapa 6815, ou estabelecerá uma segunda conexão de camada de transporte para uso pelo aplicativo 6220. Na etapa 6825, o programa de aceleração 6120 recebe uma resposta da requisição de aplicativo a partir do servidor. Na etapa 6830, o programa de aceleração 6120 intercepta uma segunda requisição, por um segundo aplicativo 6220b, e prossegue na etapa 6815 para determinar se a conexão de camada de transporte está disponível para uso pelo segundo aplicativo 6220b. Em algumas modalidades, o programa de aceleração 6120 intercepta a requisição do segundo aplicativo 6220b na etapa 6830, antes de receber a resposta da primeira requisição na etapa 6825, ou antes de receber todos os dados da resposta. Conforme discutido adicionalmente aqui, em algumas modalidades, o programa de aceleração 6120 usa a técnica de checagem de comprimento de conteúdo para determinar quando a conexão de camada de transporte está inativa ou um aplicativo recebeu todos os dados para uma resposta a uma requisição.

[00807] Em maiores detalhes, na etapa 6805, o programa de aceleração 6120 estabelece uma conexão de camada de transporte entre o cliente 6205 e o servidor 30. Em algumas modalidades, o programa de aceleração 6120 estabelece a conexão de camada de transporte com ou através da aplicação 1250 ou de um intermediário. Em uma modalidade, o programa de aceleração 6120 estabelece a conexão de camada de transporte como um grupo de conexões de camada de transporte para o servidor 30. Como tal, em algumas modalidades, a conexão de camada de transporte pode compreender uma segunda ou uma terceira conexão de camada de transporte com

o servidor 30. Em outras modalidades, o programa de aceleração 6120 pode estabelecer a conexão de camada de transporte através de um primeiro programa 6222, conforme discutido previamente aqui. Em algumas modalidades, o programa de aceleração 6120 estabeleceu a conexão de camada de transporte em resposta a uma requisição por um primeiro aplicativo 6220a do cliente 6205.

[00808] Na etapa 6810, o programa de aceleração 6120 intercepta uma primeira requisição por um primeiro aplicativo 6220a para acesso ao servidor 30. Em algumas modalidades, a requisição é interceptada na camada de protocolo de transporte antes do estabelecimento ou da transmissão da requisição através da conexão de camada de transporte. Em outras modalidades, a requisição é interceptada em qualquer camada de protocolo acima da camada de transporte ou acima da conexão de camada de transporte. Em algumas modalidades, a requisição é interceptada por um primeiro programa 6222. Em uma modalidade, a requisição do aplicativo 6220a é uma requisição para a abertura ou o estabelecimento de uma conexão de camada de transporte com o servidor 30. Em uma outra modalidade, a requisição de aplicativo é uma requisição para acesso ao servidor através da conexão de camada de transporte estabelecida ou através da aplicação 1250.

[00809] Na etapa 6815, o programa de aceleração 6120 determina se a conexão de camada de transporte está inativa ou disponível para uso pelo primeiro aplicativo 6220a ou comunica a primeira requisição do primeiro aplicativo 6220a. Em algumas modalidades, o programa de aceleração 6120 determina a partir de um grupo de uma ou mais conexões de camada de transporte, qual conexão de camada de transporte no grupo está inativa ou livre para uso pelo primeiro aplicativo 6220a. Em uma modalidade, o programa de aceleração 6120 determina que a conexão de camada de transporte está inativa

porque o programa de aceleração 6120 estabeleceu a conexão de camada de transporte em resposta à requisição, ou imediatamente antes da requisição. Em algumas modalidades, o programa de aceleração 6120 pode não ter recebido quaisquer requisições de qualquer aplicativo 6220 e reconhece esta requisição como a primeira requisição a ser interceptada e processada pelo programa de aceleração 6120. Em uma outra modalidade, o programa de aceleração 6120 acompanha o número de respostas destacadas para quaisquer requisições transmitidas na conexão de camada de transporte, e se não houver respostas destacadas, o programa de aceleração 6120 reconhecerá que a conexão de camada de transporte está disponível para uso pelo primeiro aplicativo 6220a. Ainda em uma outra modalidade, o programa de aceleração 6120 reconhece que a conexão de camada de transporte atualmente está inativa. Por exemplo, o programa de aceleração 6120 pode estar iniciando requisições de manter vigente para o servidor para manter a conexão aberta. Em algumas modalidades, a conexão de camada de transporte está inativa, já que a última transação foi completada, mas o servidor 30 e/ou o cliente 6205 ainda não transmitiu um comando RST e/ou FIN.

[00810] Em algumas modalidades, o programa de aceleração 6120 pode checar o comprimento de conteúdo de uma resposta para determinar se a resposta a partir do servidor 30 para a primeira requisição do primeiro aplicativo 6202a está completa ou de outra forma o programa de aceleração 6120 recebeu todos os dados para a resposta. Conforme mencionado acima, estas técnicas em algumas modalidades também podem ser usadas para a determinação do estabelecimento de uma outra conexão para a técnica de agrupamento. Com respeito a esta técnica, as figuras 47 e 48 serão usadas para a descrição da checagem do parâmetro de comprimento

de conteúdo de uma resposta em uma modalidade, ou em uma outra modalidade um cabeçalho de codificação de transferência fragmentada de uma resposta, para se determinar se todos os dados de uma resposta foram recebidos. A figura 47 descreve uma porção de TCP de um pacote de TCP referido como o segmento de TCP 6900. O segmento de TCP 6900 inclui um cabeçalho de TCP 6902, e um corpo 6904. O corpo 6904 compreende dentre outros dados e informação um cabeçalho de HTTP e uma mensagem em uma modalidade, em que o pacote de trocador de calor porta um protocolo de camada de aplicativo de HTTP. Em algumas modalidades, um parâmetro de comprimento de conteúdo 6906 está localizado, é encontrado ou referenciado por ou no cabeçalho de HTTP. Em uma modalidade, o programa de aceleração 6120 usa o parâmetro de comprimento de conteúdo 6906 para determinar se todos os dados para uma resposta são recebidos.

[00811] A figura 48 descreve uma outra modalidade de um segmento de TCP de um pacote de TCP. Em algumas modalidades de uso do protocolo HTTP pela conexão de camada de transporte, um cabeçalho de codificação de transferência fragmentada pode estar presente e indicando que uma codificação de transferência fragmentada foi aplicada ao segmento ou pacote de TCP. Como tal, nesta modalidade, o comprimento da mensagem é definido pela codificação fragmentada. A codificação fragmentada modifica o corpo da mensagem, de modo a transferir a mensagem como uma série de fragmentos, cada fragmento com seu próprio indicador de comprimento em um campo de tamanho de fragmento. O segmento de TCP 7600 inclui um cabeçalho de TCP (não mostrado) e um corpo. O corpo compreende, dentre uma outra informação, um cabeçalho de HTTP 7602A a 7602C e a mensagem. O cabeçalho de HTTP 7602A a 7602C compreende sete campos de tamanho de fragmento 7606A a

7601C, e seis dados de mensagem de fragmento 7604A a 7604F.

[00812] Os campos de tamanho de fragmento 7606A a 7606G são ligados em conjunto ou referenciados de outra forma ou associados, conforme ilustrado na figura 48. O campo de tamanho de fragmento 7606A indica o comprimento da mensagem nos dados de mensagem de fragmento 7604A, o campo de tamanho de fragmento 7606C indica o comprimento da mensagem nos dados de mensagem de fragmento 7604C e assim por diante. O último campo de tamanho de fragmento 7606G compreende o valor de comprimento zero indicando que não há mais fragmentos ou qualquer coisa a mais da mensagem a seguir. Em uma outra modalidade, o programa de aceleração 6120 determina através dos campos de tamanho de fragmento se o cliente 6205 recebeu todos os dados para uma resposta.

[00813] Embora as figuras 47 e 48 geralmente descrevam uma técnica para checar se todos os dados para uma resposta a uma requisição foram recebidos, estas técnicas são aplicáveis a um servidor 30 ou uma aplicação 1250 enviando uma mensagem assíncrona ou comunicação para o cliente 6205. Mais ainda, embora estas técnicas sejam geralmente descritas em conjunto com as figuras 47 e 48 para um protocolo HTTP, estas técnicas podem ser usadas para qualquer protocolo em qualquer camada de protocolo que proveja uma indicação do comprimento de dados a serem transmitidos ou recebidos pelo cliente 6205. Como tal, em algumas modalidades, o programa de aceleração 6120 acessa, extrai, inspeciona, analisa ou processa de outra forma qualquer porção do pacote de rede, incluindo qualquer camada de protocolo, para determinar se todos os dados já foram recebidos em associação com uma requisição, resposta ou comunicação entre o cliente e o servidor ou a aplicação. Ainda em uma outra modalidade, o programa de aceleração 6120 acompanha os números de bytes transmitidos, recebidos e reconhecidos entre o

cliente 6205 e o servidor 30 para determinar se quaisquer bytes estão se destacando entre o cliente 6205 e o servidor 30. Caso contrário, na etapa 6819, o programa de aceleração 6120 pode esperar até que todos os dados sejam recebidos para uma requisição destacada de um aplicativo. Por exemplo, o programa de aceleração 6120 pode regular um temporizador, por exemplo, para um período de tempo curto, e prosseguir para a etapa 6815. Em algumas modalidades, o programa de aceleração 6120 checa se todos os dados foram recebidos em resposta a um temporizador de processamento de pacote da pilha de rede 6210 do cliente 6205. Em outras modalidades, na etapa 6819, o programa de aceleração 6120 estabelece uma outra conexão de camada de transporte para a transmissão da primeira requisição do primeiro aplicativo 6220a.

[00814] Na etapa 6820, o programa de aceleração 6120 pode acompanhar qual aplicativo 6220 atualmente tem uma requisição destacada ou resposta na conexão ou atualmente está usando a conexão. Por exemplo, apenas um aplicativo 6220 em um tempo pode transmitir uma requisição e receber uma resposta na conexão. Como tal, o programa de aceleração 6120 entende qual aplicativo 6220 está usando a conexão. Em algumas modalidades, o programa de aceleração 6120 usa um número de porta para a comunicação de conexão de camada de transporte para todos os aplicativos 6220a a 6220n do cliente 6205 compartilhando a conexão. Em alguns casos, o programa de aceleração 6120 acompanha as requisições e respostas destacadas para as requisições em uma base de aplicativo por aplicativo. Em algumas modalidades, o programa de aceleração 6120 associa um id de processo do aplicativo 6220 à requisição. Ainda em uma outra modalidade, o programa de aceleração 6120 transmite a requisição do primeiro aplicativo 6220a com uma requisição do segundo aplicativo 6220b no mesmo pacote ou nos mesmos pacotes

de rede, segmento ou segmentos de TCP. Em outras modalidades, o programa de aceleração 6120 transmite uma pluralidade de requisições de aplicativos 6220a a 6220n através da mesma conexão de camada de transporte como parte de uma série de segmentos de TCP de uma ou mais janelas de segmento de TCP.

[00815] Em outras modalidades, o programa de aceleração 6120 usa um mecanismo de numeração de porta e/ou esquema para acompanhamento e reconhecimento de qual resposta ou mensagem recebida é para o aplicativo 6220a a 6220n. Em outras modalidades, o programa de aceleração 6120 provê e associa um número de porta ao aplicativo 6220, e modifica o número de porta no pacote de rede de TCP a ser transmitido para o número de porta atribuído de aplicativo. Em uma outra modalidade, o número de porta é provido pelo aplicativo 6220 e o programa de aceleração 6120 muda ou provê de outra forma o número de porta de modo conforme no pacote de rede de TCP. Como tal, em algumas modalidades, o programa de aceleração 6120 pode entrelaçar requisições de uma pluralidade de aplicativos 6220a a 6220n do cliente 6205, de modo que os aplicativos 6220a a 6220n possam usar a conexão de camada de transporte ao mesmo tempo.

[00816] Na etapa 6825, o programa de aceleração 6120 recebe uma resposta para a primeira requisição do primeiro aplicativo 6220a a partir do servidor 30, tal como através da aplicação 6205, e provê a resposta para o primeiro aplicativo 6220a. Em algumas modalidades, o programa de aceleração 6120 provê a resposta para o primeiro aplicativo 6220a através da pilha de rede 6210, tal como permitindo ou iniciando o processamento da resposta pelas camadas de protocolo acima da camada de transporte da conexão. Em uma outra modalidade, o primeiro programa 6222 provê a resposta para o primeiro aplicativo 6220a. Em outras modalidades, o programa de aceleração 6120 pode prover a resposta para o primeiro aplicativo

6220a através de um mecanismo de comunicação interprocesso ou uma interface, tal como uma API. Em algumas modalidades, o programa de aceleração 6120 apenas recebe uma porção da resposta, tal como um primeiro fragmento em uma mensagem de fragmento múltiplo, conforme descrito na figura 48.

[00817] Na etapa 6830, o programa de aceleração 6120 intercepta uma requisição de um segundo aplicativo 6220b para acesso ao servidor 30. Em algumas modalidades, o programa de aceleração 6120 intercepta a requisição do segundo aplicativo 6220b antes da etapa 6825. Em outras modalidades, o programa de aceleração 6120 intercepta a requisição do segundo aplicativo 6220b durante o recebimento da resposta na etapa 6825. Em uma outra modalidade, o programa de aceleração 6120 intercepta a requisição do segundo aplicativo 6220b antes do cliente 6205 ou do programa de aceleração 6120 receber todos os dados para uma resposta da primeira requisição do primeiro aplicativo 6220a. Mediante uma interceptação da requisição do segundo aplicativo 6220b, o programa de aceleração 6120 prossegue para a etapa 6815 em uma modalidade, para determinar se é para multiplexar a segunda requisição através da conexão de camada de transporte ou se é para estabelecer uma outra conexão de camada de transporte, tal como uma outra conexão em um grupo de conexões. Em outras modalidades, o programa de aceleração 6120 transmite a requisição do segundo aplicativo 6220b através da mesma conexão que o primeiro aplicativo 6220a, enquanto o primeiro aplicativo 6220a tem uma resposta destacada ou não recebeu todos os dados a partir da resposta da primeira requisição. Em uma outra modalidade, o programa de aceleração 6120 transmite a requisição do segundo aplicativo 6220b após o primeiro aplicativo 6220a ter recebido a resposta e antes de quaisquer comandos gerados RST e/ou FIN serem gerados em relação ao primeiro

aplicativo 6220a.

[00818] Embora o programa de aceleração 6120 tenha sido geralmente discutido em relação à implementação de lado de cliente e à execução de técnicas de aceleração, o programa de aceleração 6120 tem uma interface e trabalha em conjunto com a aplicação 1250, a qual também implementa e executa técnicas de aceleração de lado de aplicação. Em uma modalidade, o programa de aceleração de lado de cliente 6120 e a aplicação 1250 podem trabalhar em conjunto um com o outro para a realização de uma pluralidade de técnicas de aceleração em comunicações entre os clientes 6205 e os servidores 30. Em algumas modalidades, o programa de aceleração de lado de cliente 120 e a aplicação 1250 provêm agrupamento de TCP e multiplexação, tal como provendo um mecanismo de agrupamento e multiplexação em cascada ou de extremidade a extremidade entre os clientes 6205 e os servidores 30. Por exemplo, o programa de aceleração 6120 pode prover uma primeira conexão de camada de transporte agrupada para a aplicação 1250, a qual por sua vez provê uma segunda conexão de camada de transporte agrupada para o servidor 30. Em um outro exemplo, o programa de aceleração 6120 pode multiplexar uma requisição de aplicativo através de uma primeira conexão de camada de transporte agrupada no cliente 6205, a qual por sua vez é multiplexada pela aplicação 1250 através da segunda conexão de camada de transporte agrupada com o servidor 30. Em algumas modalidades, o programa de aceleração 120 provê um mecanismo de estrangulamento para transmissão de requisições a partir do cliente 6205, enquanto a aplicação 1250 provê um mecanismo de estrangulamento para transmissão de respostas a partir dos servidores 30 para os clientes 6205. Em uma outra modalidade, o programa de aceleração 6120 realiza um armazenamento em cache de lado de cliente para o cliente 6205, enquanto a aplicação 1250

provê um armazenamento em cache de objetos, tais como objetos gerados dinamicamente, para o cliente 6205 juntamente com outros clientes 6205.

[00819] Em algumas modalidades, além de ou em conjunto com a realização de técnicas de aceleração no cliente 6205 e/ou na aplicação, o programa de aceleração 6120 e a aplicação podem prover uma conexão de rede privada virtual e comunicações entre o cliente 6205 e um acesso de rede 40 através da aplicação 1250. Em uma outra modalidade, o programa de aceleração 6120 pode comprimir os dados comunicados a partir de um aplicativo 6220, e a aplicação 1250 pode descomprimir os dados comprimidos mediante o recebimento dos mesmos. Inversamente, a aplicação 1250 pode comprimir os dados comunicados a partir de um aplicativo 6220 no servidor 30 em uma rede de comunicação de dados privada 40' e o programa de aceleração 6120 pode descomprimir os dados comprimidos mediante o recebimento dos mesmos. Também, o programa de aceleração 6120 e a aplicação 1250 podem atuar como pontos finais em uma sessão de comunicação de dados encriptados ou de tunelamento, na qual o programa de aceleração 6120 encripta dados comunicados a partir de um aplicativo 6220, e a aplicação 1250 descripta os dados encriptados mediante o recebimento dos mesmos. De uma maneira similar, a aplicação 1250 encripta os dados comunicados a partir de um aplicativo 6220 na rede de comunicação de dados privada e o programa de aceleração 6120 pode descriptar os dados mediante o recebimento dos mesmos.

D. Exemplo de Aceleração de Entrega de um Ambiente de Computação

[00820] Tendo em vista a estrutura, as funções e as operações descritas acima nas Seções B e C, em algumas modalidades, a entrega de um ambiente de computação para um cliente pode ser

acelerada. Por exemplo, as modalidades descritas aqui podem ser usadas para a entrega de um aplicativo de transmissão e de um arquivo de dados processável pelo aplicativo a partir de um centro de dados corporativo central para uma localização de usuário remota, tal como uma filial da companhia. A aplicação e o programa de aceleração provêm técnicas de aceleração de extremidade a extremidade para a aceleração de qualquer carga útil de camada de transporte, tais como aplicativos transmitidos e arquivos de dados, de um servidor para um cliente remoto. O sistema de gerenciamento de entrega de aplicativo provê técnicas de entrega de aplicativo para a entrega de um ambiente de computação para uma área de trabalho de um usuário remoto com base em uma pluralidade de métodos de execução e com base em quaisquer políticas de autenticação e autorização aplicadas através de um agente de política. Com estas técnicas, um usuário remoto pode obter um ambiente de computação e acessar os aplicativos e arquivos de dados armazenados em servidor a partir de qualquer dispositivo conectado na rede.

[00821] Com referência, agora, à figura 49A, uma modalidade para a prática dos sistemas e métodos de aceleração e entrega de aplicativo descritos acima é descrita. Em uma visão geral breve, um cliente 10 está em comunicação com um servidor 30 através de uma rede 40, 40' e da aplicação 1250. Por exemplo, o cliente 10 pode residir em um centro de dados corporativo. O cliente 10 compreende um agente de cliente 560 e um ambiente de computação 15. O ambiente de computação 15 pode executar ou operar um aplicativo que acesse, processe ou use um arquivo de dados. O ambiente de computação 15, um aplicativo e/ou arquivo de dados podem ser entregues através da aplicação 1250 e/ou do servidor 30. Em algumas modalidades, o cliente 10 também inclui um programa de aceleração 4120, um agente de coleta 404 e um cliente de transmissão 562. O

servidor 30 inclui um sistema de entrega de aplicativo 500 e, em algumas modalidades, um agente de política 406.

[00822] Em uma modalidade, o sistema de entrega de aplicativo 500 pode residir ou ser executado em um servidor 30. Em uma outra modalidade, o sistema de entrega de aplicativo 500 pode residir ou estar em execução em uma pluralidade de servidores 30-30''. Em algumas modalidades, o sistema de entrega de aplicativo 500 pode ser executado em um banco de servidores. Em uma modalidade, o servidor 30 executando o sistema de entrega de aplicativo 500 também pode armazenar ou prover o aplicativo e o arquivo de dados. Em uma outra modalidade, um primeiro conjunto de um ou mais servidores 30 pode executar o sistema de entrega de aplicativo 500, e um servidor diferente 30' pode armazenar ou prover o aplicativo e o arquivo de dados. Em algumas modalidades, cada um dentre o sistema de entrega de aplicativo 500, o aplicativo e o arquivo de dados podem residir ou estar localizados em servidores diferentes. Em uma modalidade, o sistema de entrega de aplicativo 500 também inclui o agente de política 406. Em uma outra modalidade, o agente de política 406 se executa separadamente do sistema de entrega de aplicativo 500. Em algumas modalidades, o agente de política 406 está no mesmo servidor 30 que o sistema de entrega de aplicativo 500. Em outras modalidades, o agente de política 406 se executa na aplicação 1250. Em ainda uma outra modalidade, qualquer porção do sistema de entrega de aplicativo 500 e/ou do agente de política 406 pode residir, ser executada ou armazenada em ou distribuída para a aplicação 1250, ou para uma pluralidade de aparelhagens.

[00823] Em algumas modalidades, o agente de cliente 560 inclui qualquer um dentre o cliente de transmissão 562, o agente de coleta 404 e/ou o programa de aceleração 6120, conforme descrito previamente acima. Em uma modalidade, o agente de cliente 560, o

cliente de transmissão 562, o agente de coleta 404 e/ou o programa de aceleração 6120 formam ou são incorporados em um programa único ou um conjunto de instruções executáveis provendo a funcionalidade, a lógica e as operações de cada um. Em outras modalidades, cada um dentre o cliente de transmissão 562, o agente de coleta 404 e o programa de aceleração 6120 se executa separadamente do agente de cliente 560. Em uma modalidade, o cliente 10 executa o agente de cliente 560. Em uma outra modalidade, o cliente 10 executa o cliente de transmissão 562. Em algumas modalidades, o cliente 10 executa o agente de coleta 404. Em uma modalidade, o cliente 10 executa o programa de aceleração 6120. Em algumas modalidades, o cliente 10 executa o agente de cliente 560 com um ou mais dentre o cliente de transmissão 562, o agente de coleta 404 ou o programa de aceleração 6120. Em outras modalidades, o cliente 10 executa o cliente de transmissão 562 e o programa de aceleração 6120. Em uma modalidade, o cliente 10 executa o programa de aceleração 6120 e o agente de coleta 404.

[00824] Em algumas modalidades, o cliente 10 obtém o agente de cliente 560, o cliente de transmissão 562 e/ou o agente de coleta 404 a partir do servidor 30. Em outras modalidades, o cliente 10 obtém o agente de cliente 560, o cliente de transmissão 562 e/ou o agente de coleta 404 a partir da aplicação 1250. Em uma modalidade, qualquer um dentre o agente de cliente 560, o cliente de transmissão 562 e/ou o agente de coleta 404 pode ser armazenado na aplicação 1250. Por exemplo, em algumas modalidades, o agente de cliente 560, o cliente de transmissão 562 e/ou o agente de coleta 404 podem ser armazenados em cache na aplicação 1250. Em outras modalidades, mediante uma determinação pela aplicação 1250 que um aplicativo pode ser acelerado, a aplicação 1250 pode transmitir o agente de cliente 560, o cliente de transmissão 562, o programa de aceleração

6120 e/ou o agente de coleta 404 para o cliente 10. Em outras modalidades, o cliente 10 pode automaticamente instalar e executar qualquer um dentre o agente de cliente 560, o cliente de transmissão 562, o programa de aceleração 6120 e/ou o agente de coleta 404. Em ainda uma outra modalidade, qualquer um dentre o agente de cliente 560, o cliente de transmissão 562, o programa de aceleração 6120 e/ou o agente de coleta 404 pode se executar de forma transparente para um usuário ou aplicativo do cliente, ou para qualquer porção da pilha de rede do cliente.

[00825] Em algumas modalidades, a aplicação 1250 estabelece uma conexão de VPN ou de SSL VPN para o cliente 10 com o servidor 30 ou a rede 40'. Em outras modalidades, a aplicação 1250 atua como um proxy, um servidor de acesso ou um equilibrador de carga para a provisão de acesso para um ou mais servidores 30. Em uma modalidade, a aplicação 1250 e/ou o programa de aceleração 6120 aceleram a entrega do cliente de transmissão 562, do agente de coleta 404 e/ou do agente de cliente 560 para o cliente 10. Em uma modalidade, a aplicação 1250 acelera a entrega do programa de aceleração 6120 para o cliente 10. Em outras modalidades, a aplicação 1250 e/ou o programa de aceleração 6120 aceleram a entrega do ambiente de computação 15, de um aplicativo e/ou arquivo de dados para o cliente 10. Em uma modalidade, o cliente 10 tem um ambiente de computação 15 e a aplicação 1250 e/ou o programa de aceleração 6120 aceleram a entrega do aplicativo e/ou do arquivo de dados. Em uma modalidade, a aplicação 1250 e/ou o programa de aceleração 6120 aceleram a entrega do aplicativo. Em uma outra modalidade, a aplicação 1250 e/ou o programa de aceleração 6120 aceleram a entrega do arquivo de dados. Ainda em uma outra modalidade, a aplicação 1250 e/ou o programa de aceleração 6120 aceleram a entrega de um ambiente de computação 15, tal como um

ambiente de execução ou um ambiente de execução virtualizado previamente descrito aqui.

[00826] Em uma modalidade, a aplicação 1250 usa uma informação coletada a partir do agente de coleta 404 para determinar se um ambiente de computação 15, um aplicativo e/ou um arquivo de dados podem ser acelerados. Em algumas modalidades, o agente de política do aplicativo 1250 compreende o agente de política 406. Em outras modalidades, a aplicação 1250 se comunica ou tem uma interface com o agente de política 406 para determinar uma autenticação e/ou uma autorização de um usuário remoto ou de um cliente remoto 10 para acesos ao ambiente de computação 15, ao aplicativo e/ou ao arquivo de dados a partir de um servidor 30. Em uma outra modalidade, a aplicação 1250 se comunica ou tem uma interface com o agente de política 406 para determinar uma autenticação e/ou uma autorização de um usuário remoto ou de um cliente remoto 10 para ter o sistema de entrega de aplicativo 500 entregando um ou mais dentro o ambiente de computação 15, o aplicativo e/ou o arquivo de dados. Ainda em uma outra modalidade, a aplicação 1250 estabelece uma conexão de VPN ou de SSL VPN com base na autenticação e/ou na autorização do agente de coleta 404 de um usuário remoto ou de um cliente remoto 10. Em uma modalidade, a aplicação 1250 controla o fluxo de tráfego de rede e sessões de comunicação com base em políticas do agente de política 406. Por exemplo, a aplicação 1250 pode controlar o acesso a um ambiente de computação 15, um aplicativo ou arquivo de dados com base no agente de política 406.

[00827] Com referência, agora, à figura 49B, uma modalidade de um método para aceleração da entrega de um ambiente de computação para um usuário remoto de um cliente e uma localização remota é descrita. Em uma visão geral breve do método 8000, na etapa 8005, o servidor 30 recebe uma requisição para execução de

um aplicativo no cliente 10. Na etapa 8010, o servidor 30 transmite para o cliente 10 um aplicativo para execução. Na etapa 8015, a aplicação 1250 e/ou o programa de aceleração de lado de cliente 6120 aceleram a transmissão ou a entrega do aplicativo para o cliente 10. Na etapa 8020, o cliente 10 ou o aplicativo requisita um arquivo de dados a partir do servidor 30 para uso pelo aplicativo. Na etapa 8025, o servidor 30 e/ou a aplicação 1250 transmitem o arquivo de dados para o cliente 10. Na etapa 8030, a aplicação 1250 e/ou o programa de aceleração de lado de cliente 6120 aceleram a transmissão ou a entrega do arquivo de dados para o cliente 10.

[00828] Em maiores detalhes, na etapa 8005, um servidor 30 recebe uma requisição para execução de um aplicativo em um cliente 10. Em algumas modalidades, o usuário do cliente 10 faz a requisição. Em outras modalidades, um aplicativo, sistema operacional ou ambiente de computação 15 transmite a requisição. Em uma modalidade, a aplicação 1250 intercepta a requisição a partir do cliente 10 e encaminha a requisição para o servidor 30. Em uma modalidade, a aplicação 1250 encaminha a requisição para o servidor 30 com base na autenticação e/ou autorização do usuário ou cliente 10. Em uma outra modalidade, a aplicação 1250 encaminha a requisição para o servidor 30 com base em uma informação provida pelo agente de coleta 404. Em uma modalidade, a requisição inclui uma requisição para execução do aplicativo por um método de uma pluralidade de métodos de execução. Por exemplo, o usuário do cliente 10 pode requisitar a execução do aplicativo como um aplicativo transmitido a partir do servidor, como um aplicativo instalado e executado localmente, ou como um aplicativo baseado em servidor em execução no servidor 30 e se exibindo remotamente para o cliente 10. Em algumas modalidades, a requisição é com base em uma associação de tipo de arquivo. Por exemplo, um usuário pode selecionar um

arquivo associado a um aplicativo que é usado para leitura do ou acesso ao arquivo.

[00829] Na etapa 8010, em resposta à requisição da etapa 8005, o servidor 30 transmite o aplicativo para execução para o cliente 10. Em algumas modalidades, o servidor 30 transmite o aplicativo para o cliente 10. Por exemplo, pela transmissão do aplicativo em algumas modalidades, o aplicativo opera no cliente 10 sem uma instalação. Em outras modalidades, o servidor 30 transmite para o cliente 10 um aplicativo para instalação e execução locais. Por exemplo, usando as técnicas de instalação e execução automáticas descritas em conjunto com o programa de aceleração 6120 na Seção C, o cliente 10 pode instalar e executar automaticamente o aplicativo mediante o recebimento. Em uma outra modalidade, o servidor 30 executa o aplicativo em um servidor em nome do cliente, e transmite a exibição para o cliente 10 através de um protocolo de exibição remota ou camada de apresentação. Em ainda uma outra modalidade, a aplicação 1250 transmite o aplicativo para o cliente 10 ou transmite o aplicativo para o cliente 10 para instalação e/ou execução. Em algumas modalidades, a aplicação 1250 e/ou o servidor 30 transmitem o ambiente de computação 15 compreendendo o aplicativo. Em outras modalidades, a aplicação 1250 e/ou o servidor 30 transmitem o ambiente de computação 15 em resposta a uma requisição.

[00830] Na etapa 8015, a aplicação 1250 e/ou o programa de aceleração 6120 aceleram a entrega do aplicativo para execução para o cliente 10. Em uma modalidade, a aplicação 1250 executa ou aplica uma ou mais da pluralidade de técnicas de aceleração descritas na Seção C acima. Em uma outra modalidade, o programa de aceleração 6120 executa ou aplica uma ou mais da pluralidade de técnicas de aceleração de lado de cliente também descritas na Seção C acima. Em algumas modalidades, o programa de aceleração 1250 e a

aplicação 6120 trabalham juntos ou em conjunto com cada outro para a realização de uma pluralidade de técnicas de aceleração no cliente 10 e na aplicação 1250. Por exemplo, o programa de aceleração 6120 pode executar um primeiro conjunto de uma ou mais técnicas de aceleração, enquanto a aplicação 1250 executa um segundo conjunto de uma ou mais técnicas de aceleração. Em uma modalidade, o programa de aceleração 1250 e a aplicação 6120 executam as mesmas técnicas de aceleração. Em uma outra modalidade, o programa de aceleração 1250 e a aplicação 6120 executam técnicas de aceleração diferentes.

[00831] Em uma modalidade, a aplicação 1250 e/ou o programa de aceleração 6120 aceleram qualquer carga útil comunicada através de uma conexão de camada de transporte entre o cliente 10 e o servidor 30. Em algumas modalidades, o servidor 30 transmite o aplicativo como um ou mais arquivos de dados através de uma conexão de camada de transporte, tal como uma carga útil de um pacote de TCP/IP. Em outras modalidades, o servidor 30 transmite o aplicativo através de um protocolo de camada de aplicativo ou de um protocolo de transmissão por uma conexão de camada de transporte. Em uma outra modalidade, o servidor 30 transmite uma saída de exibição através de um protocolo de ICA ou de RDP através da conexão de camada de transporte. Em qualquer uma destas modalidades, a aplicação 1250 e/ou o programa de aceleração 6120 aceleram a entrega do aplicativo através de cargas úteis de pacotes de camada de transporte.

[00832] Na etapa 8020, o cliente 10 transmite uma requisição para um arquivo de dados para uso pelo aplicativo ou ambiente de computação 15. Em algumas modalidades, a requisição para o arquivo de dados é transmitida com a requisição para execução de um aplicativo na etapa 8005. Em uma modalidade, a requisição para

execução de um aplicativo inclui a requisição pelo arquivo de dados. Em outras modalidades, o aplicativo ou o ambiente de computação requisita o arquivo de dados no decorrer da realização de qualquer funcionalidade, operações ou lógica do aplicativo ou ambiente de computação. Por exemplo, o aplicativo ou ambiente de computação 15 pode requisitar quaisquer macros, scripts, dados de configuração, perfil, modelos ou regras de um servidor 30. Em algumas modalidades, o aplicativo requisita o arquivo de dados como um processo de fundo ou uma tarefa do aplicativo. Em uma modalidade, o usuário do aplicativo ou do ambiente de computação 15 requisita o arquivo de dados para leitura, acesso ou processamento de outra forma do arquivo para edição através de um aplicativo, tal como a abertura de um documento para edição através de um aplicativo de processamento de texto. Em algumas modalidades, o usuário arrasta e solta um aplicação em um aplicativo do ambiente de computação para requisitar o arquivo de dados. Em outras modalidades, o usuário pode requisitar o arquivo de dados através de uma interface de arquivo e diretório, por exemplo, um explorador de arquivo no sistema operacional Windows, para um armazenamento de um sistema de armazenamento em rede ou remoto, tal como um driver de rede de um servidor central.

[00833] Na etapa 8025, o servidor 30 ou a aplicação 1250 transmite o arquivo de dados requisitado para o cliente 10. Em algumas modalidades, o servidor 30 ou a aplicação 1250 transmite o arquivo de dados para o cliente 10 em resposta à requisição da etapa 8020. Em outras modalidades, o servidor 30 ou a aplicação 1250 transmite o arquivo de dados para o cliente 10 sem uma requisição do cliente 10. Por exemplo, o servidor 30 pode "empurrar" uma atualização para um arquivo de dados para o cliente 10. Em uma modalidade, o servidor 30 transmite o arquivo de dados requisitado para o cliente 10. Em uma

outra modalidade, a aplicação 1250 transmite o arquivo de dados requisitado para o cliente 10. Por exemplo, em uma modalidade, a aplicação 1250 intercepta uma requisição pelo arquivo de dados, checa o cache da aplicação 1250 quanto ao arquivo de dados, e transmite o arquivo de dados armazenado em cache para o cliente 10. Ainda em uma outra modalidade, o programa de aceleração 6120 intercepta a requisição de arquivo de dados no cliente 10 e provê o arquivo de dados para o cliente 10 através de um cache do programa de aceleração 6120. Em algumas modalidades, a aplicação 1250 ou o servidor 30 transmite o arquivo de dados através de um protocolo de transmissão ou de uma transmissão. Em outras modalidades, a aplicação 1250 ou o servidor 30 transmite o arquivo de dados através de qualquer tipo e forma de protocolo de armazenamento em cache.

[00834] Na etapa 8030, a aplicação 1250 e/ou o programa de aceleração 6120 aceleram a entrega ou a transmissão do arquivo de dados para o cliente 10. Em algumas modalidades, o arquivo de dados pode ser transmitido através de qualquer tipo e forma de protocolo, tal como um protocolo de camada de aplicativo por um protocolo de camada de transporte. Em uma modalidade, a aplicação 1250 acelera a transmissão do arquivo de dados. Em uma outra modalidade, o programa de aceleração 6120 acelera a transmissão do arquivo de dados. Em algumas modalidades, a aplicação 1250 em conjunto com o programa de aceleração 1250 acelera a transmissão do arquivo de dados. Conforme discutido aqui, a aplicação 1250 e/ou o programa de aceleração 6120 podem realizar uma ou mais de uma pluralidade de técnicas de aceleração no cliente 10 e na aplicação 30 para aceleração da transmissão do arquivo de dados. Em algumas modalidades, a aplicação 1250 e/ou o programa de aceleração 6120 podem armazenar em cache um ou mais arquivos de dados no cliente 10 ou na aplicação 1250 para uso pelo aplicativo ou ambiente de

computação 15.

Exemplos Representativos

[00835] Como uma modalidade de exemplo, um usuário pode estar localizado em uma filial trabalhando em uma máquina local 10. O usuário pode desejar usar um aplicativo de processamento de texto, tal como o MICROSOFT Word para editar um documento da companhia, residente nas máquinas remotas 30 localizadas em um escritório central. O usuário então pode navegar por um navegador da web para um website corporativo hospedado pela máquina remota 30. Uma vez que o usuário seja autenticado pela máquina remota 30, a máquina remota 30 pode preparar e transmitir para a máquina local 10 uma página em HTML que inclua uma janela de Vizinhança de Programa, conforme descrito aqui nas figuras 3A e 3B na qual aparecem ícones gráficos representando programas aplicativos aos quais a máquina local 10 tem acesso. O usuário da máquina local 10 pode invocar a execução de um aplicativo ao clicar em um ícone. Um agente de política conforme descrito nas figuras 4A a 4D pode determinar, então, se e como a máquina local 10 pode acessar o aplicativo de processamento de texto. O aplicativo então pode ser localmente instalado e executado usando-se as técnicas descritas nas figuras 20 a 21. O usuário então pode usar o aplicativo para selecionar um documento na máquina remota 30 para edição. Uma aplicação 1250 pode então acelerar a entrega do arquivo para a máquina local 10 usando quaisquer técnicas descritas aqui, tal como multiplexação de TCP.

[00836] Como um outro exemplo, um segundo usuário pode estar localizado em uma filial trabalhando em uma máquina local 10. O usuário pode desejar acessar, através da conta corporativa do usuário, um e-mail contendo um arquivo anexado. O aplicativo de e-mail e os arquivos de dados de e-mail podem residir em um escritório central.

Mediante uma requisição do usuário para acessar o aplicativo de e-mail, um agente de política conforme descrito nas figuras 4A a 4D pode determinar que é para transmitir o aplicativo de e-mail para o usuário usando as técnicas de transmissão descritas aqui. Um agente de política também pode determinar que é para instalar um programa de aceleração conforme descrito aqui na máquina local 10. A transmissão de aplicativo pode ser acelerada usando-se técnicas descritas aqui, tal como um armazenamento em cache dinâmico. Uma aplicação 1250 pode acelerar a entrega do arquivo pelo uso de uma técnica de aceleração, tal como agrupamento de TCP, conforme descrito aqui. A aplicação também pode armazenar em cache alguns ou todos os arquivos de dados entregues para a máquina remota, de modo a acelerar as requisições posteriores. O armazenamento em cache pode ser feito na aplicação 1250 ou na máquina local 10 em conjunto com o programa de aceleração.

[00837] Como um terceiro exemplo, um usuário localizado em uma filial pode desejar acessar um programa de planilha, tal como o MICROSOFT Excel para a atualização de uma planilha. O usuário pode usar uma máquina local 10 para o estabelecimento de uma conexão de SSL com uma máquina remota 30 em um escritório central, e selecionar o aplicativo de planilha a partir de uma vizinhança de programa, conforme descrito nas figuras 3A e 3B. Um agente de coleta conforme descrito na figura 4D então pode coletar uma informação sobre a máquina local para determinar se o aplicativo de planilha pode ser transmitido para a máquina local 10 através da conexão de SSL. A conexão de SSL pode ser acelerada por uma aplicação 1250 provendo agrupamento e multiplexação de conexão de SSL ou TCP, conforme descrito aqui. O usuário então pode selecionar um arquivo a partir de dentro do aplicativo de planilha para edição. A máquina local 10 pode transmitir a requisição para o arquivo para a

máquina remota. Uma aplicação 1250 então pode usar as técnicas de compressão descritas aqui para a aceleração da entrega do arquivo para o usuário.

[00838] Embora descritos geralmente acima como um sistema de entrega de aplicativo e uma entrega de aceleração de aplicativo de um ambiente de computação para um cliente, o sistema de entrega de aplicativo e a aplicação podem acelerar a entrega de uma pluralidade de ambientes de computação, aplicativos e/ou arquivos de dados para um cliente. Por exemplo, o sistema de entrega de aplicativo e a aplicação podem acelerar a entrega para o cliente de um primeiro ambiente de computação associado a um tipo de sistema operacional e de um segundo ambiente de computação associado a um segundo tipo de sistema operacional. Adicionalmente, o sistema de entrega de aplicativo e a aplicação podem acelerar a entrega de um ambiente de computação, aplicativo e/ou arquivo de dados para uma pluralidade de clientes. Mais ainda, embora geralmente descrito acima como um sistema de entrega de aplicativo e uma aplicação acelerando a entrega de um ambiente de computação para um usuário remoto ou um cliente remoto, o sistema de entrega de aplicativo e a aplicação podem acelerar a entrega de um ambiente de computação, um aplicativo e/ou um arquivo de dados para qualquer cliente, local, remoto ou de outra forma, tal como um cliente em uma LAN do servidor.

[00839] Mais ainda, embora geralmente descrita acima como uma aplicação entre o cliente e o sistema de entrega de aplicativo, uma pluralidade de aparelhagens pode ser usada entre um ou mais clientes e um ou mais servidores. Em algumas modalidades, uma primeira aplicação reside na rede do cliente, e uma segunda aplicação reside na rede do servidor. Em uma modalidade, a primeira aplicação e a segunda aplicação se comunicam entre si na realização das

operações descritas aqui. Por exemplo, a primeira aplicação e a segunda aplicação podem se comunicar através de qualquer protocolo de comunicação interno, de alta performance ou de aplicação para aplicação. Adicionalmente, uma pluralidade de sistemas de entrega de aplicativo pode ser usada em conjunto com uma aplicação ou uma pluralidade de aparelhagens. O sistema de entrega de aplicativo e a aplicação podem ser empregados em uma variedade de ambientes de rede e arquiteturas de infra-estrutura.

[00840] As modalidades podem ser providas como um ou mais programas que podem ser lidos em computador concretizados em um ou mais artigos de fabricação. O artigo de fabricação pode ser um disco flexível, um disco rígido, um disco compacto, um disco versátil digital, um cartão de memória flash, uma PROM, uma RAM, uma ROM ou uma fita magnética. Em geral, os programas que podem ser lidos em computador podem ser implementados em qualquer linguagem de programação. Alguns exemplos de linguagens que podem ser usadas incluem C, C++, C# ou JAVA. Os programas de software podem ser armazenados em um ou mais artigos de fabricação como um código de objeto.

REIVINDICAÇÕES

1. Método para aceleração da entrega de um ambiente de computação para um cliente remoto (10, 10', 10"), o método **caracterizado** pelo fato de que compreende as etapas de:

recebimento, por um servidor (30, 30', 30"), de uma requisição a partir de um cliente remoto (10, 10', 10") para execução de um aplicativo, o cliente remoto e o servidor em comunicação através de uma aplicação (1250);

transmissão, pelo servidor, para o cliente remoto de um aplicativo para execução;

determinação, pela aplicação (1250), que o aplicativo é capaz de ser acelerado;

transmissão, pela aplicação (1250), em resposta à determinação de um programa de aceleração para o cliente remoto;

identificação, pelo servidor, de um arquivo de dados utilizável pelo aplicativo;

transmissão, pelo servidor, o dito arquivo de dados identificado para acesso pelo aplicativo transmitido, antes de receber uma requisição para o arquivo de dados;

aceleração, pela aplicação (1250), da transmissão do arquivo de dados para o cliente remoto;

em que o programa de aceleração no cliente remoto acelera as comunicações entre o cliente remoto e o servidor; e

em que a transmissão continua do aplicativo ou transmissão do arquivo de dados identificado é realizada através de uma conexão de camada de transporte agrupada usada por uma pluralidade de clientes

2. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que compreende a aceleração, pela aplicação (1250), da transmissão do aplicativo para o cliente remoto (10, 10', 10").

3. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que compreende a aceleração, pela aplicação (1250), de uma dentre a transmissão do arquivo de dados ou a transmissão do aplicativo pela execução de uma das técnicas de aceleração a seguir:

compressão;

multiplexação de Protocolo de Controle de Transmissão;

armazenamento em buffer de Protocolo de Controle de Transmissão; e

armazenamento em cache.

4. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que compreende o estabelecimento, pela aplicação (1250), de uma dentre uma conexão de rede privada virtual ou uma conexão de Camada de Soquete Segura (SSL) com o cliente remoto (10, 10', 10").

5. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que compreende a instalação e a execução automática, pelo cliente remoto (10, 10', 10"), do programa de aceleração mediante o recebimento a partir da aplicação (1250).

6. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que compreende a realização, por um programa de aceleração no cliente remoto (10, 10', 10"), de uma das técnicas de aceleração a seguir:

compressão;

multiplexação de Protocolo de Controle de Transmissão;

armazenamento em buffer de Protocolo de Controle de Transmissão; e

armazenamento em cache.

7. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que compreende a multiplexação, pela aplicação (1250), de pacotes de rede da transmissão de um dentre o aplicativo

transmitido ou o arquivo de dados através de uma conexão de camada de transporte agrupada utilizada por uma pluralidade de clientes.

8. Sistema para a aceleração da entrega de um ambiente de computação para um cliente remoto (10, 10', 10''), o sistema **caracterizado** pelo fato de que compreende:

uma aplicação (1250), que compreende um processador, para a aceleração das comunicações entre um ou mais clientes e um ou mais servidores (30, 30', 30'');

um servidor recebendo uma requisição a partir de um cliente remoto para execução de um aplicativo, o servidor transmitindo para o cliente remoto através da aplicação um aplicativo para execução, em resposta à requisição;

onde o servidor identifica um arquivo de dados utilizável pelo aplicativo e transmite o arquivo de dados identificado para acesso pelo aplicativo transmitido, identifica um arquivo de dados utilizável pelo aplicativo; e

a aplicação acelera a transmissão do arquivo de dados para o cliente remoto, armazenando em cache o arquivo de dados no cache, interceptando uma requisição para o arquivo de dados e transmitindo o arquivo de dados em cache para o cliente remoto em resposta à requisição;

em que a transmissão da aplicação ou transmissão do arquivo de dados identificado é realizada através de uma conexão de camada de transporte agrupada utilizada por uma pluralidade de clientes.

9. Sistema, de acordo com a reivindicação 8, **caracterizado** pelo fato de que a aplicação (1250) acelera a transmissão do aplicativo para o cliente remoto (10, 10', 10'').

10. Sistema, de acordo com a reivindicação 8, **caracterizado** pelo fato de que compreende um programa de

aceleração no cliente remoto (10, 10', 10'') acelerando as comunicações entre o cliente remoto e o servidor (30, 30', 30'').

11. Sistema, de acordo com a reivindicação 8, **caracterizado** pelo fato de que a aplicação (1250) estabelece uma dentre uma conexão de rede privada virtual ou uma conexão de Camada de Soquete Segura (SSL) com o cliente remoto (10, 10', 10'').

12. Sistema, de acordo com a reivindicação 8, **caracterizado** pelo fato de que a aplicação (1250) transmite um programa de aceleração para o cliente remoto (10, 10', 10'') mediante uma requisição a partir do cliente para o estabelecimento de uma dentre uma conexão ou uma sessão com o servidor (30, 30', 30'').

13. Sistema, de acordo com a reivindicação 12, **caracterizado** pelo fato de que o cliente remoto (10, 10', 10'') automaticamente instala e executa o programa de aceleração mediante o recebimento a partir da aplicação (1250).

14. Sistema, de acordo com a reivindicação 12 **caracterizado** pelo fato de que o programa de aceleração no cliente remoto (10, 10', 10'') executa uma das técnicas de aceleração a seguir:

compressão;

descompressão;

multiplexação de Protocolo de Controle de Transmissão;

armazenamento em buffer de Protocolo de Controle de Transmissão; e

armazenamento em cache.

15. Sistema, de acordo com a reivindicação 8, **caracterizado** pelo fato de que a aplicação (1250) determina que o aplicativo é capaz de ser acelerado e transmite, em resposta à determinação, um programa de aceleração para o cliente remoto (10, 10', 10'').

16. Sistema, de acordo com a reivindicação 8,

caracterizado pelo fato de que a aplicação (1250) acelera uma dentre a transmissão do arquivo de dados ou a transmissão do aplicativo pela execução de uma das técnicas de aceleração a seguir:

compressão;

multiplexação de Protocolo de Controle de Transmissão;

armazenamento em buffer de Protocolo de Controle de Transmissão; e

armazenamento em cache.

17. Sistema, de acordo com a reivindicação 9, **caracterizado** pelo fato de que a aplicação (1250) multiplexa os pacotes de rede da transmissão de um dentre o aplicativo transmitido ou o arquivo de dados através de uma conexão de camada de transporte agrupada utilizada por uma pluralidade de clientes.

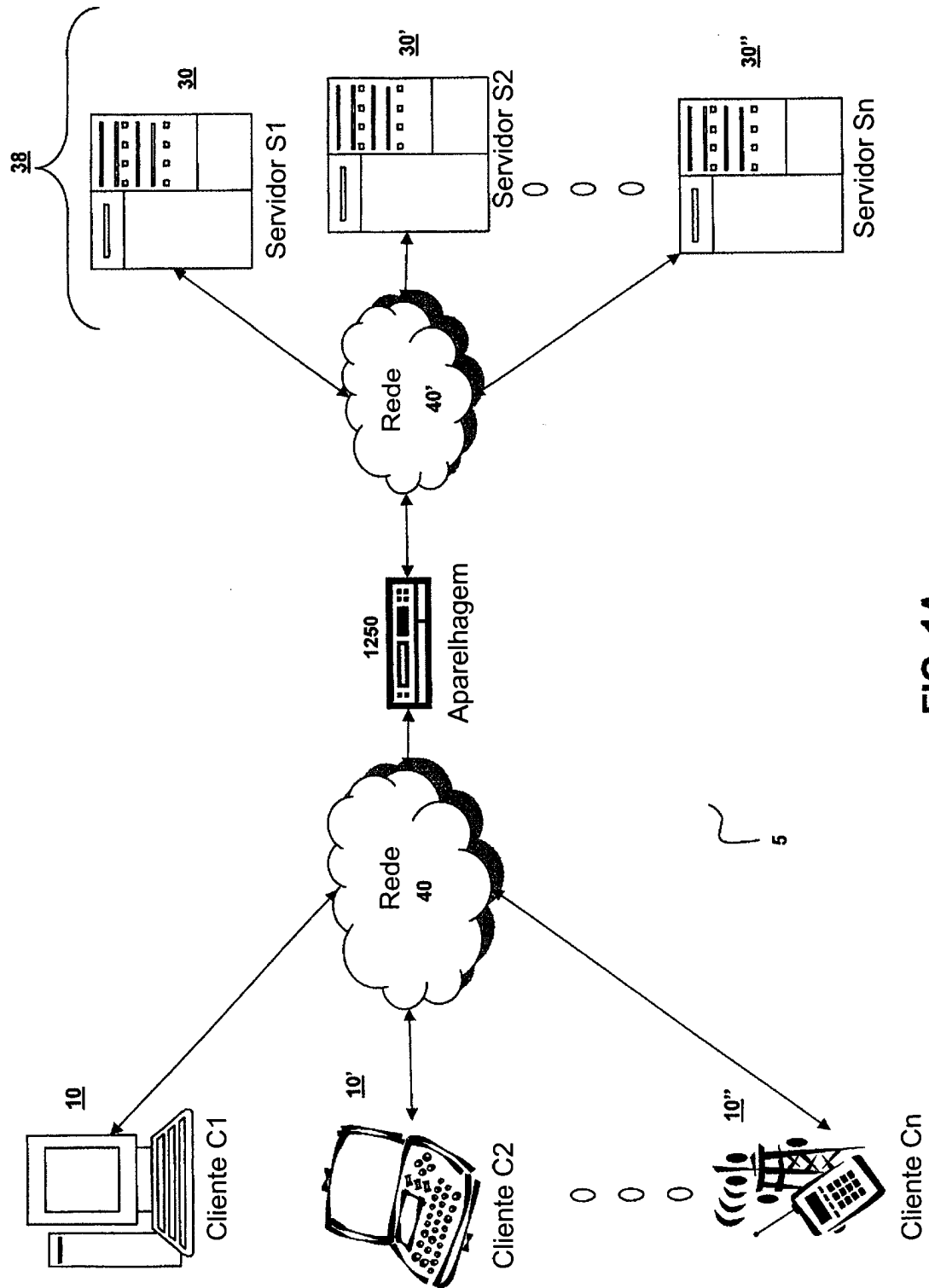


FIG. 1A

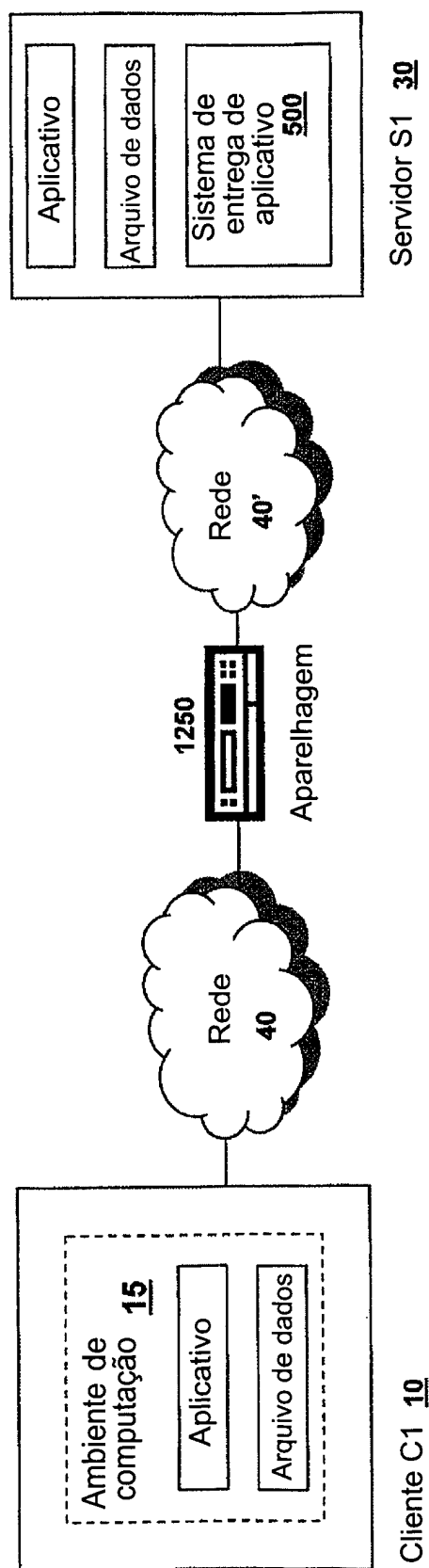
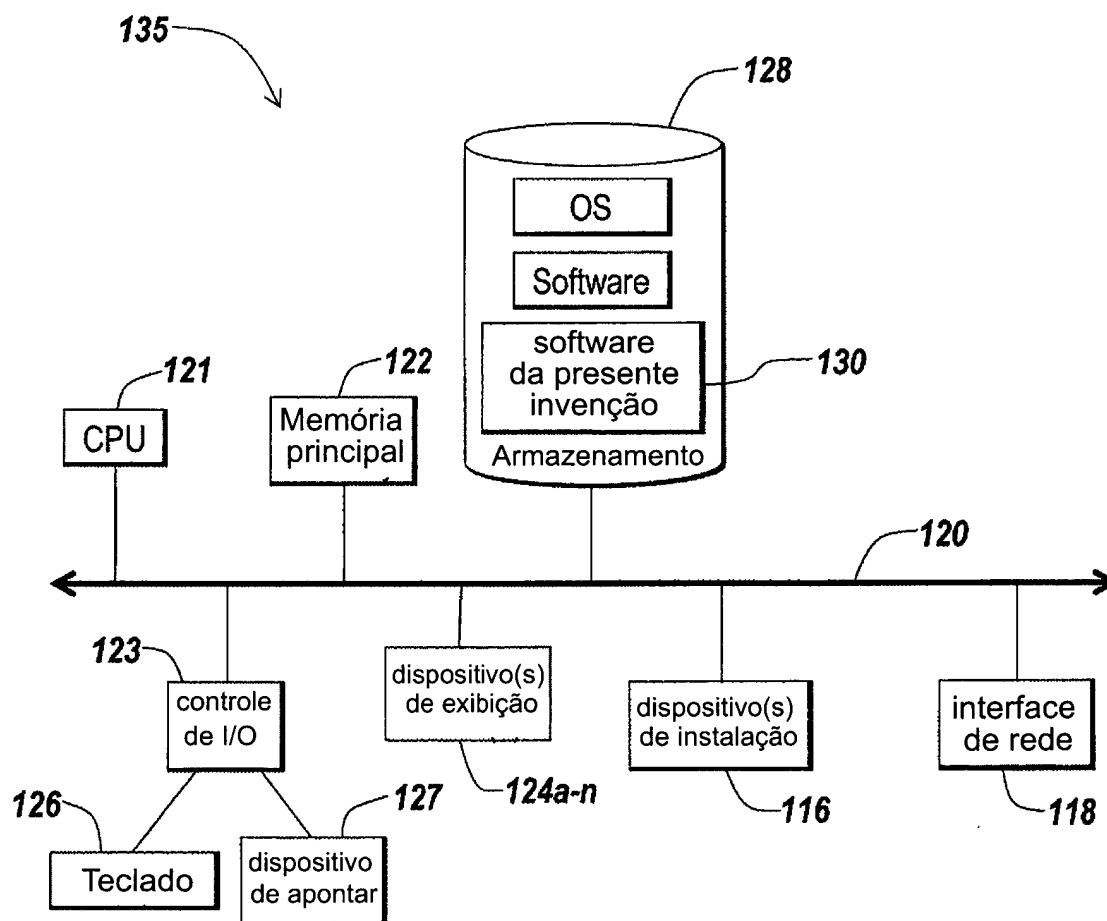


FIG. 1B

*Fig. 1C*

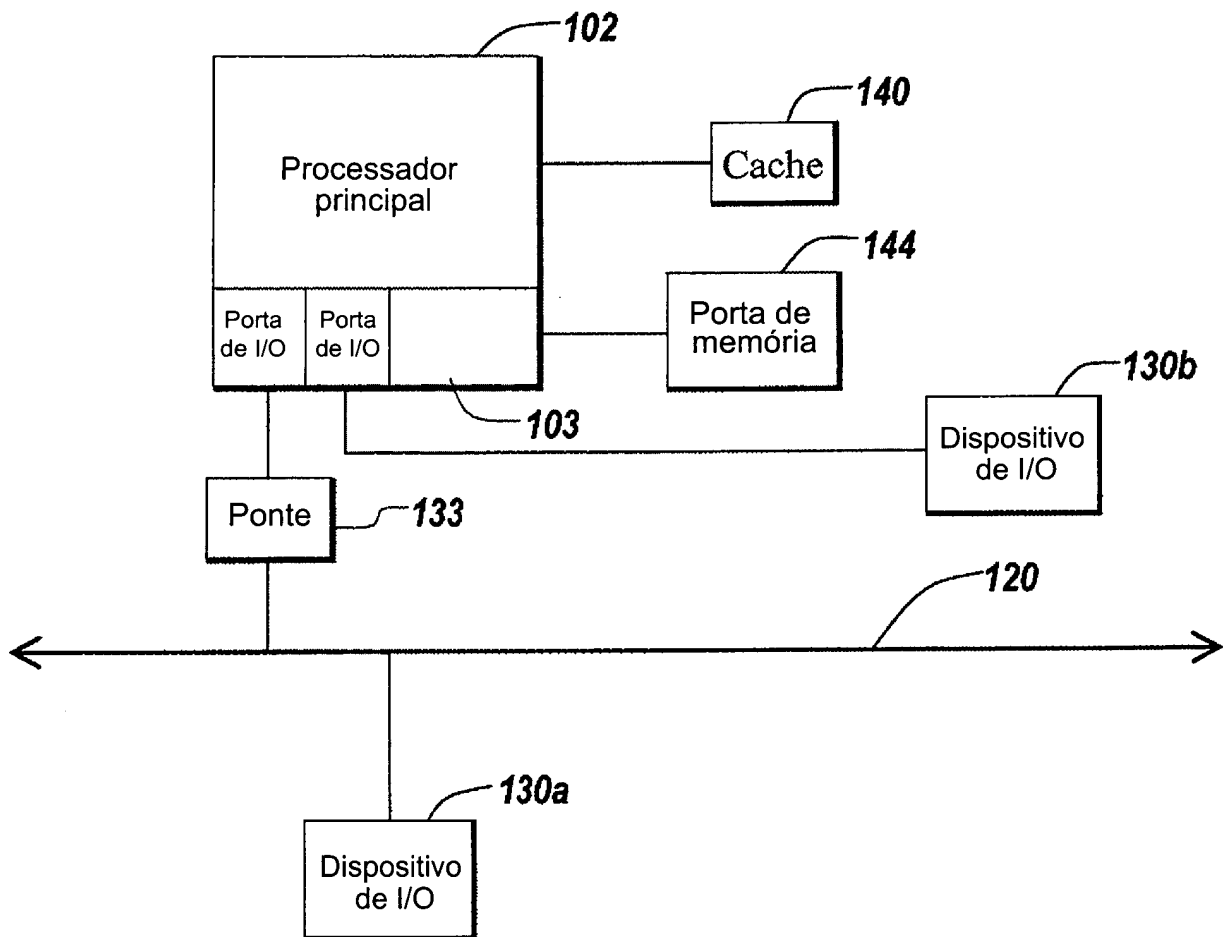


Fig. 1D

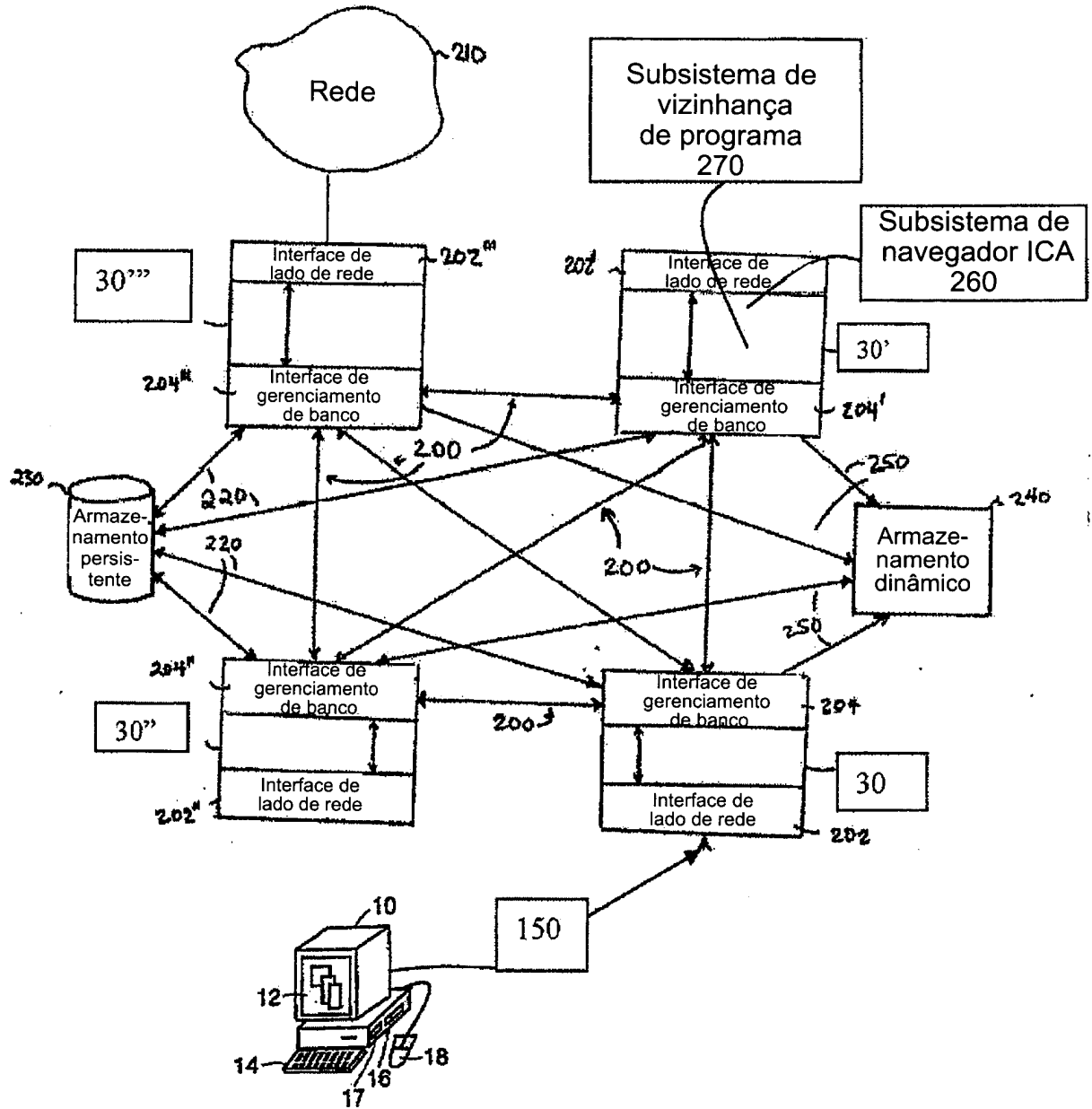


FIG. 1E

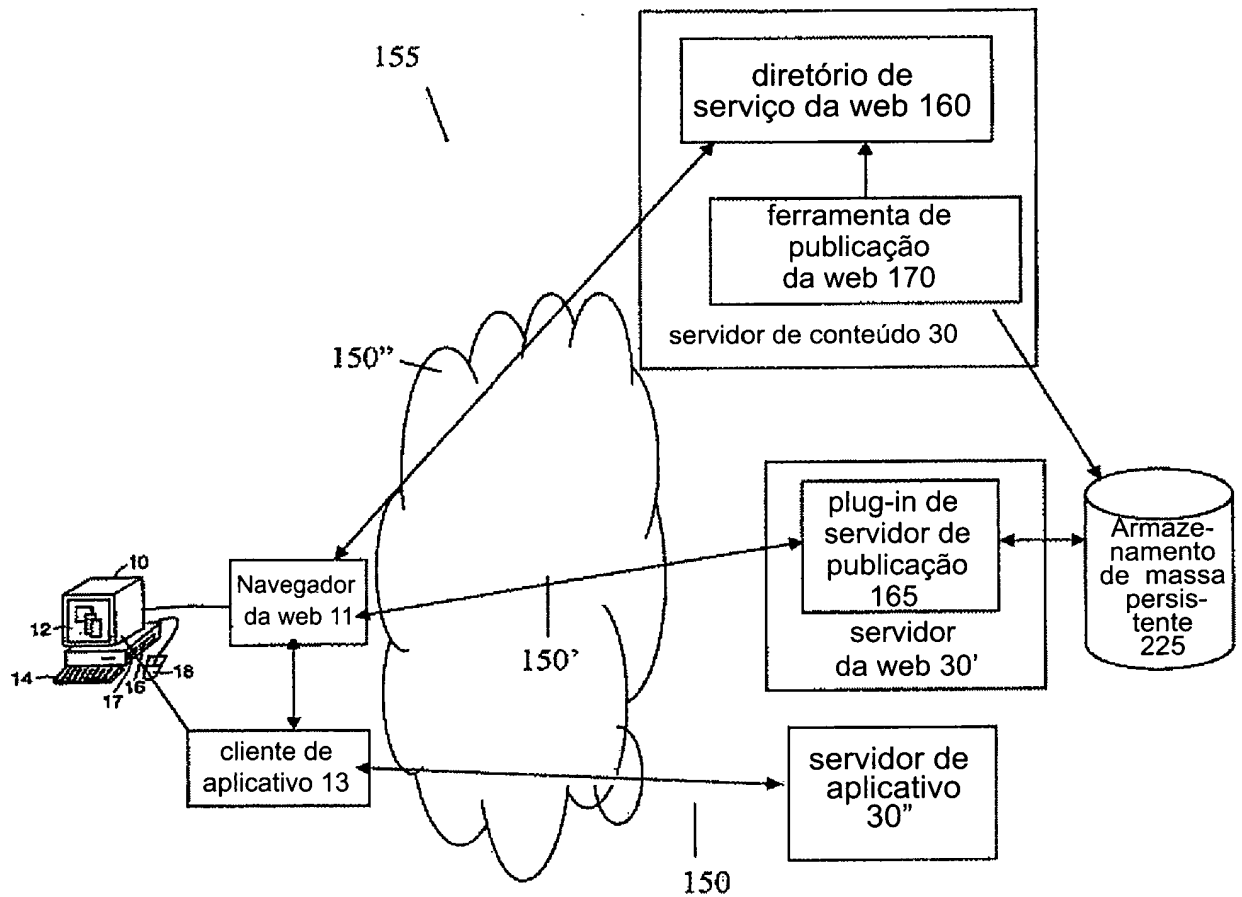
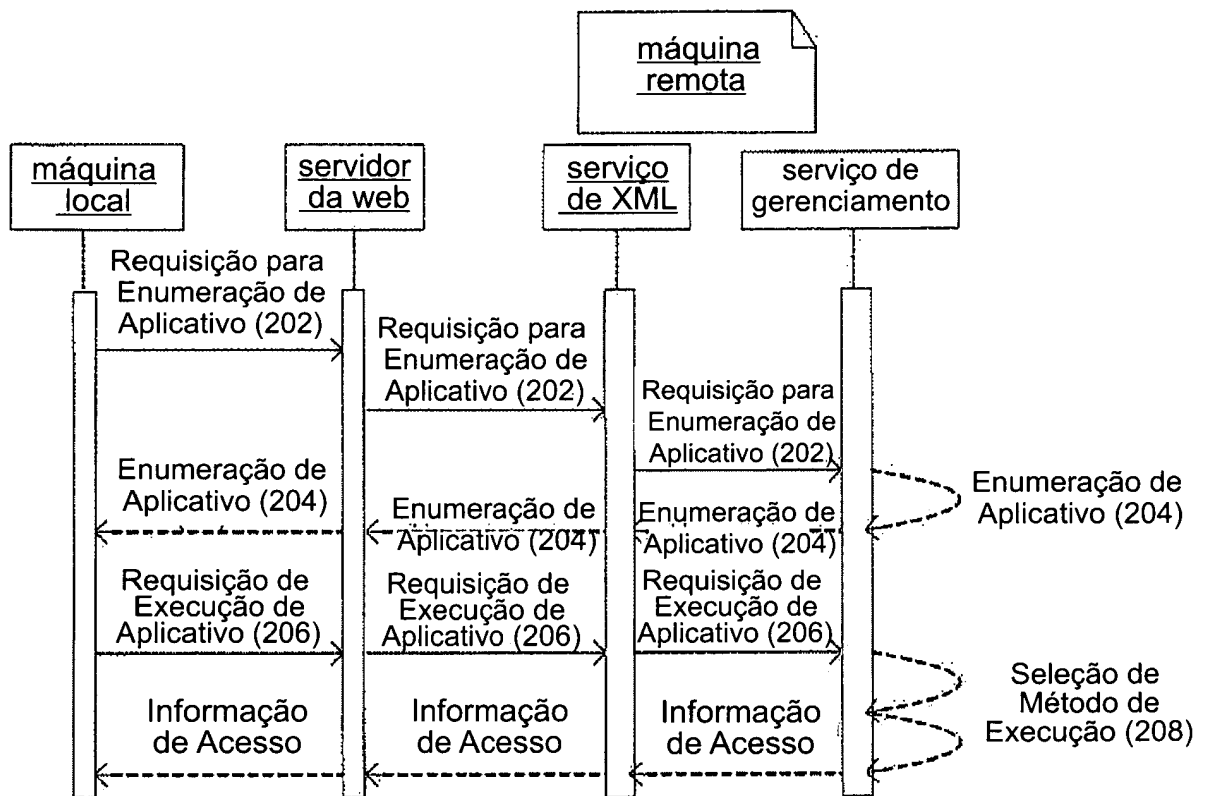


FIG. 1F

**FIG. 2**

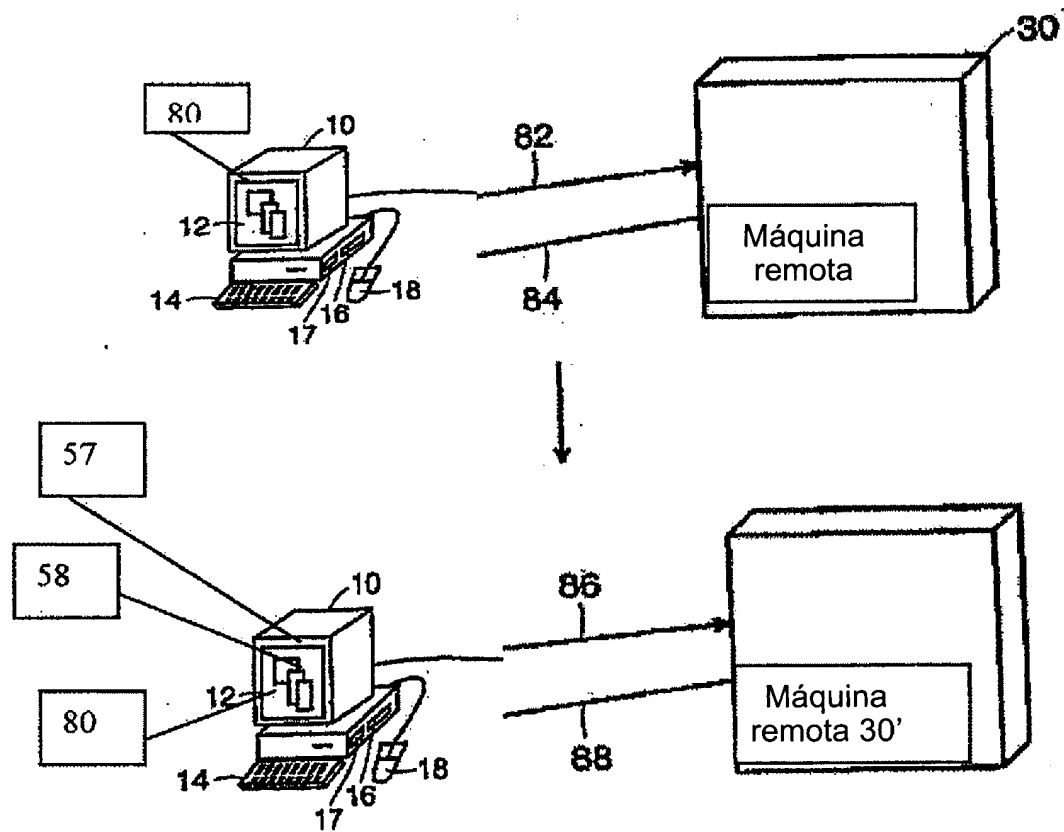


FIG. 3A

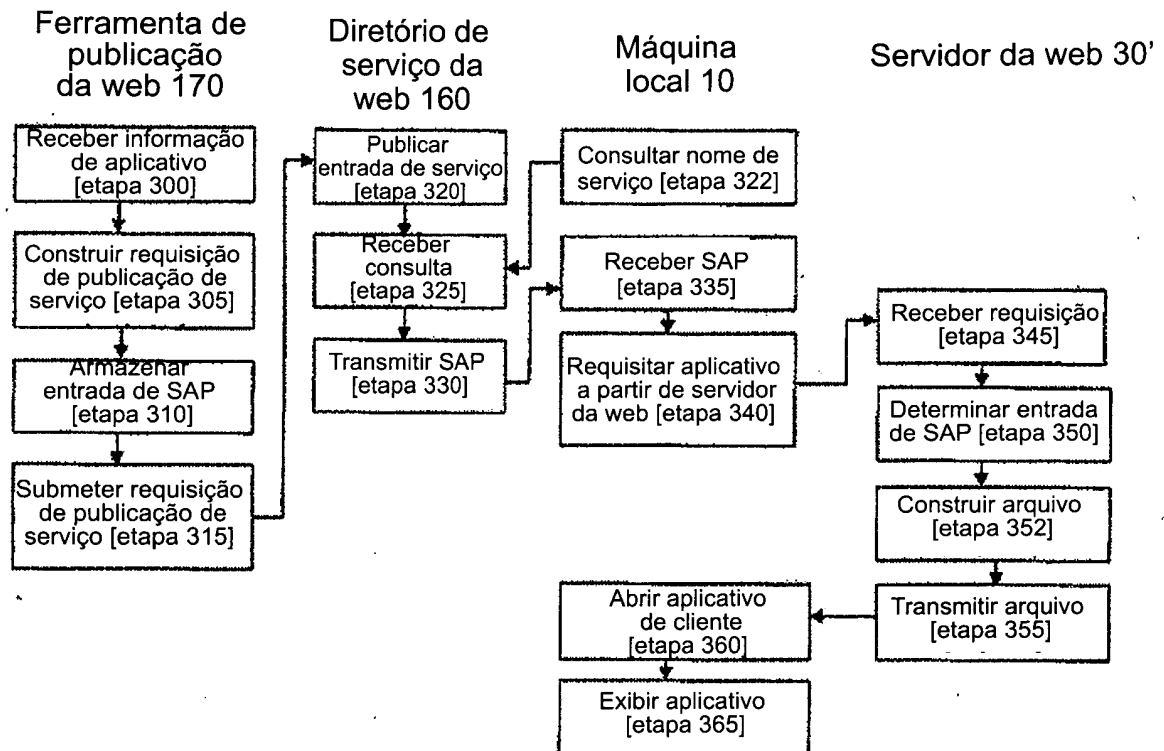
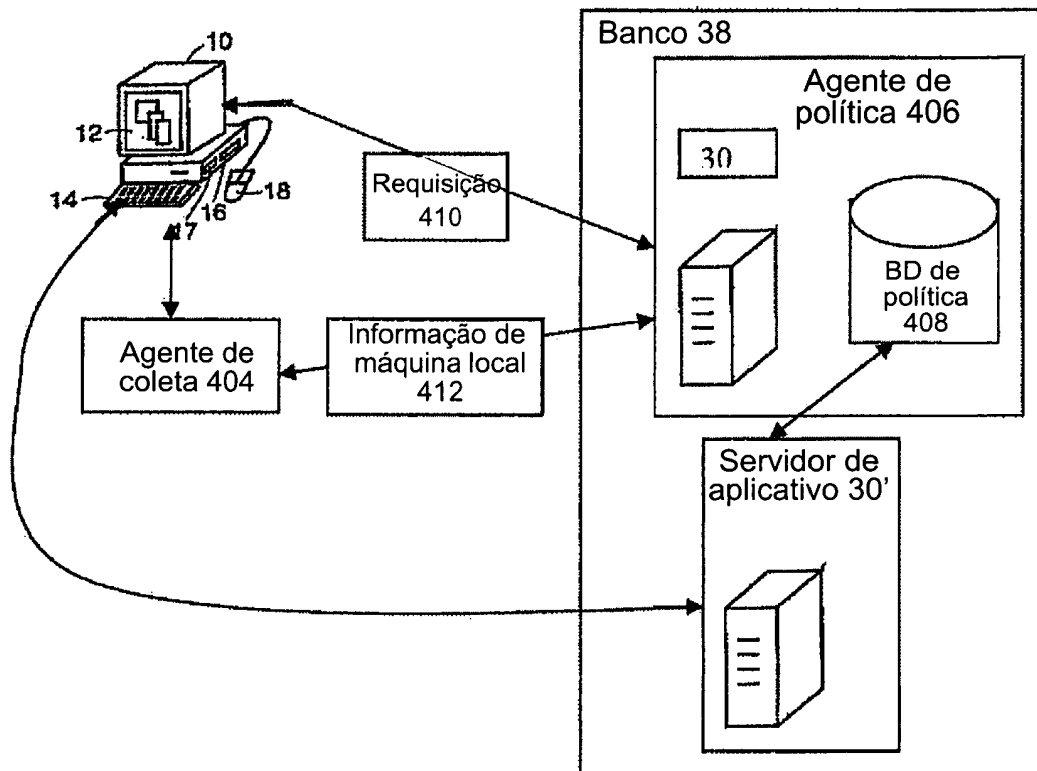


FIG. 3B

**FIG. 4A**

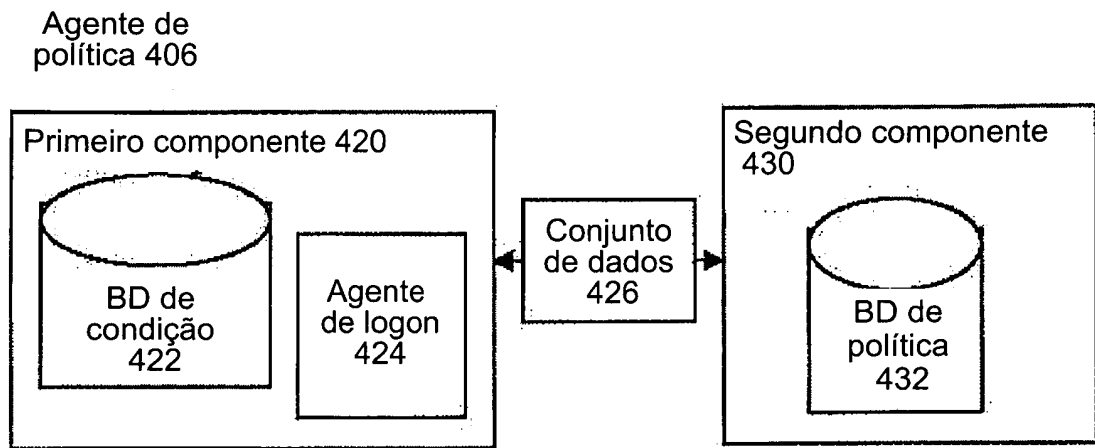
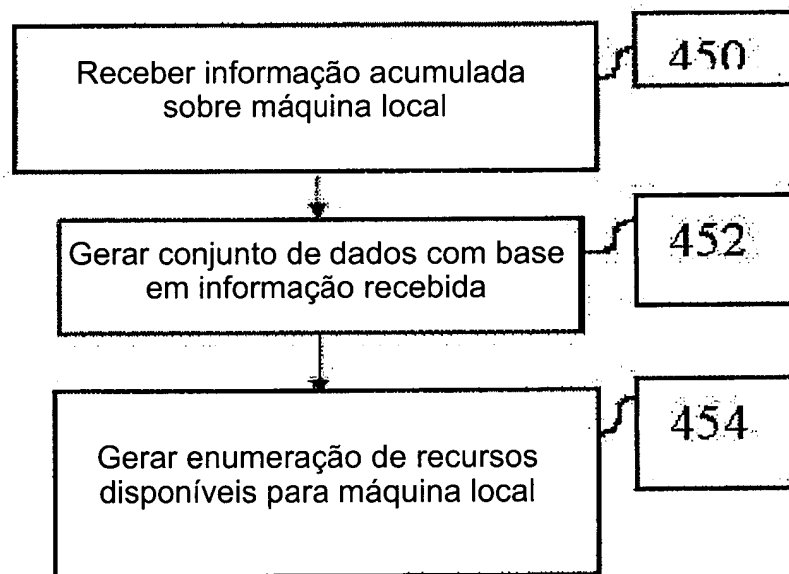


FIG. 4B

**FIG. 4C**

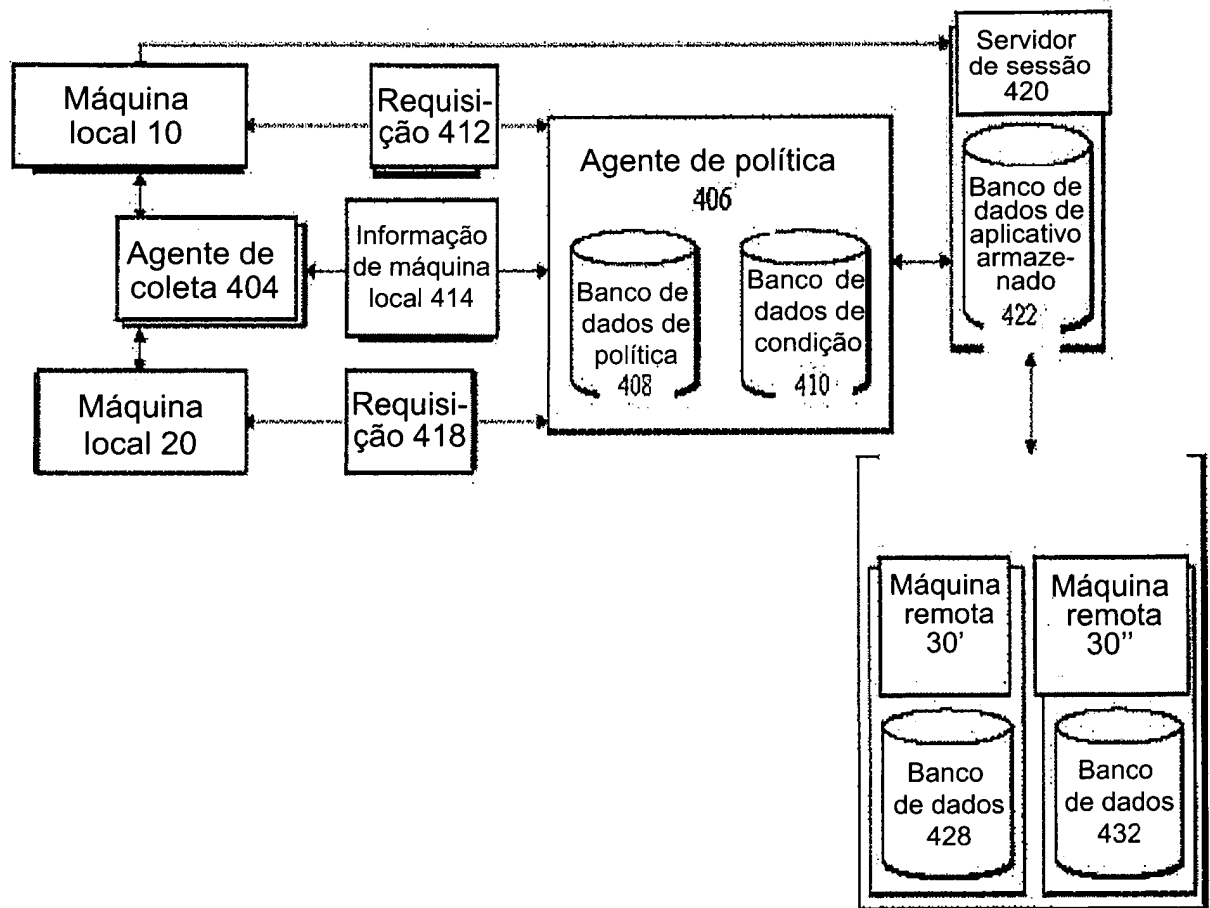
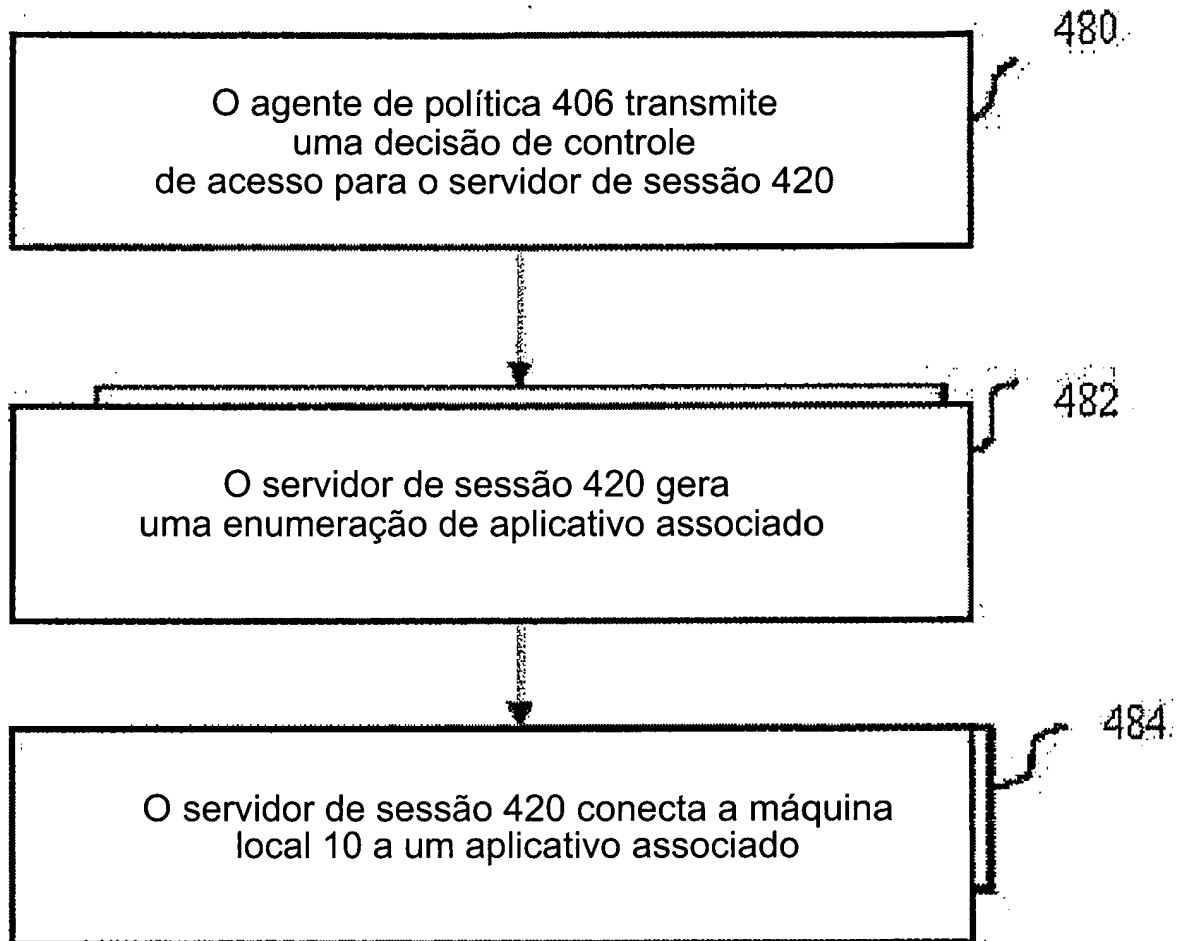


FIG. 4D

**FIG. 4E**

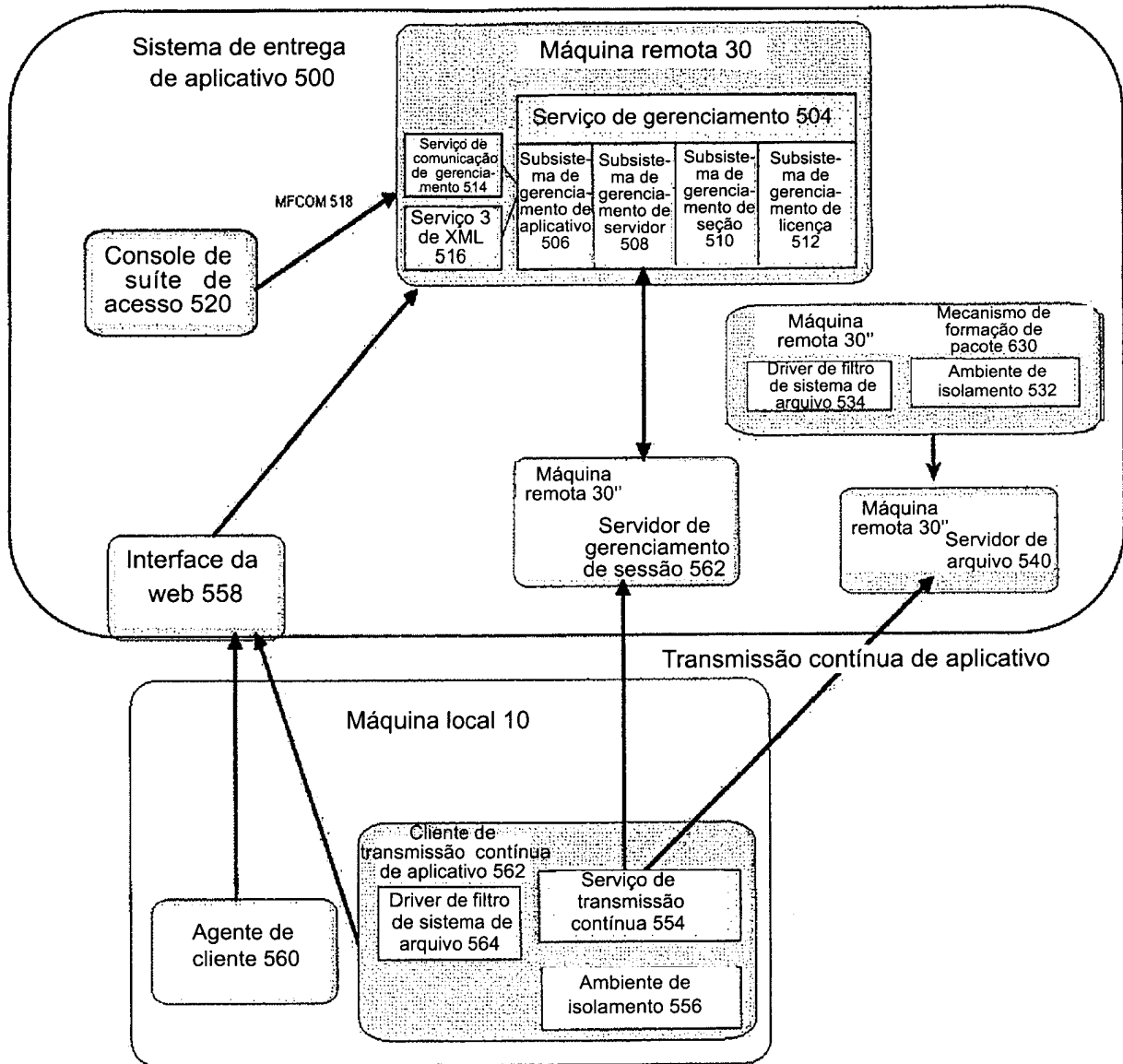
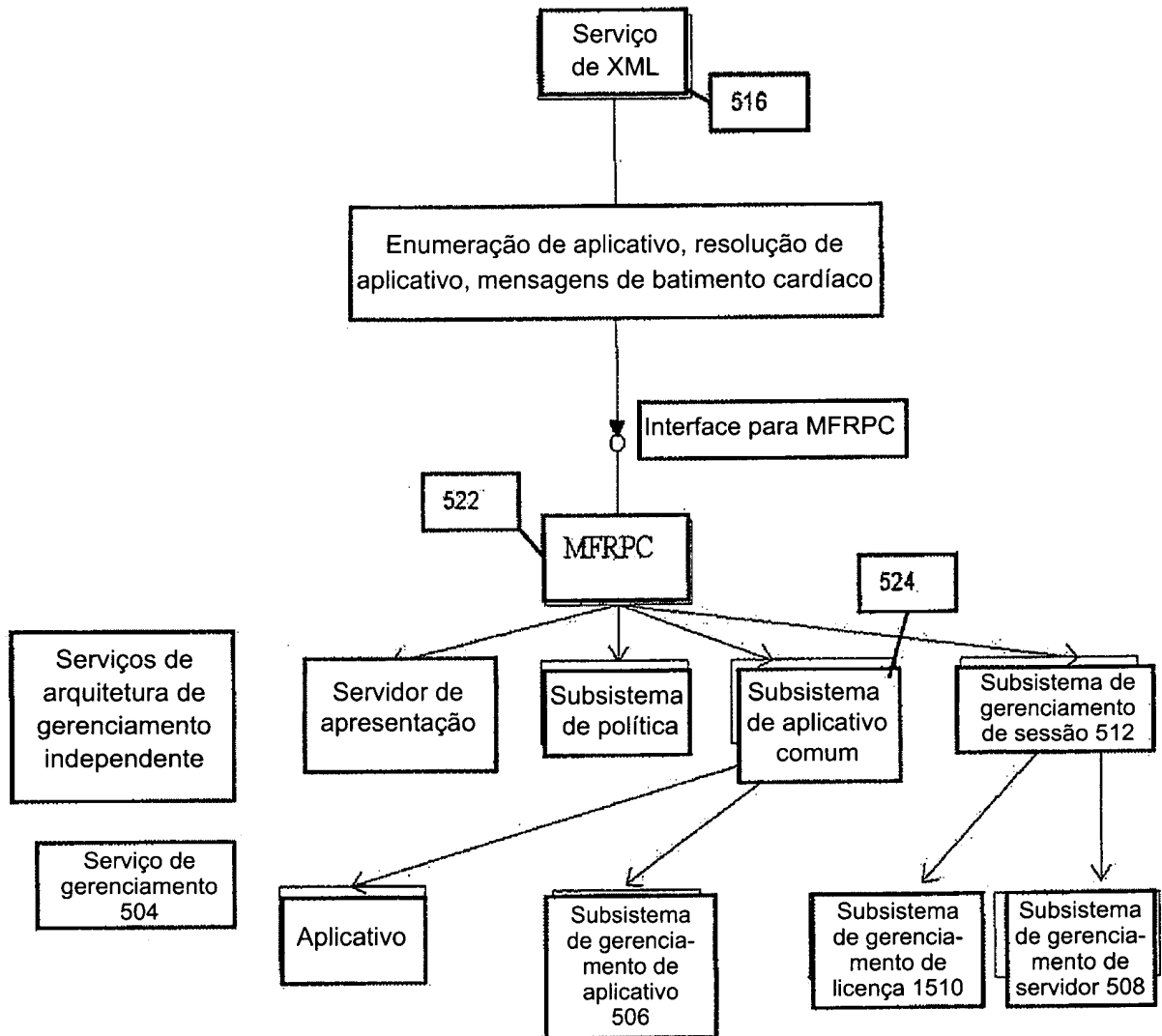


FIG. 5

**FIG. 6**

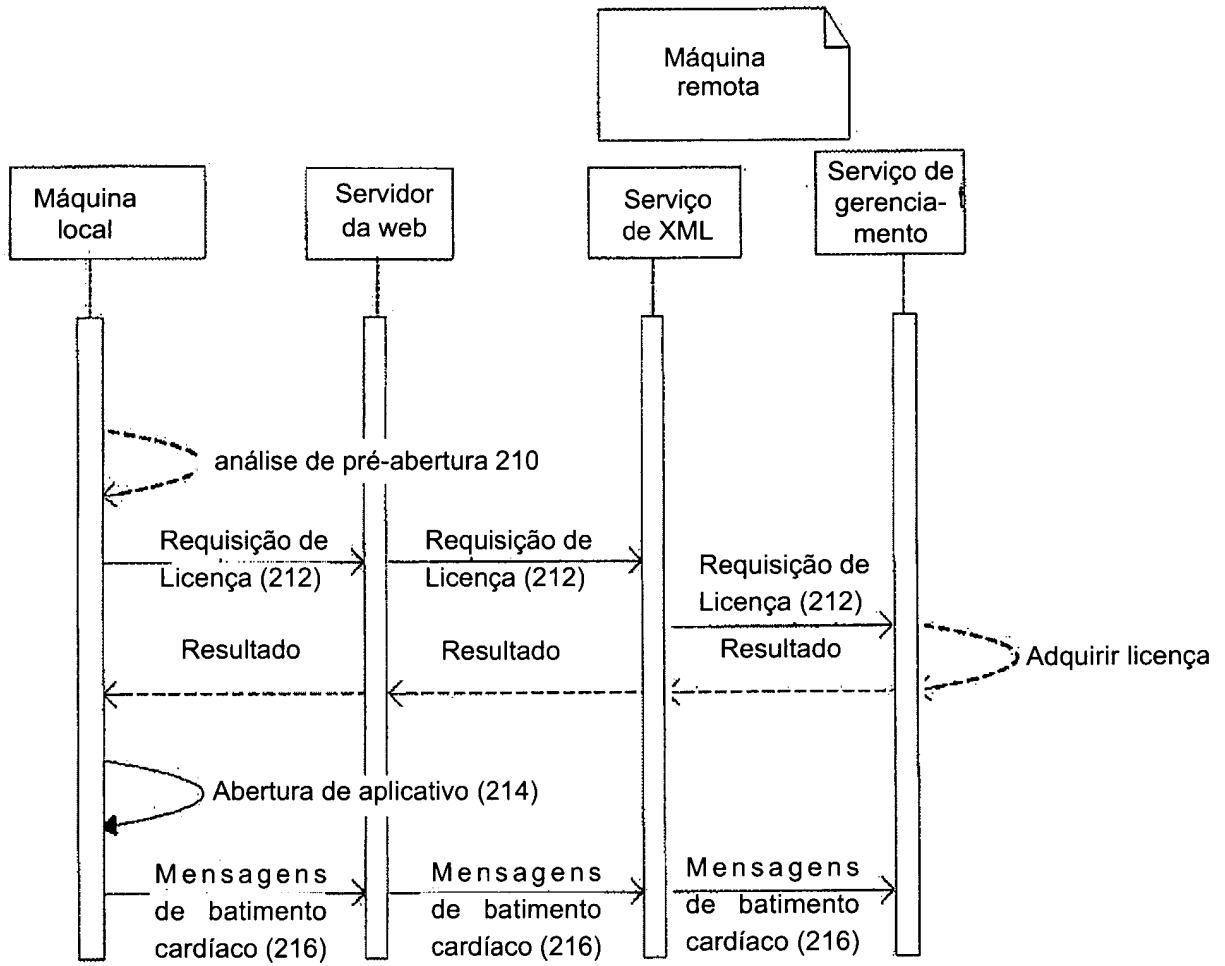


FIG. 7

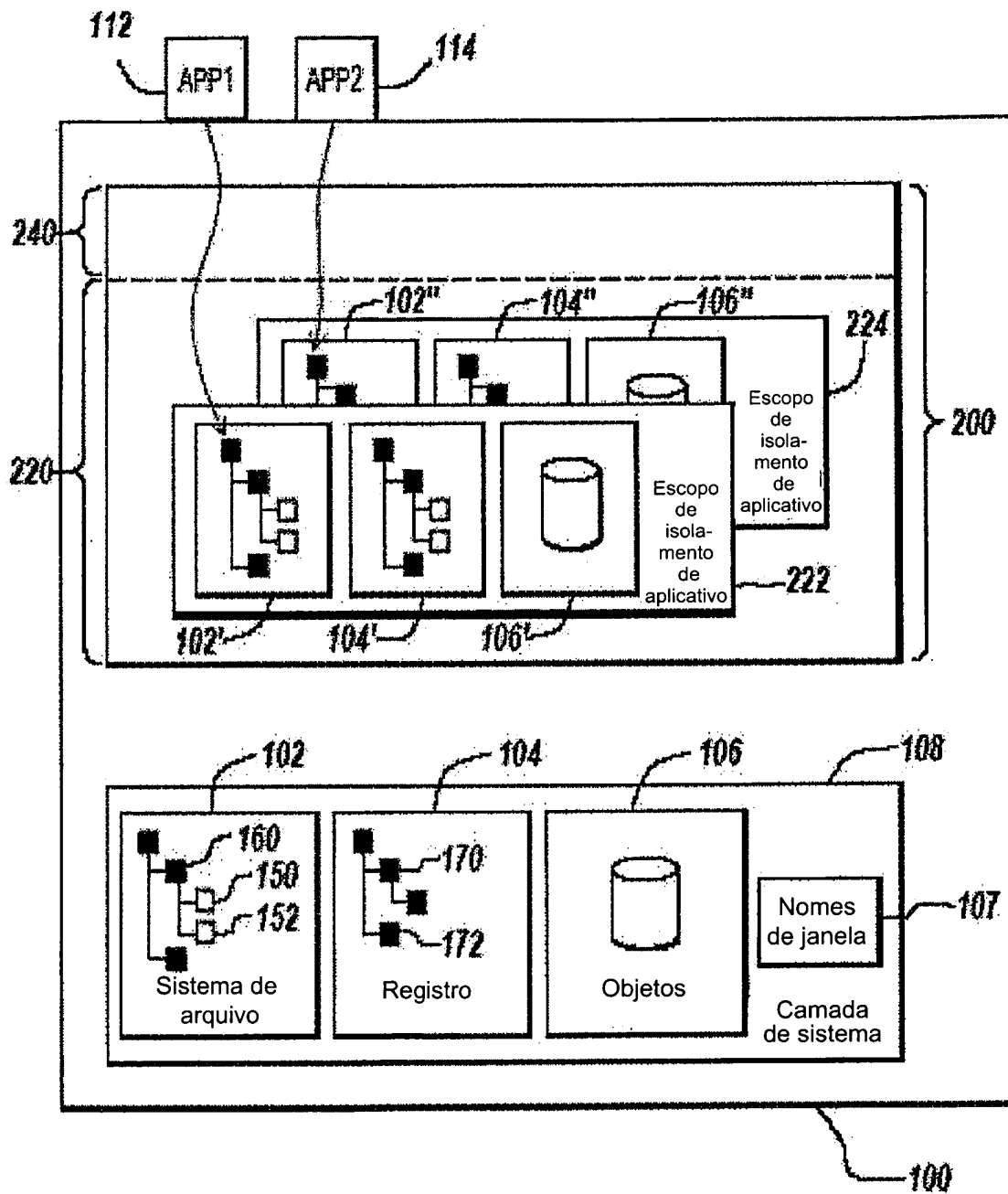


FIG. 8A

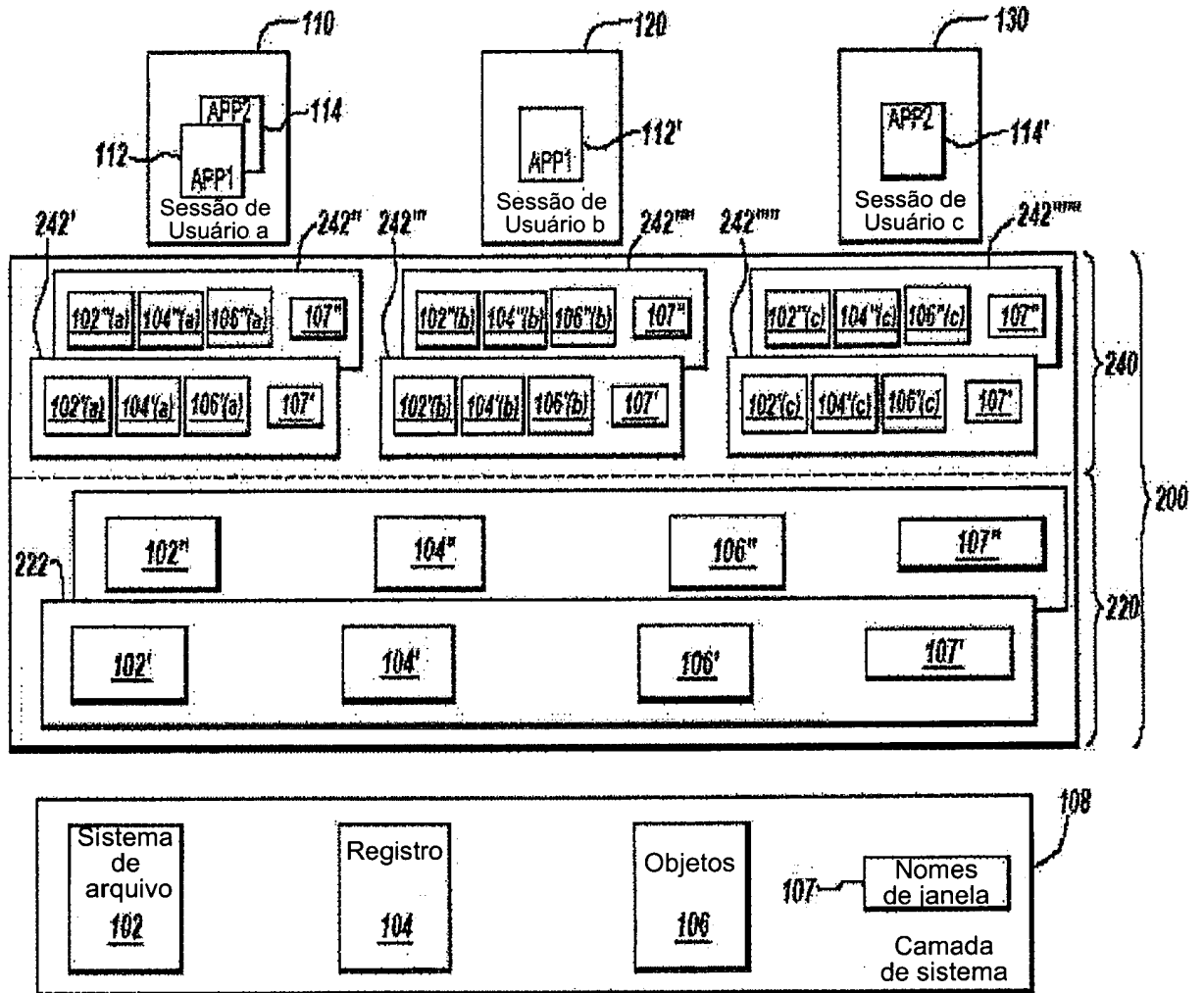


FIG. 8B

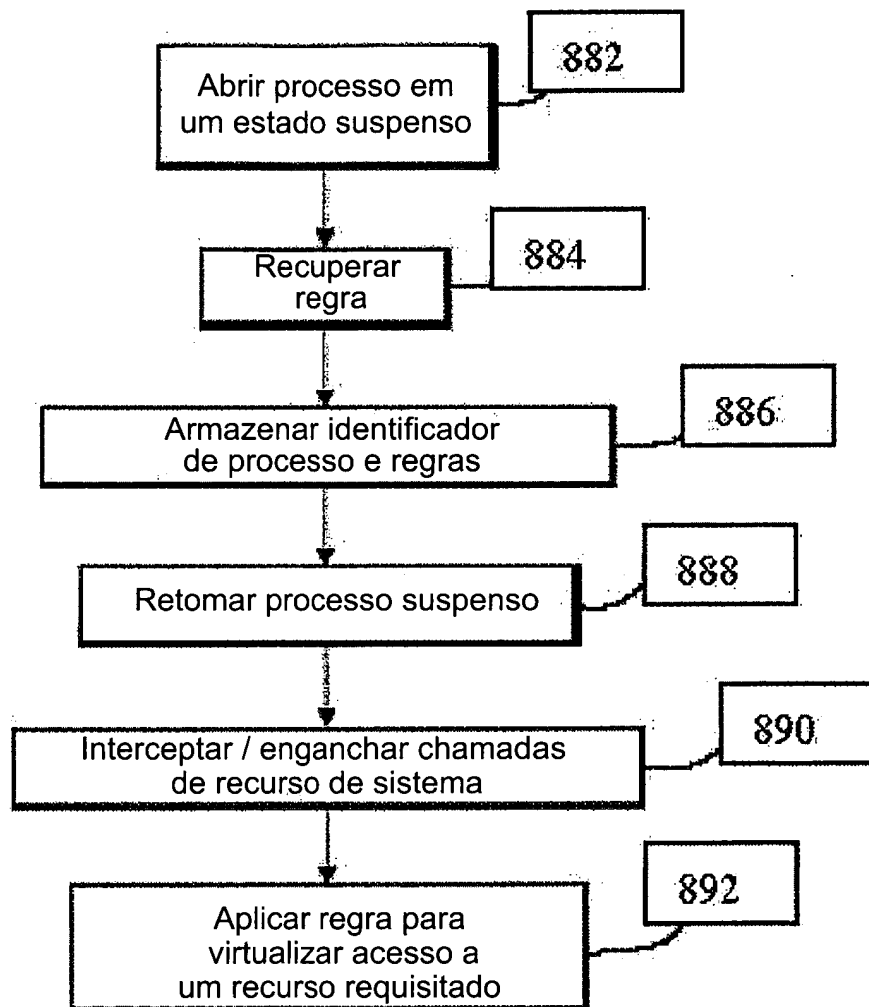
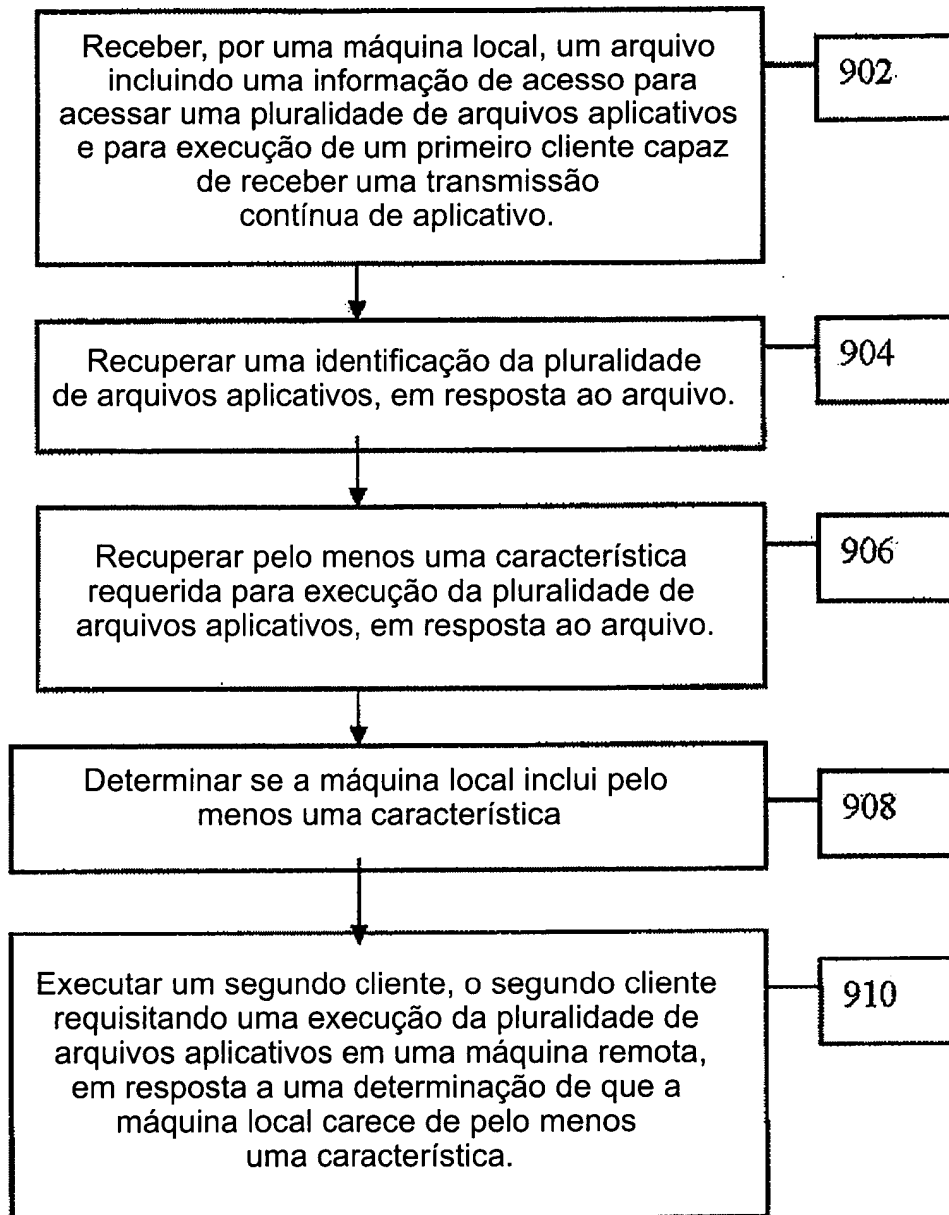


FIG. 8C

**FIG. 9**

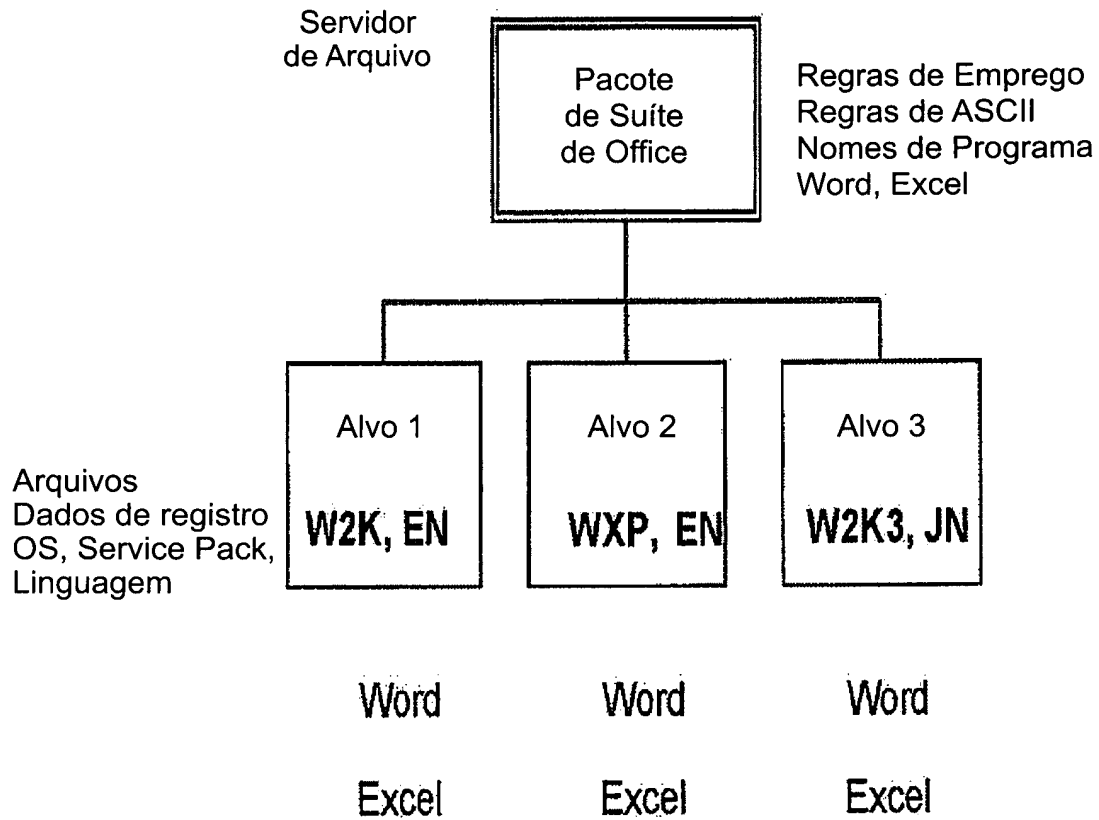
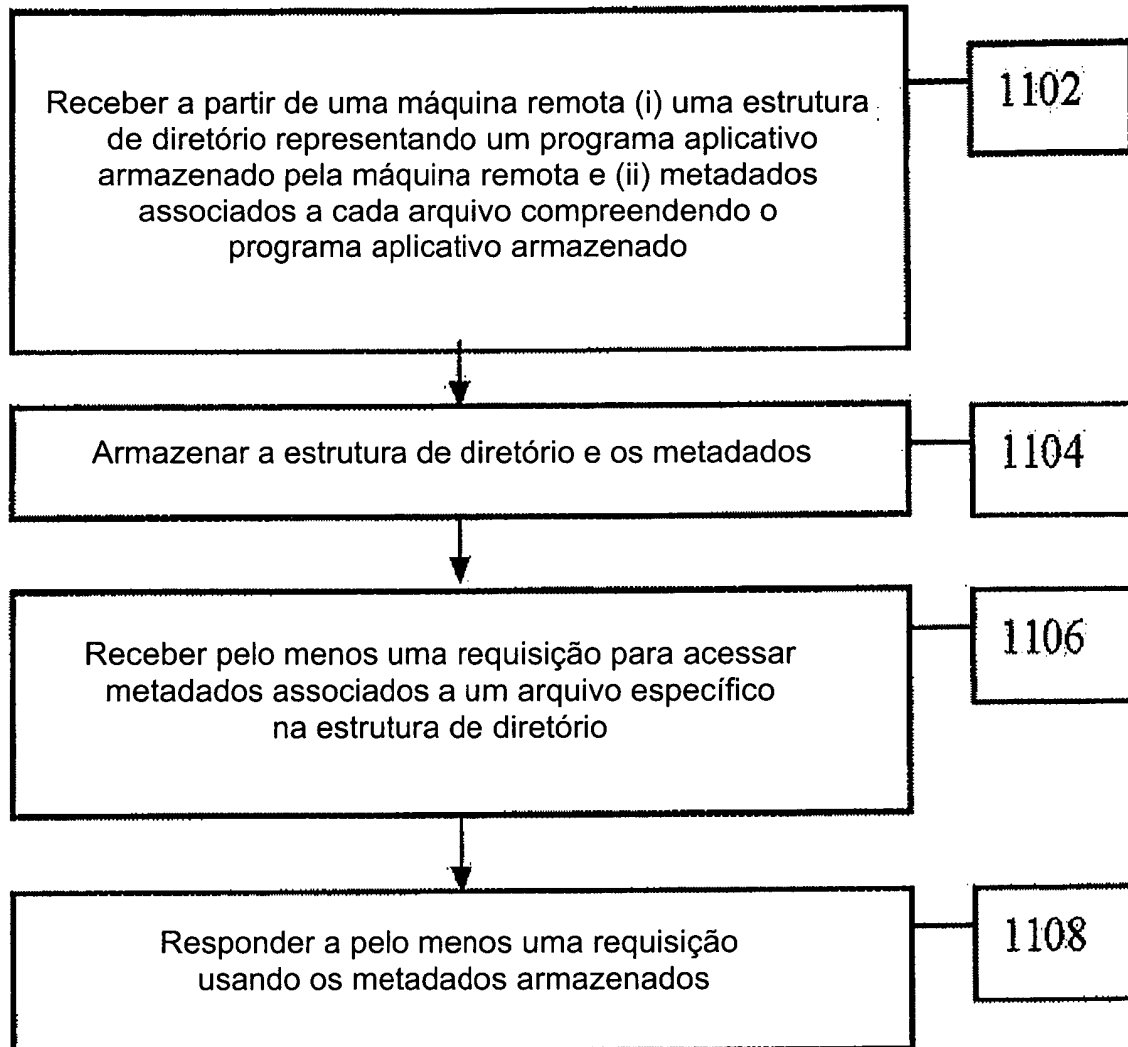
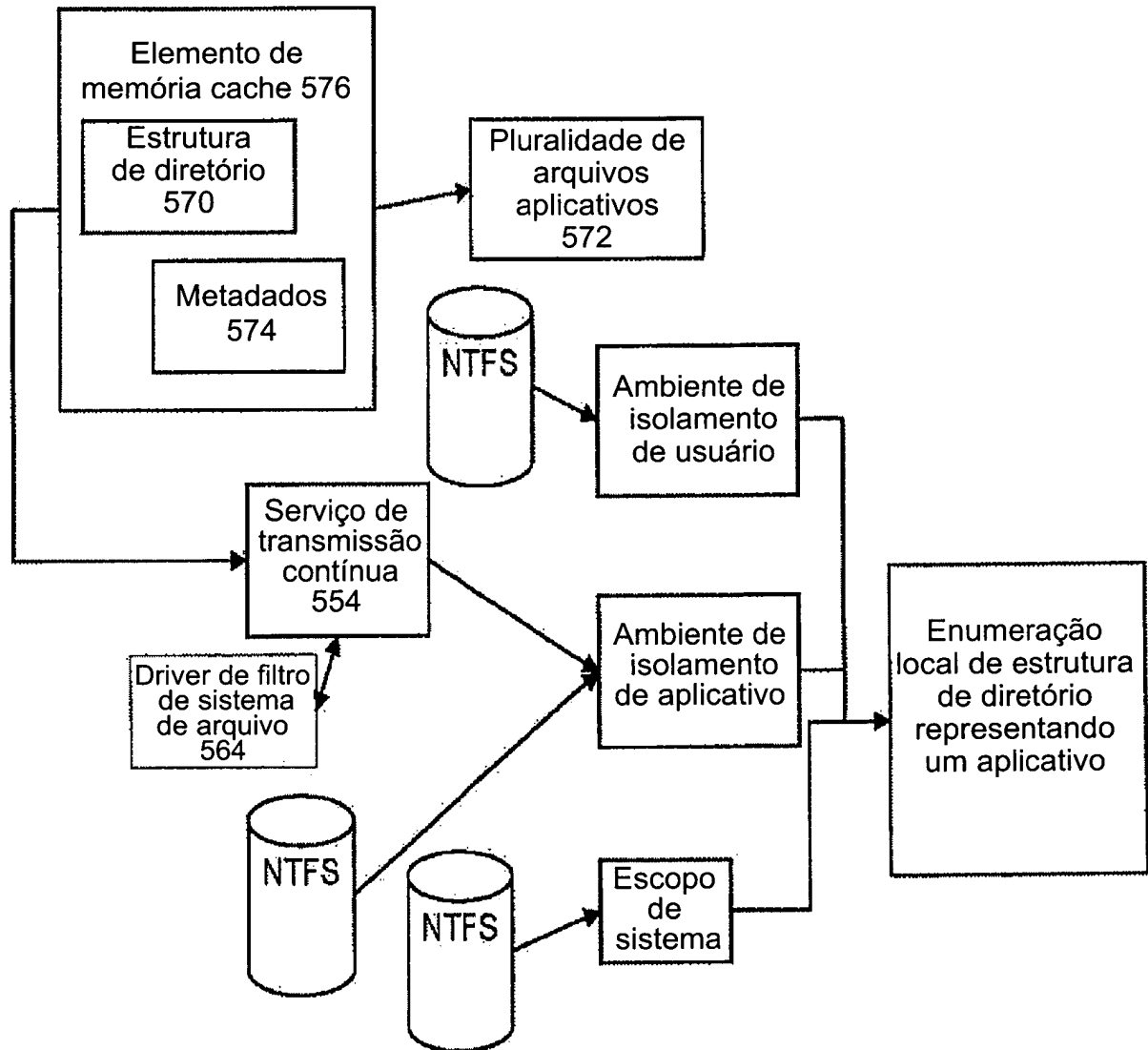
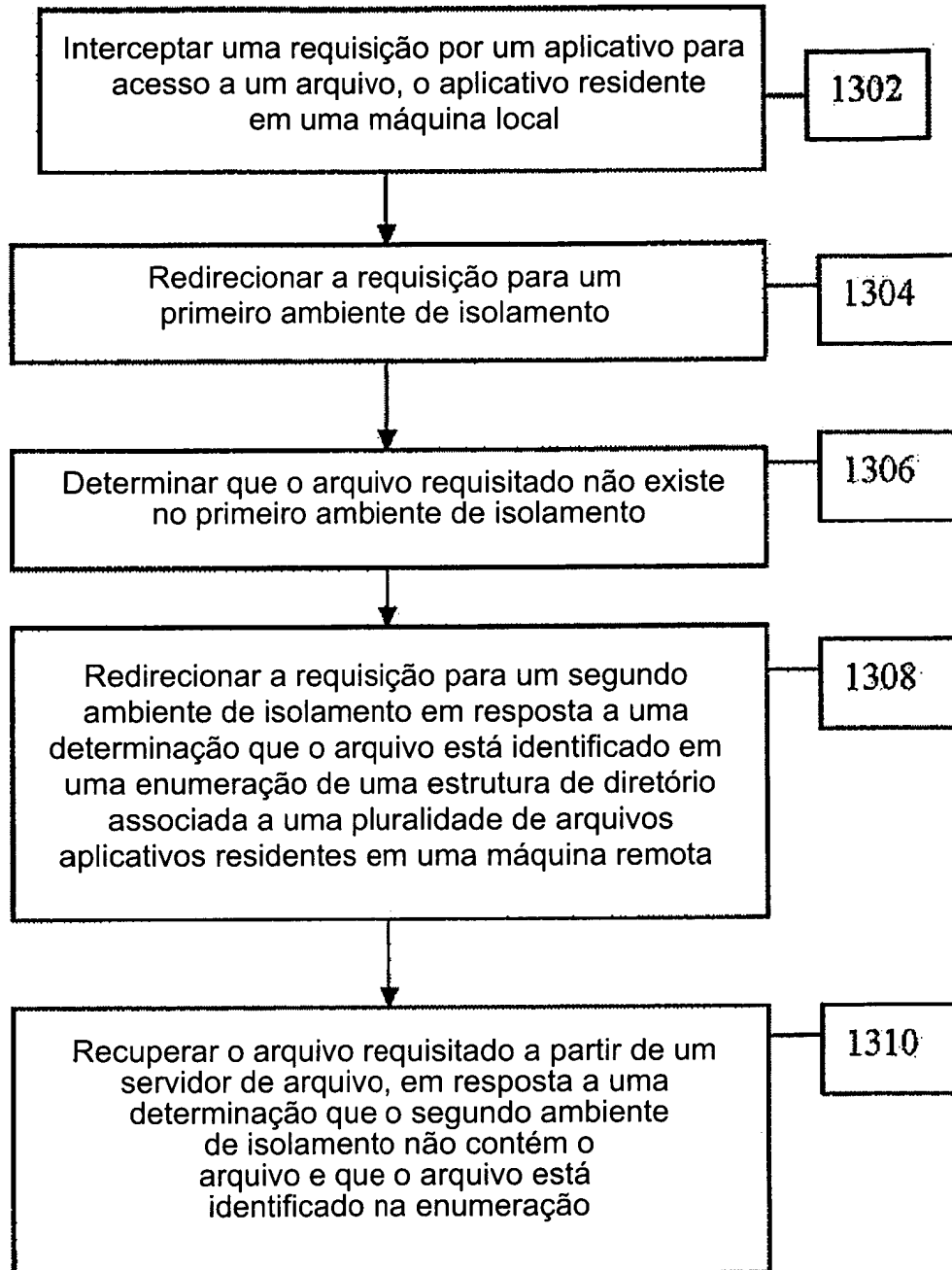
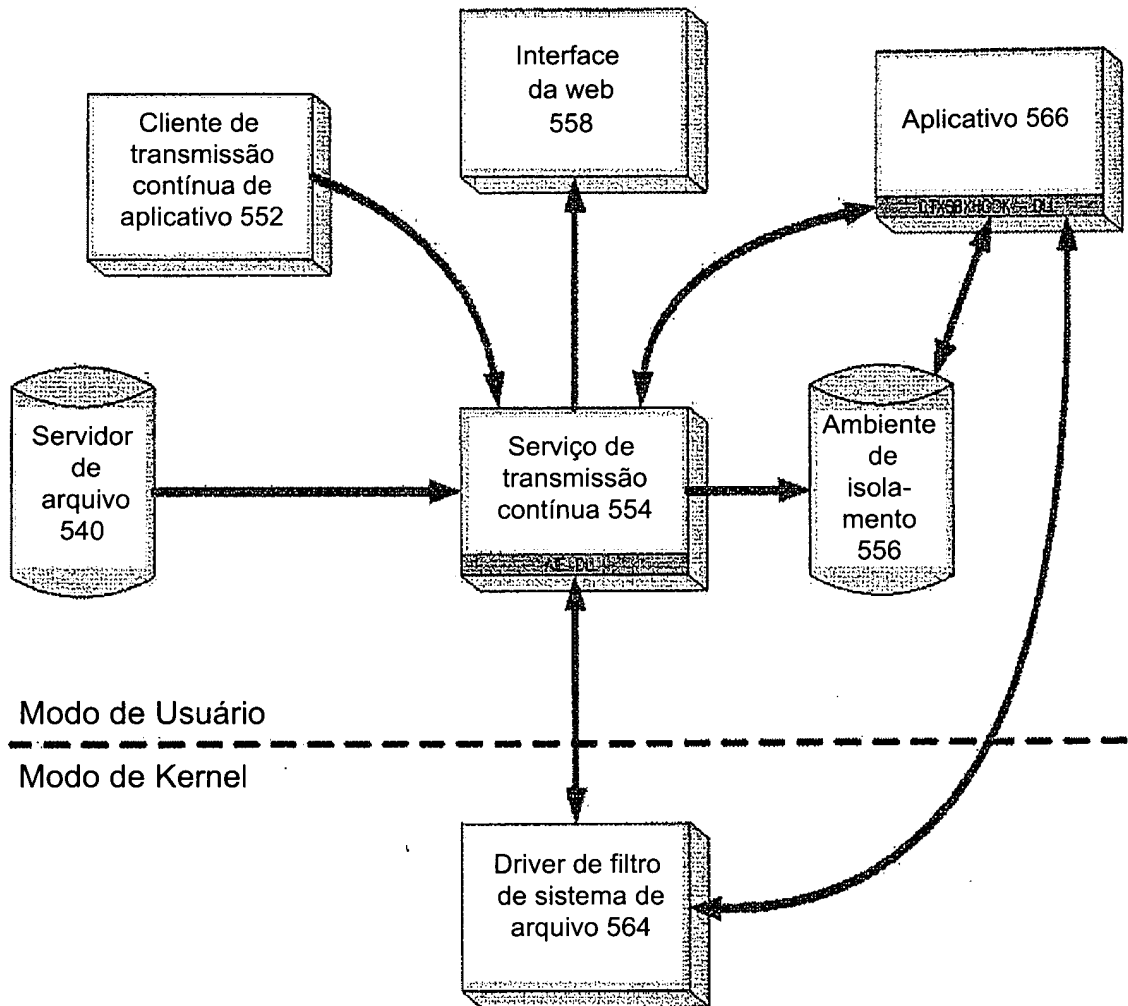


FIG. 10

**FIG. 11**

**FIG. 12**

**FIG. 13**

**FIG. 14**

Máquina remota 30

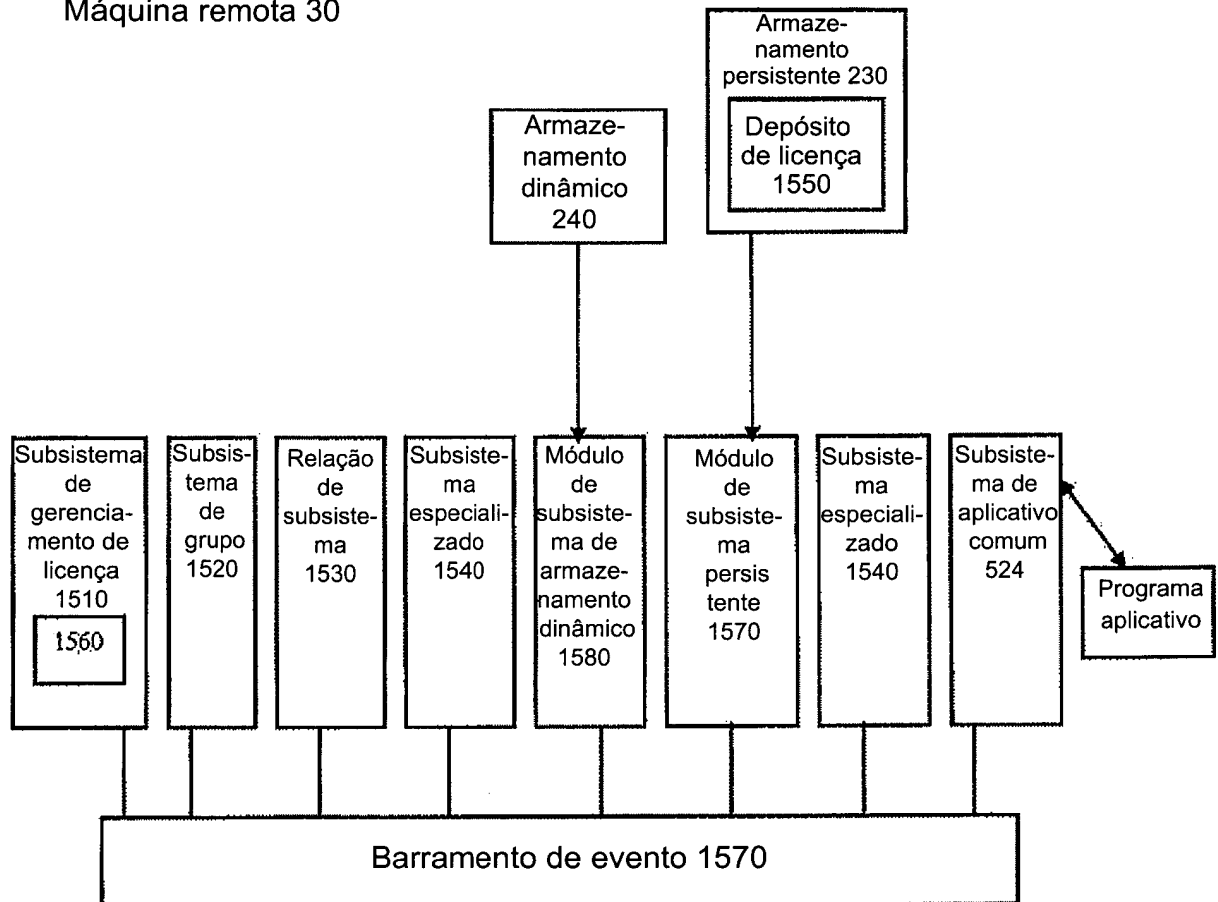


FIG. 15

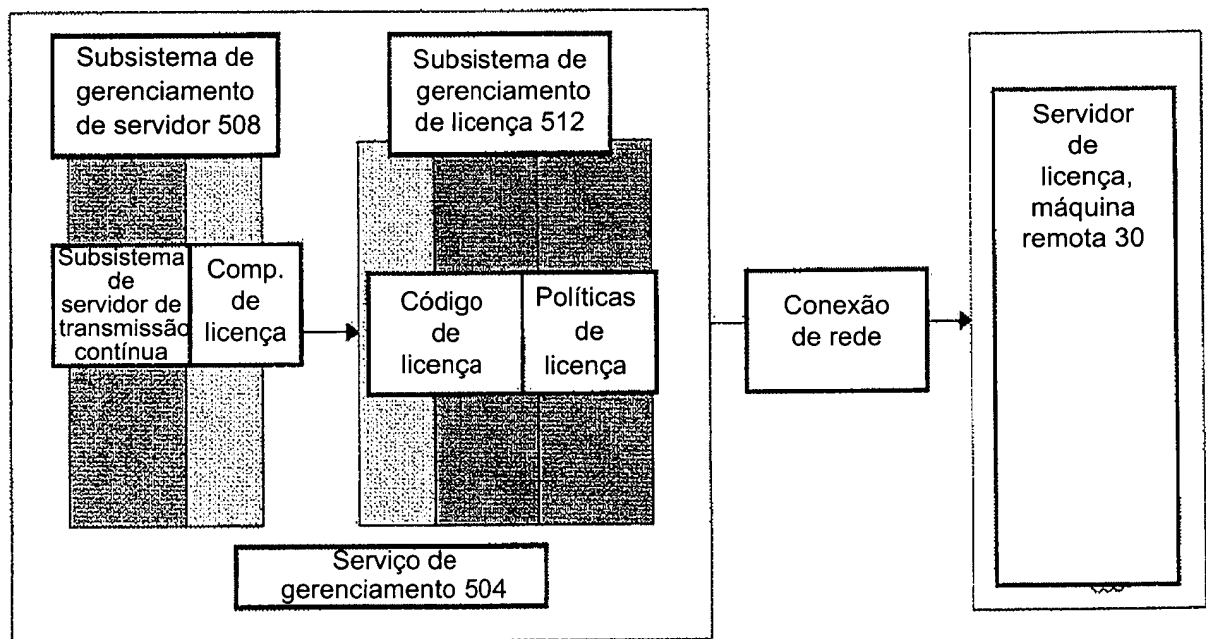


FIG. 16

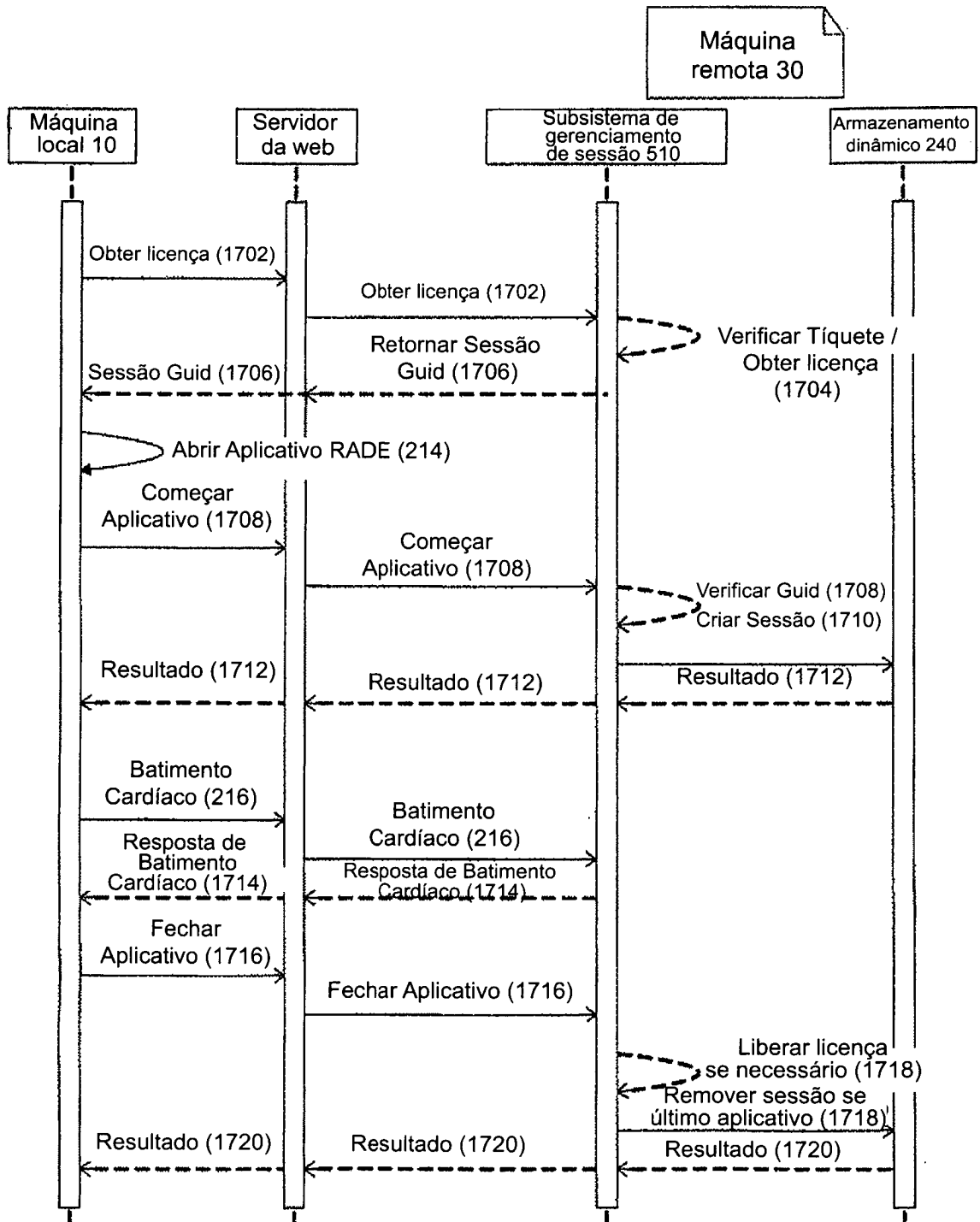
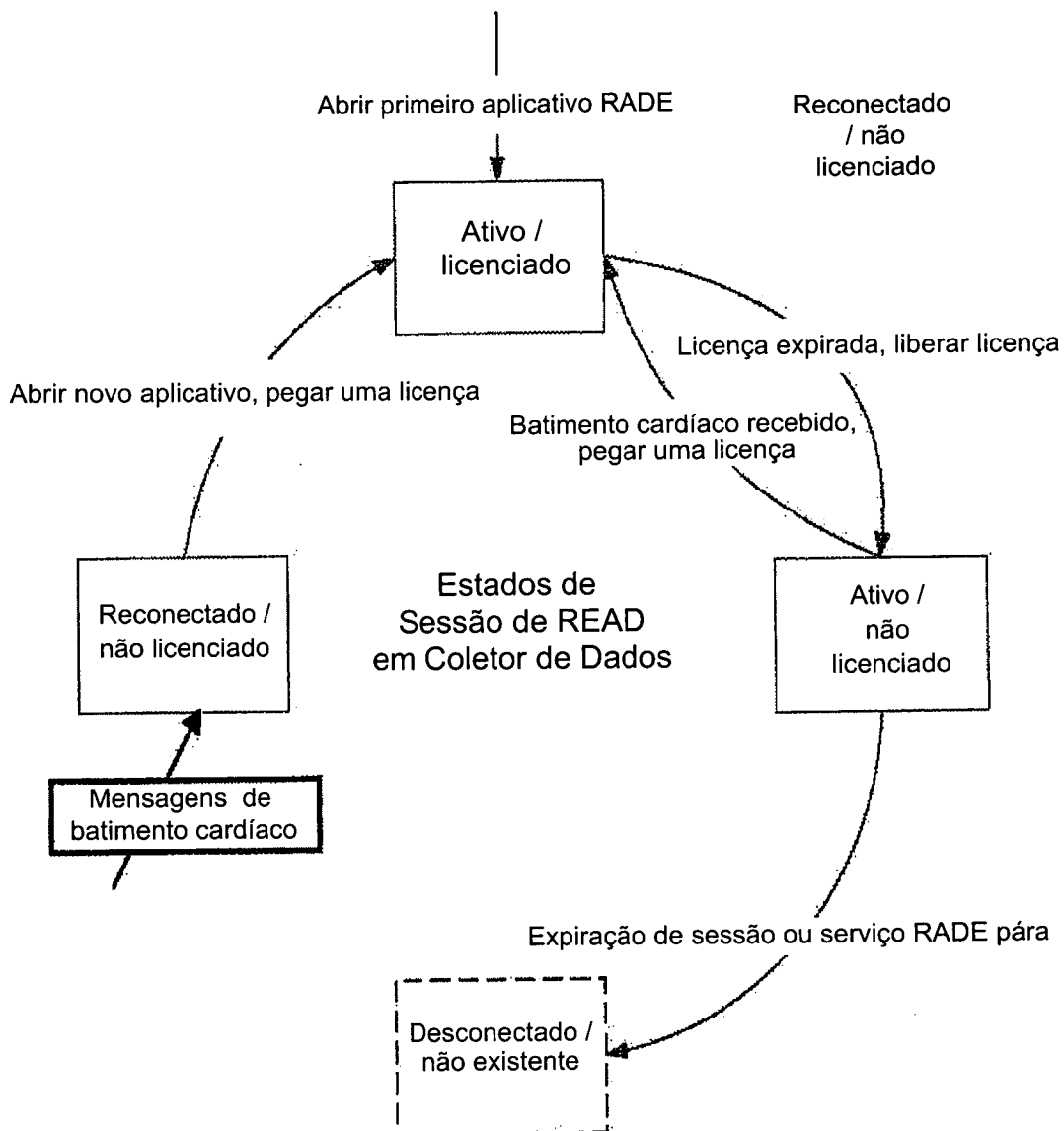


FIG. 17

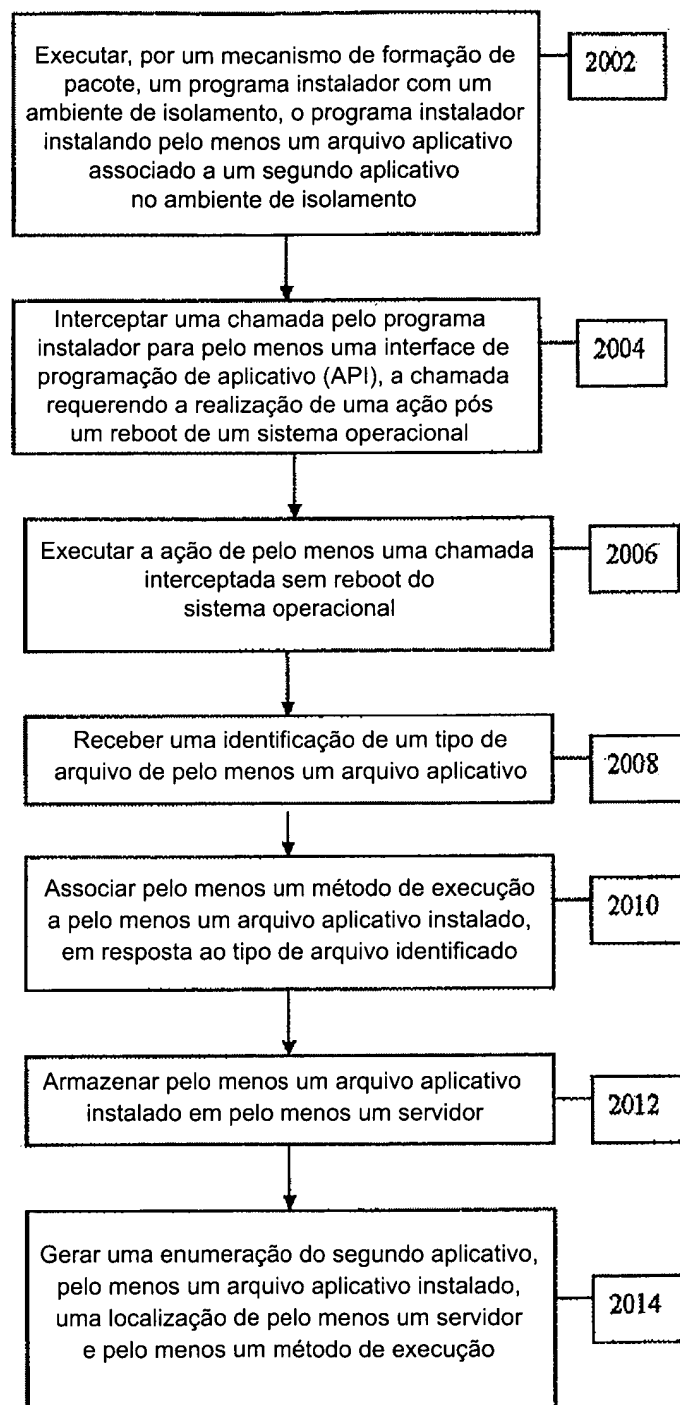
**FIG. 18**

PACOTE

Alvo 1	
Aplicativo:	Foo
OS Alvo:	W2K PRO (Todos os Service Packs) WXP PRO (Todos os Service Packs) W2K3 PRO (Todos os Service Packs)
Linguagem Alvo:	Inglês
Unidade Mestre:	C:\

Alvo 1	
Aplicativo:	Foo
OS Alvo:	W2K PRO (Todos os Service Packs) WXP PRO (Todos os Service Packs) W2K3 PRO (Todos os Service Packs)
Linguagem Alvo:	Alemão
Unidade Mestre:	C:\

FIG. 19

**FIG. 20**

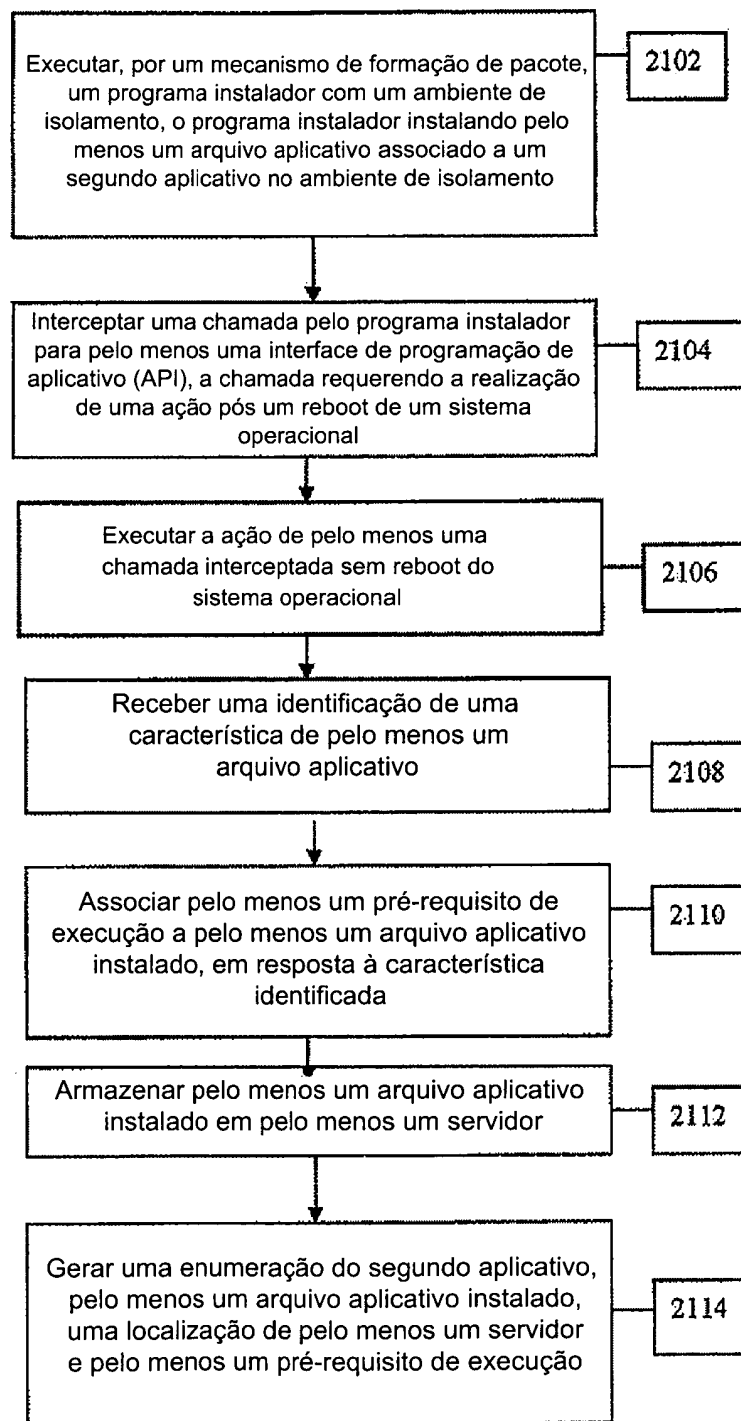


FIG. 21

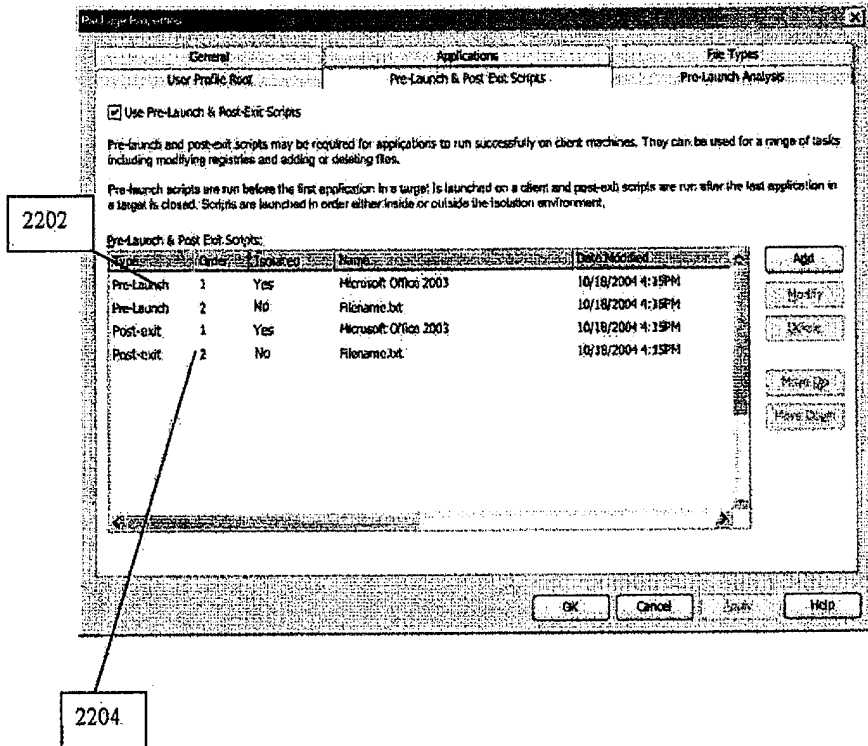


FIG. 22

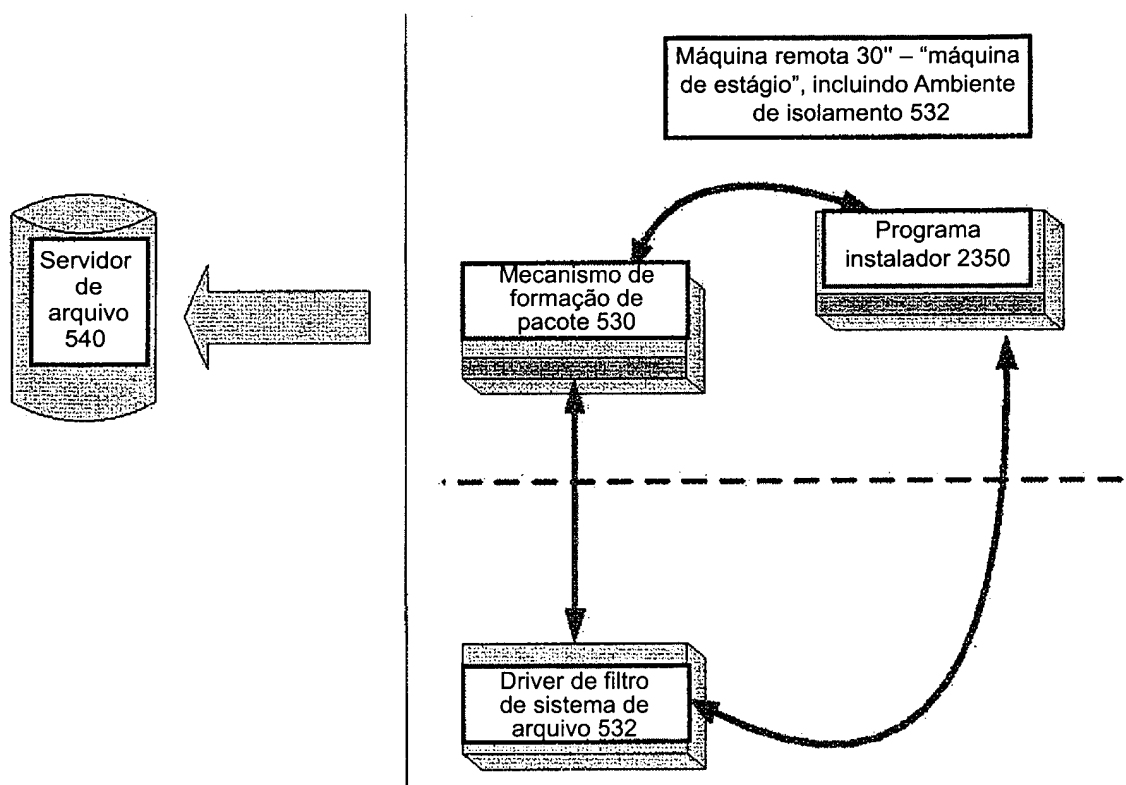
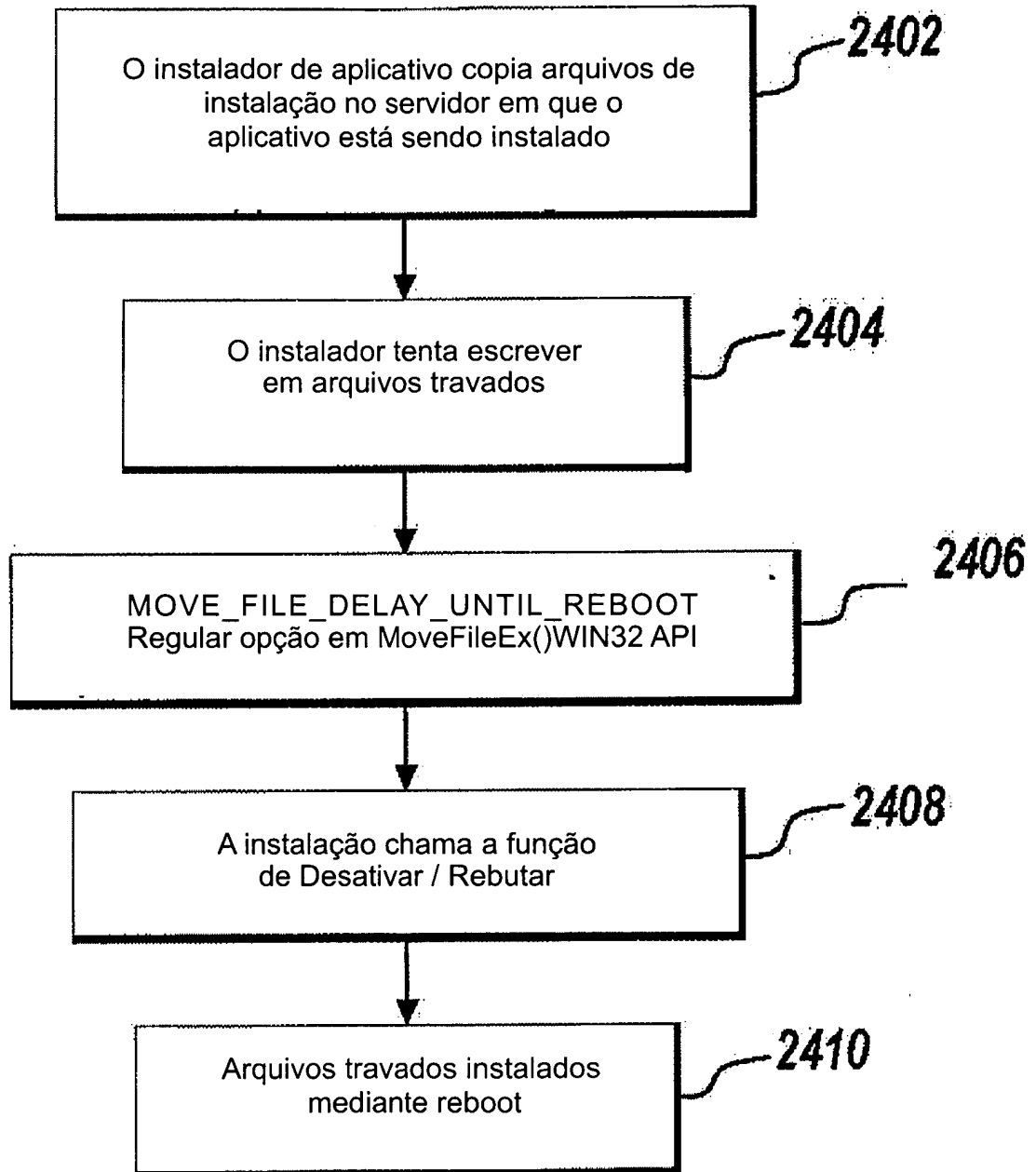


FIG. 23

**FIG. 24**

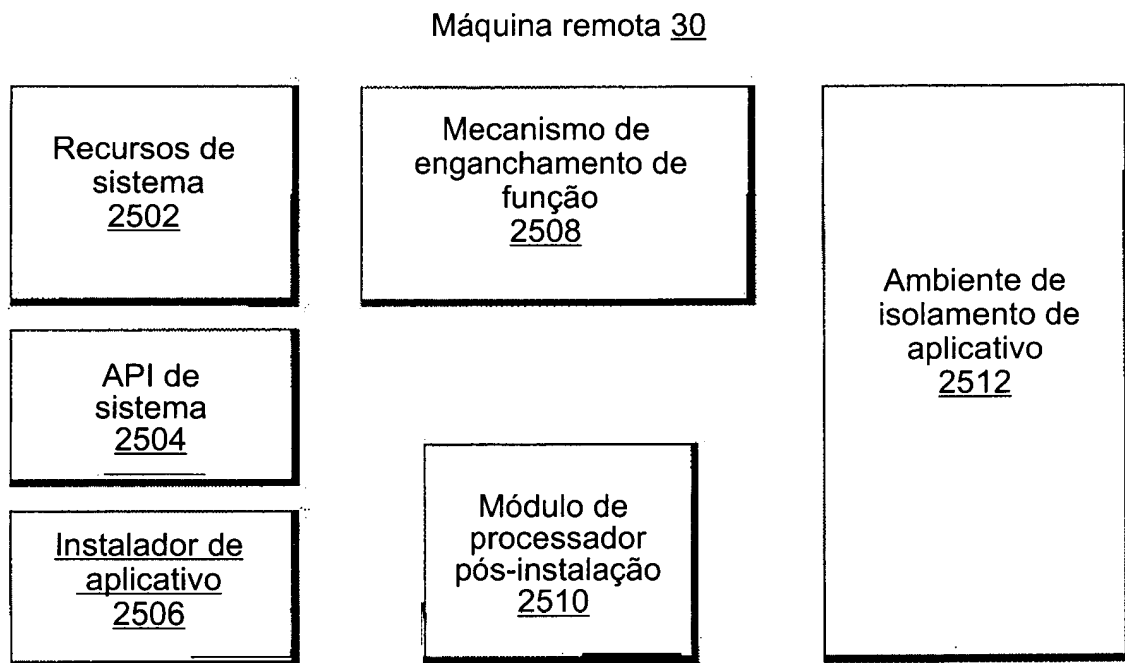


FIG. 25

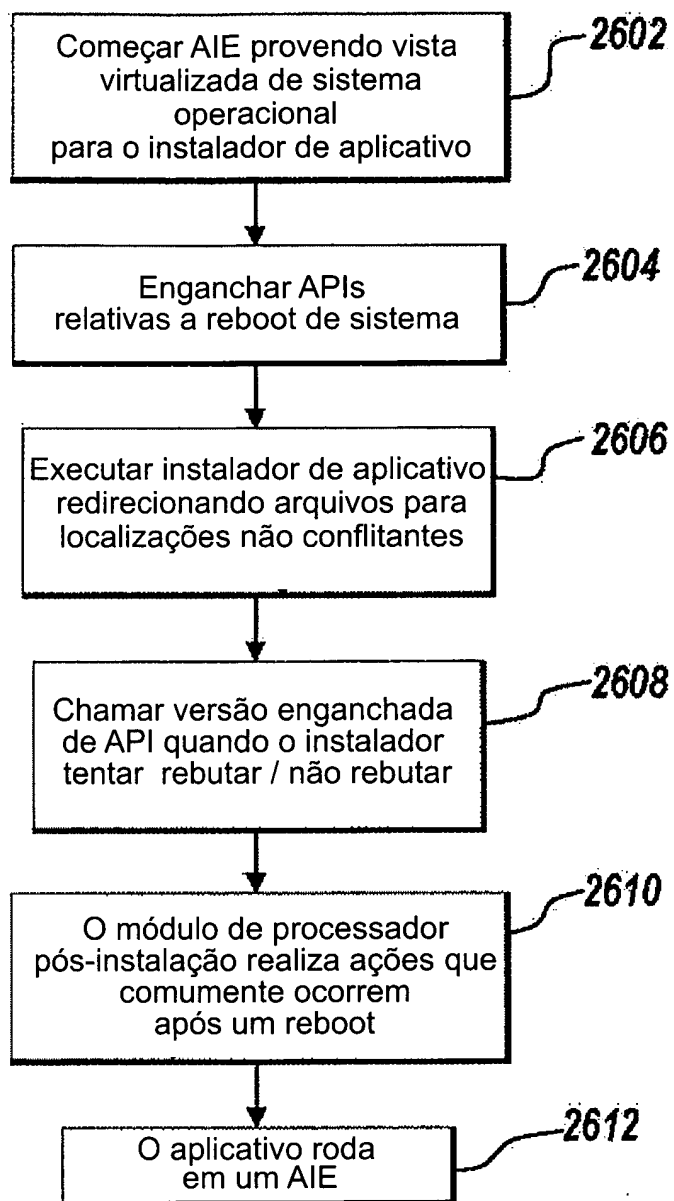


FIG. 26

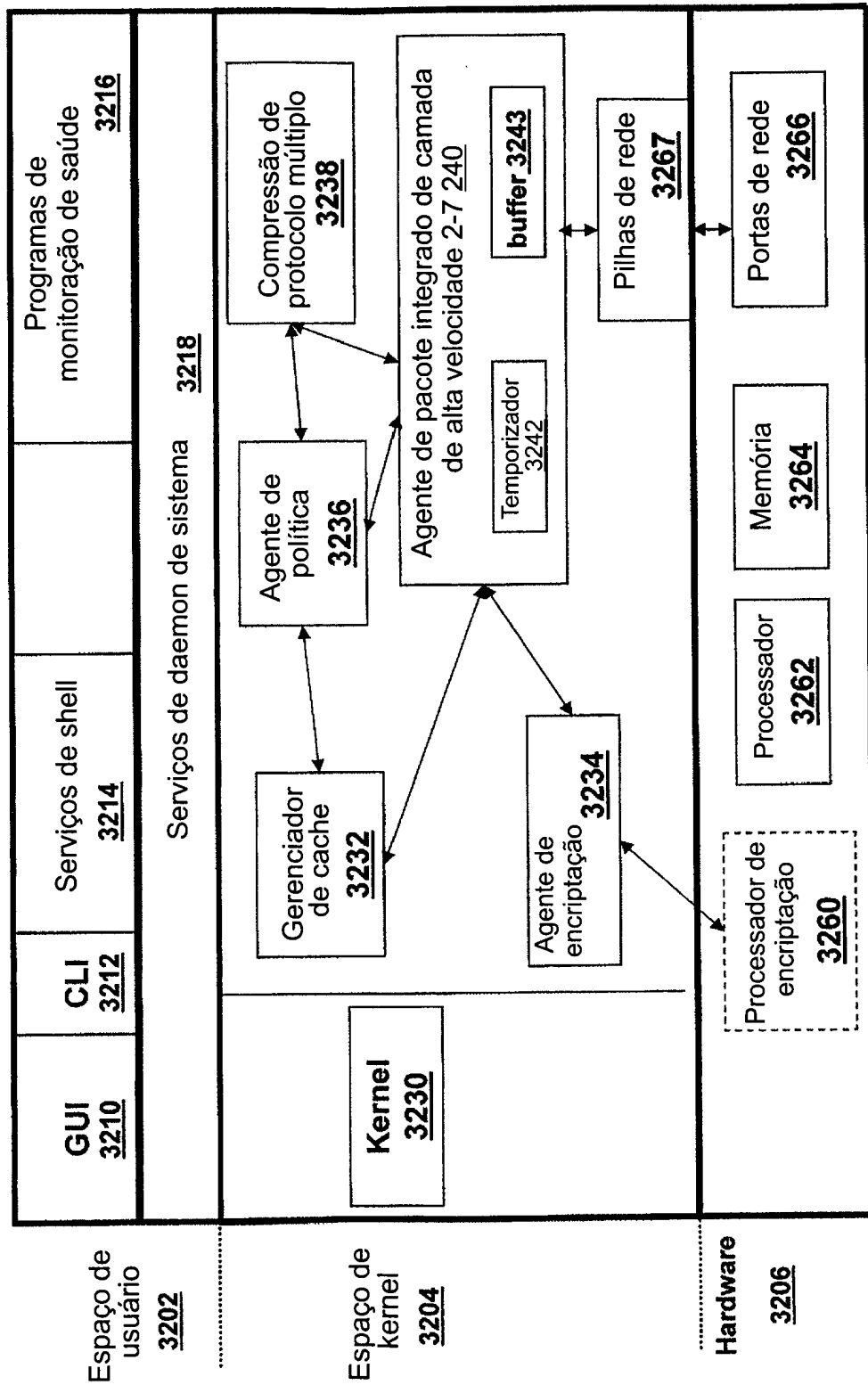
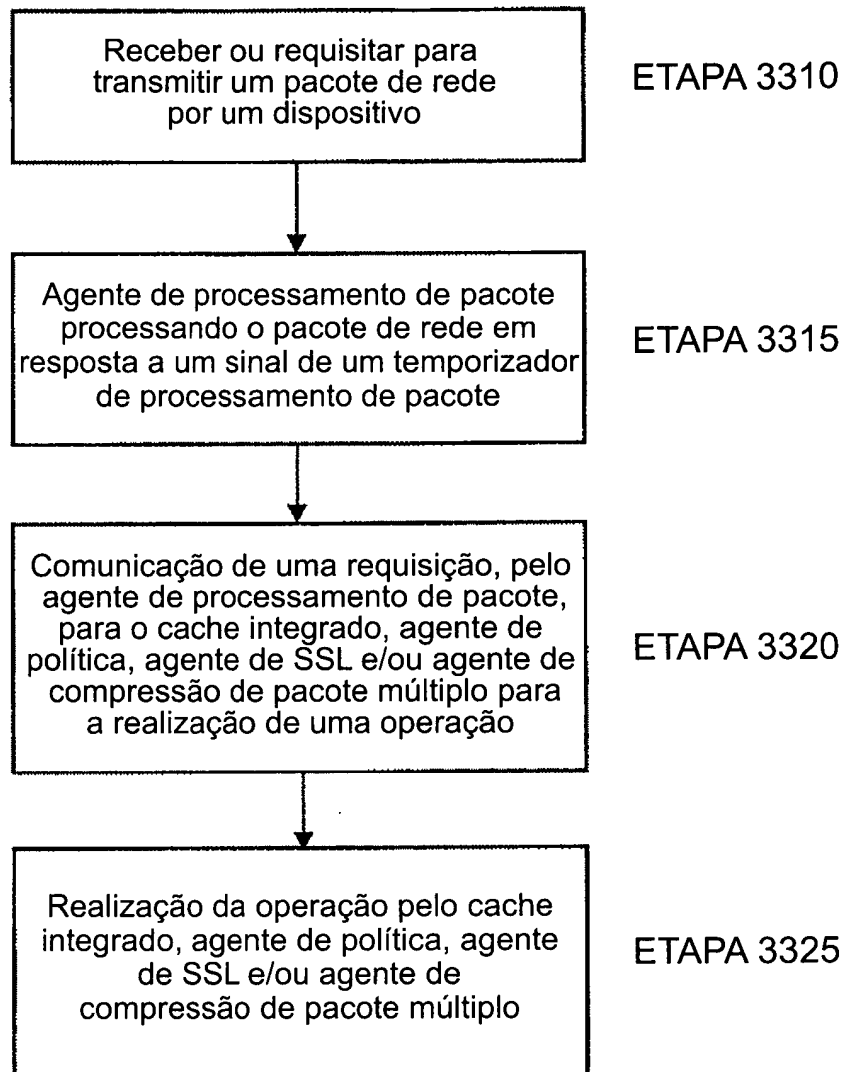
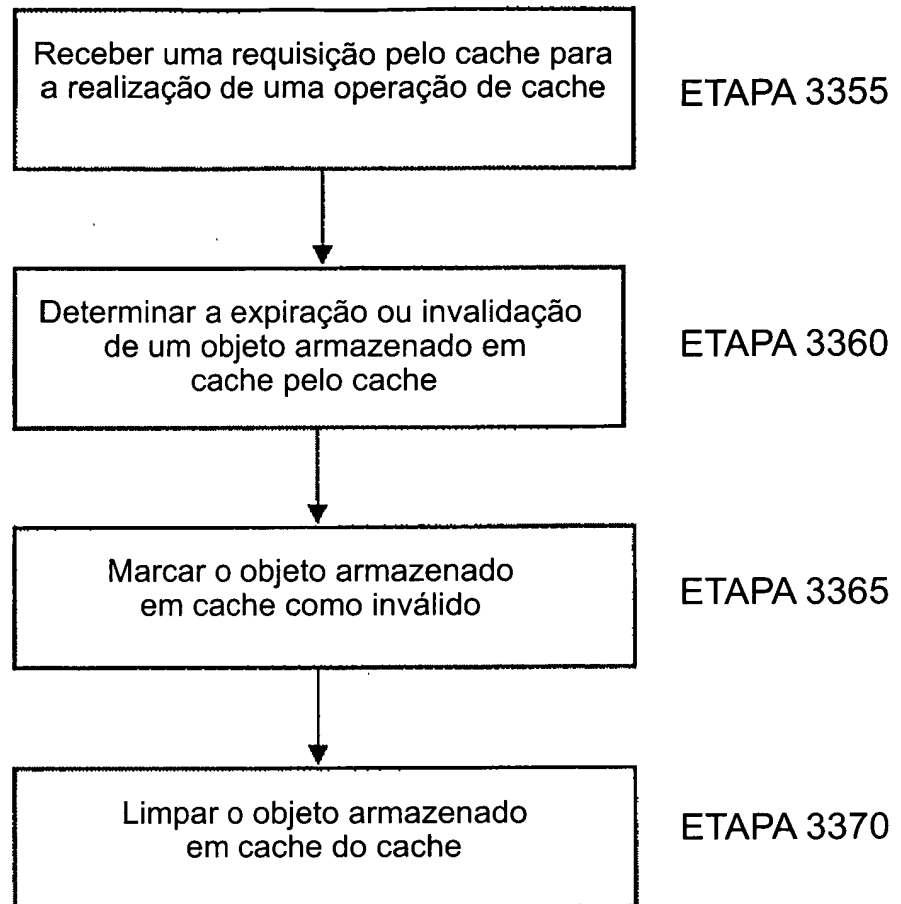


FIG. 27



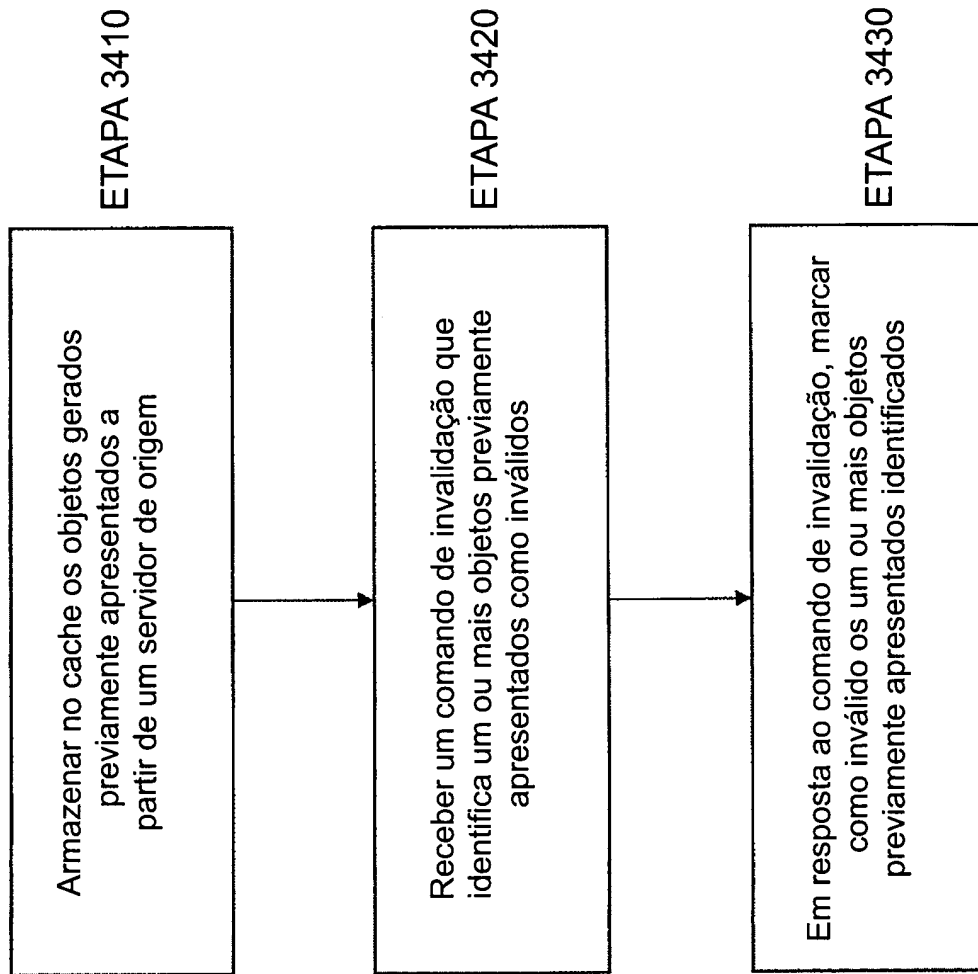
3300 ✓

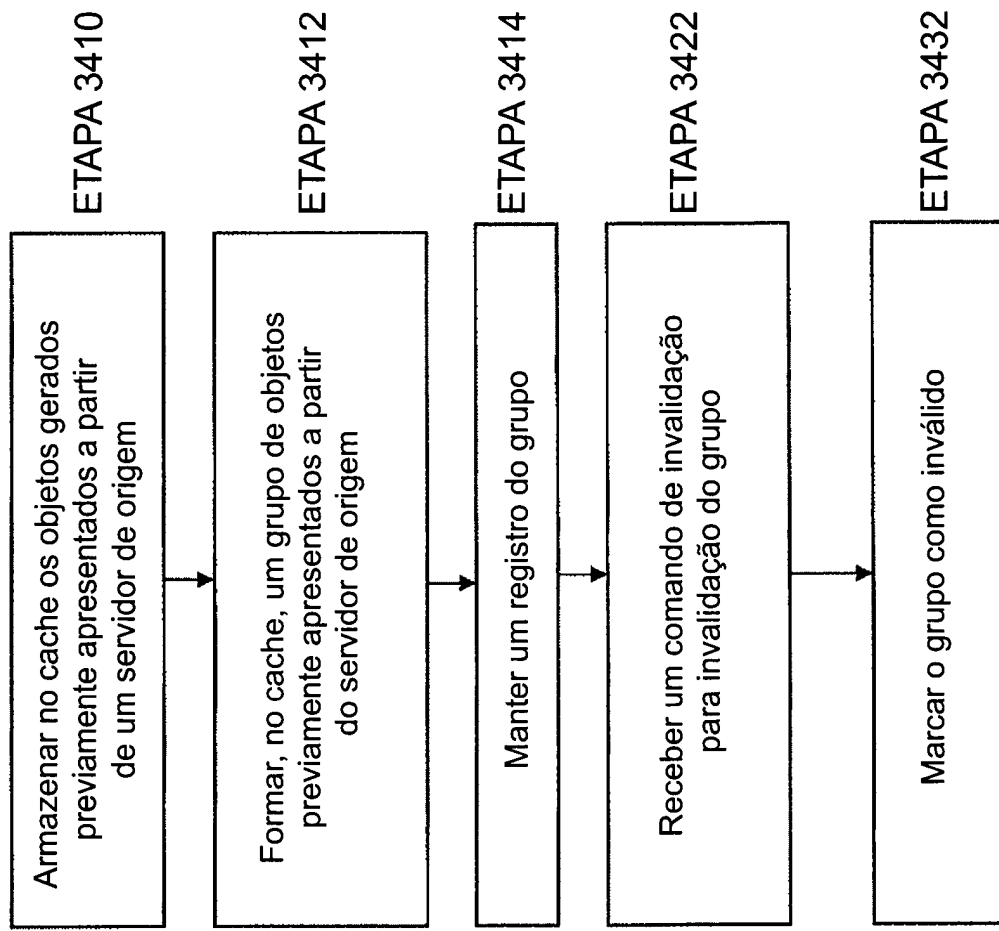
FIG. 28A



3350

FIG. 28B

**FIG. 29A**

**FIG. 29B**

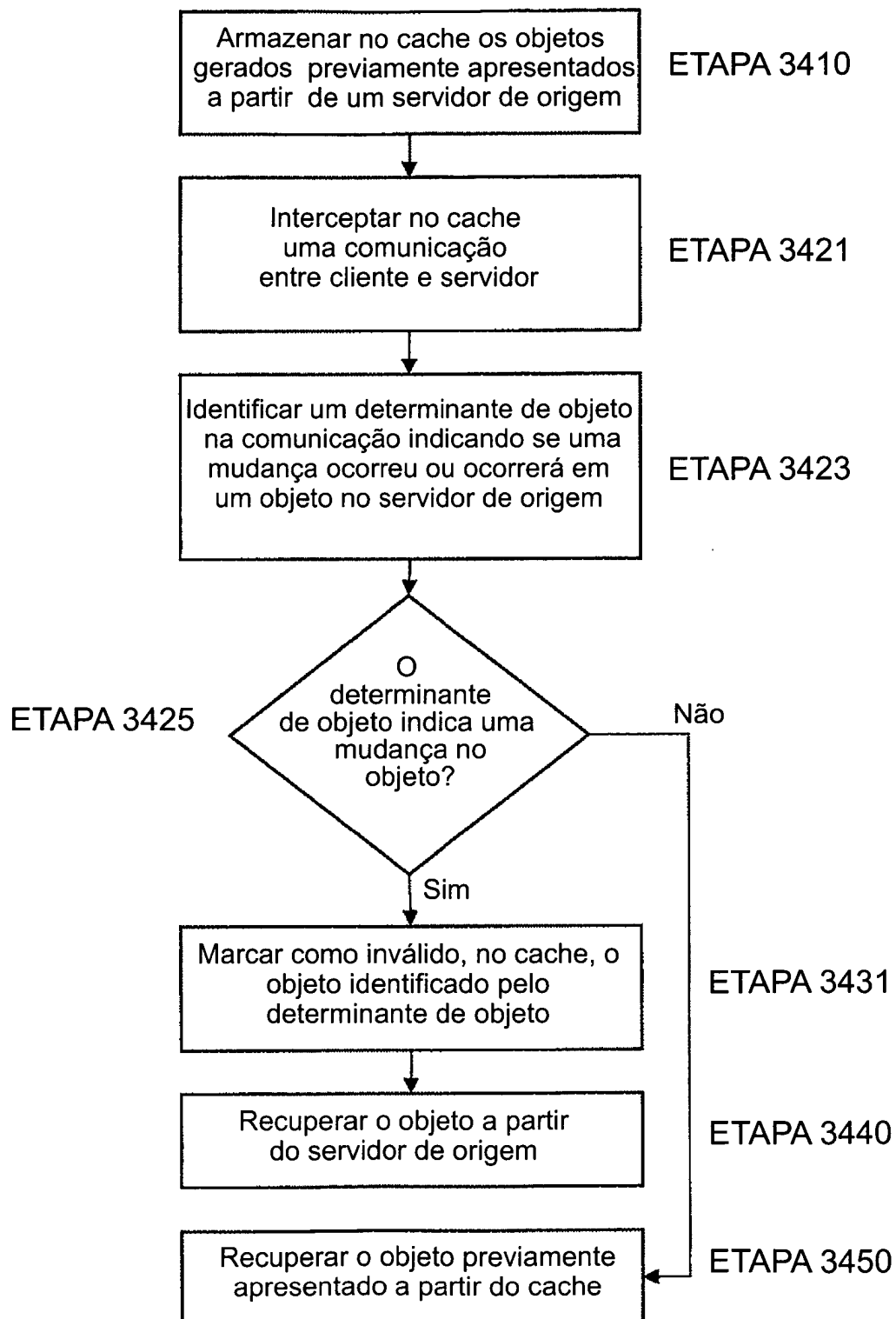


FIG. 29C

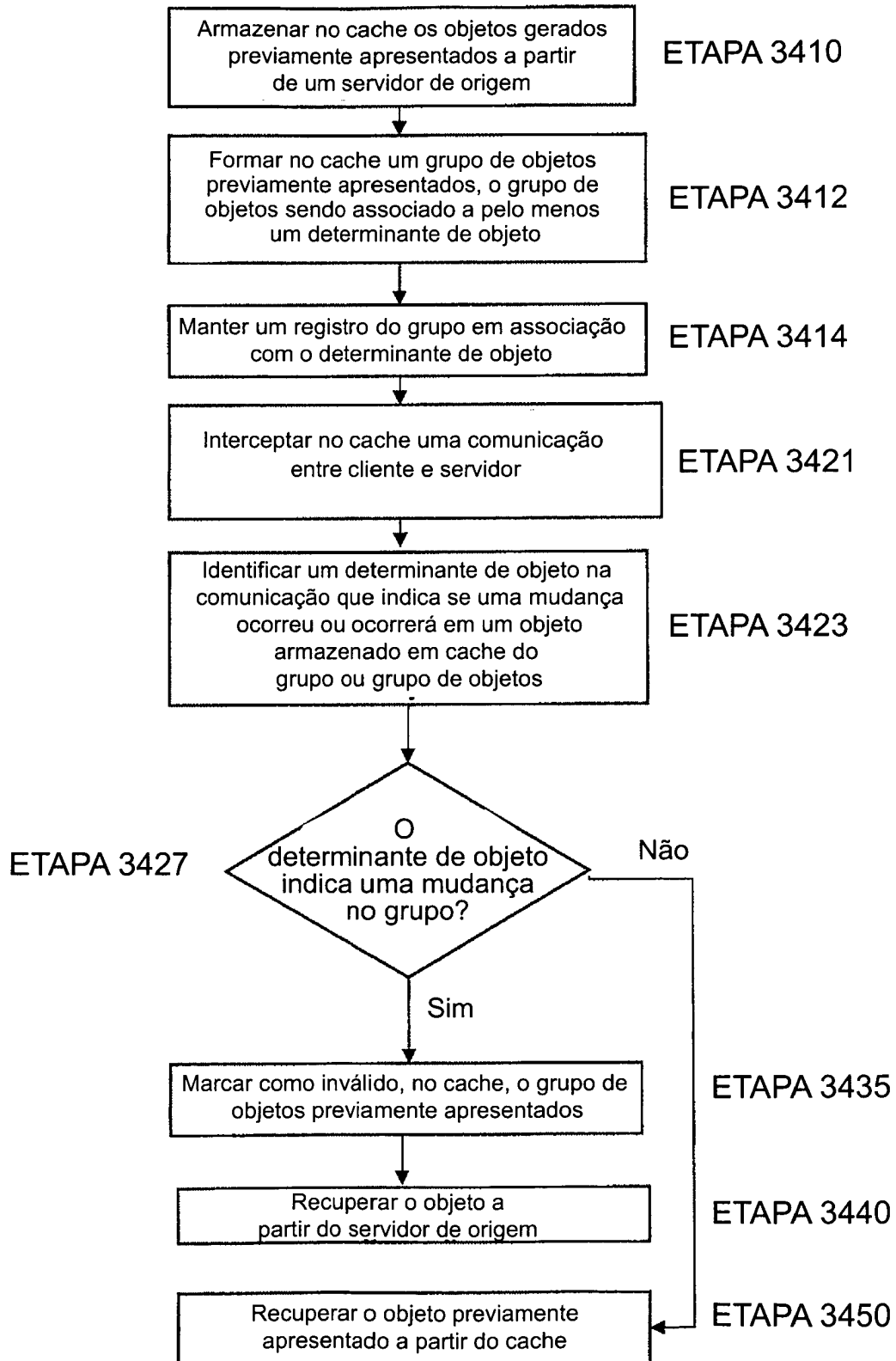
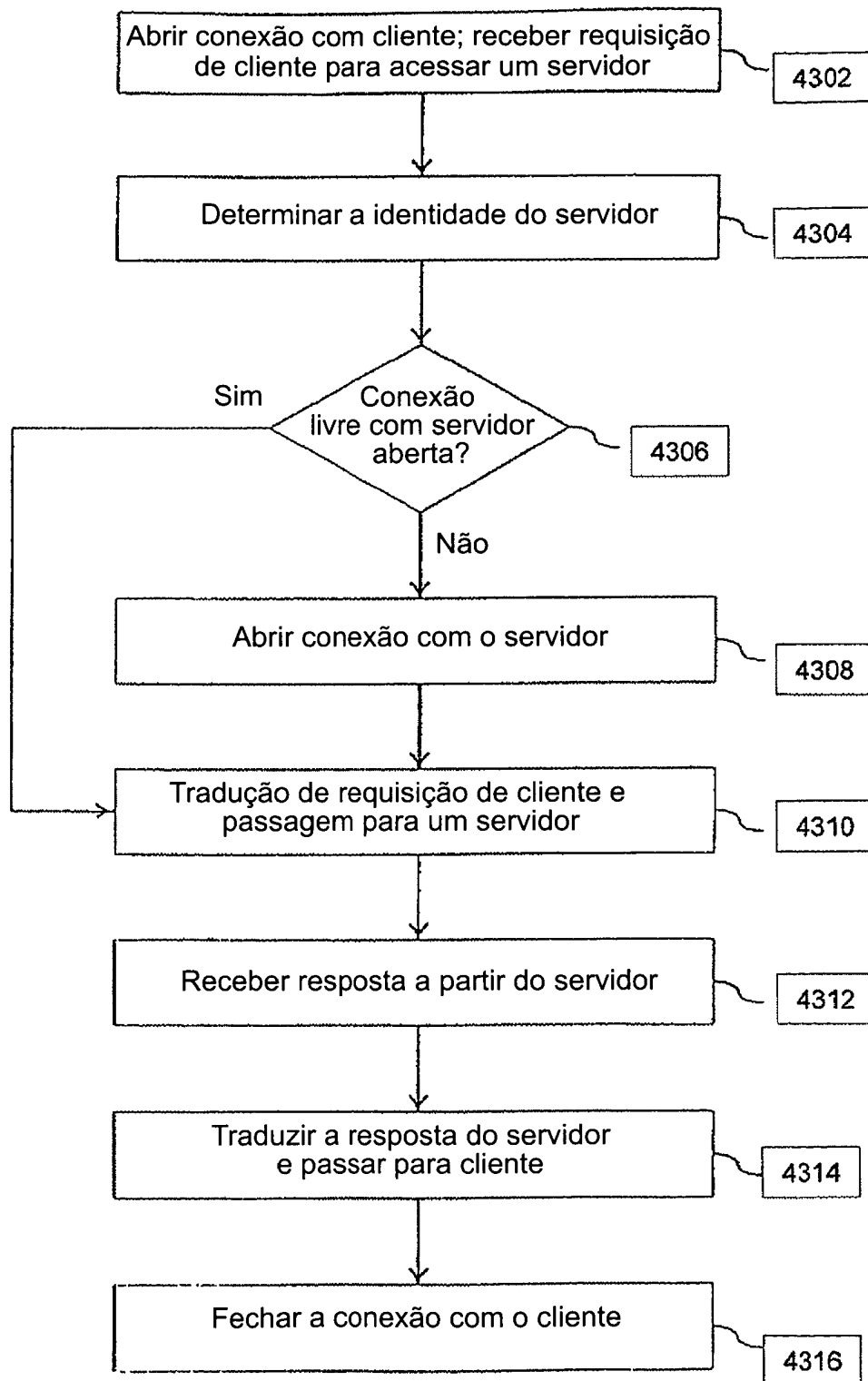
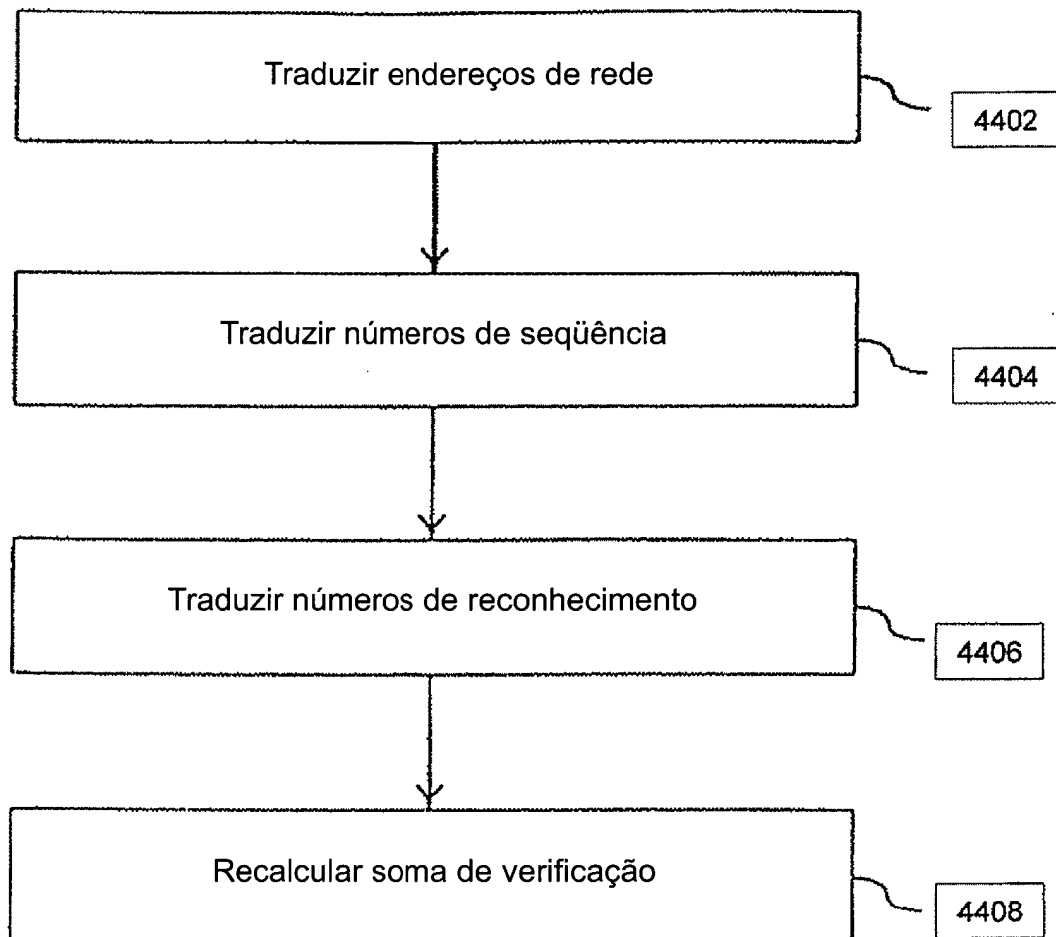


FIG. 29D

**FIG. 30**

**FIG. 31**

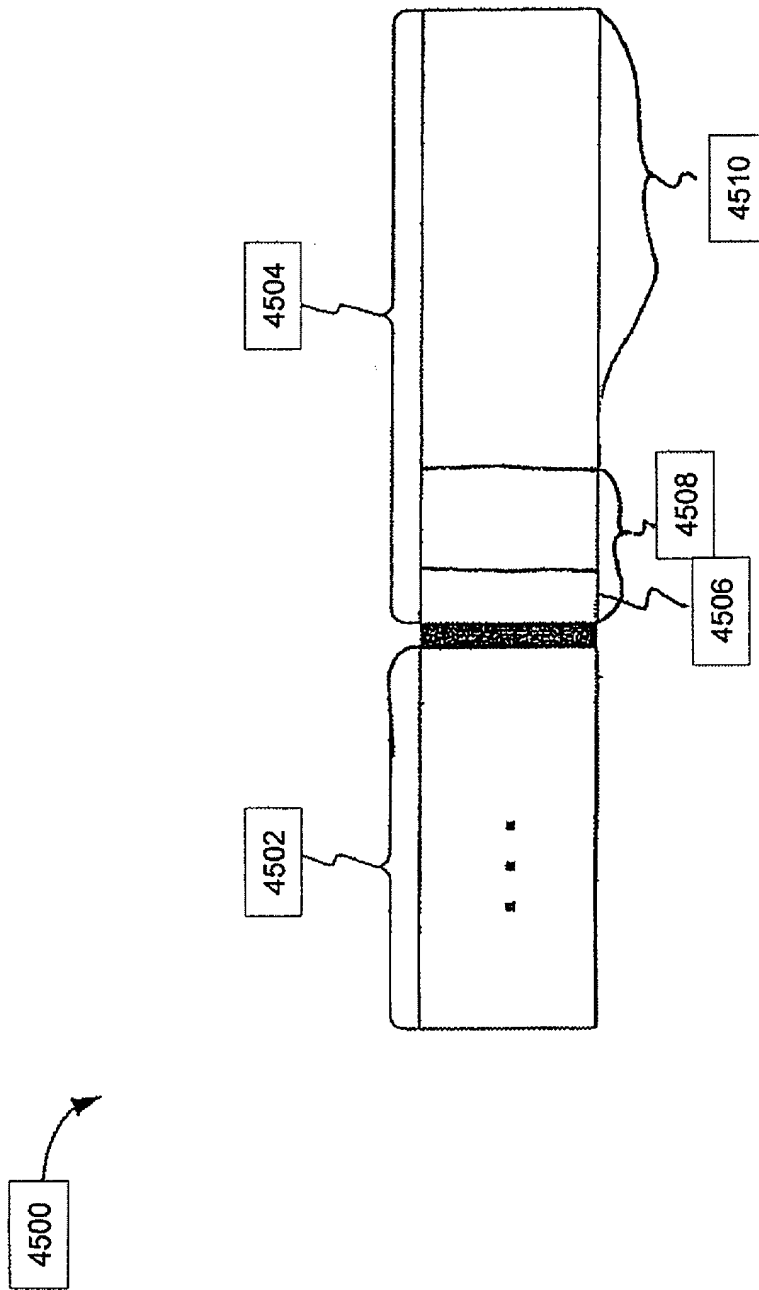


FIG. 32

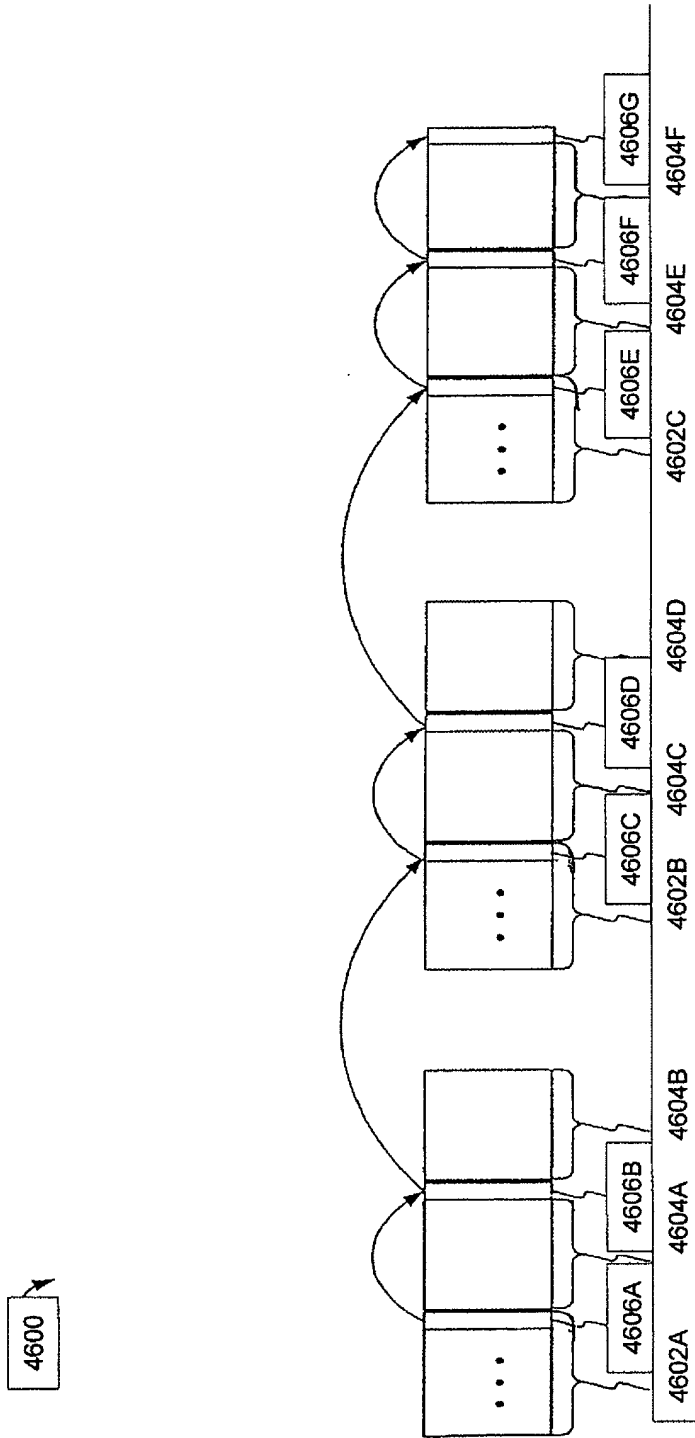


FIG. 33

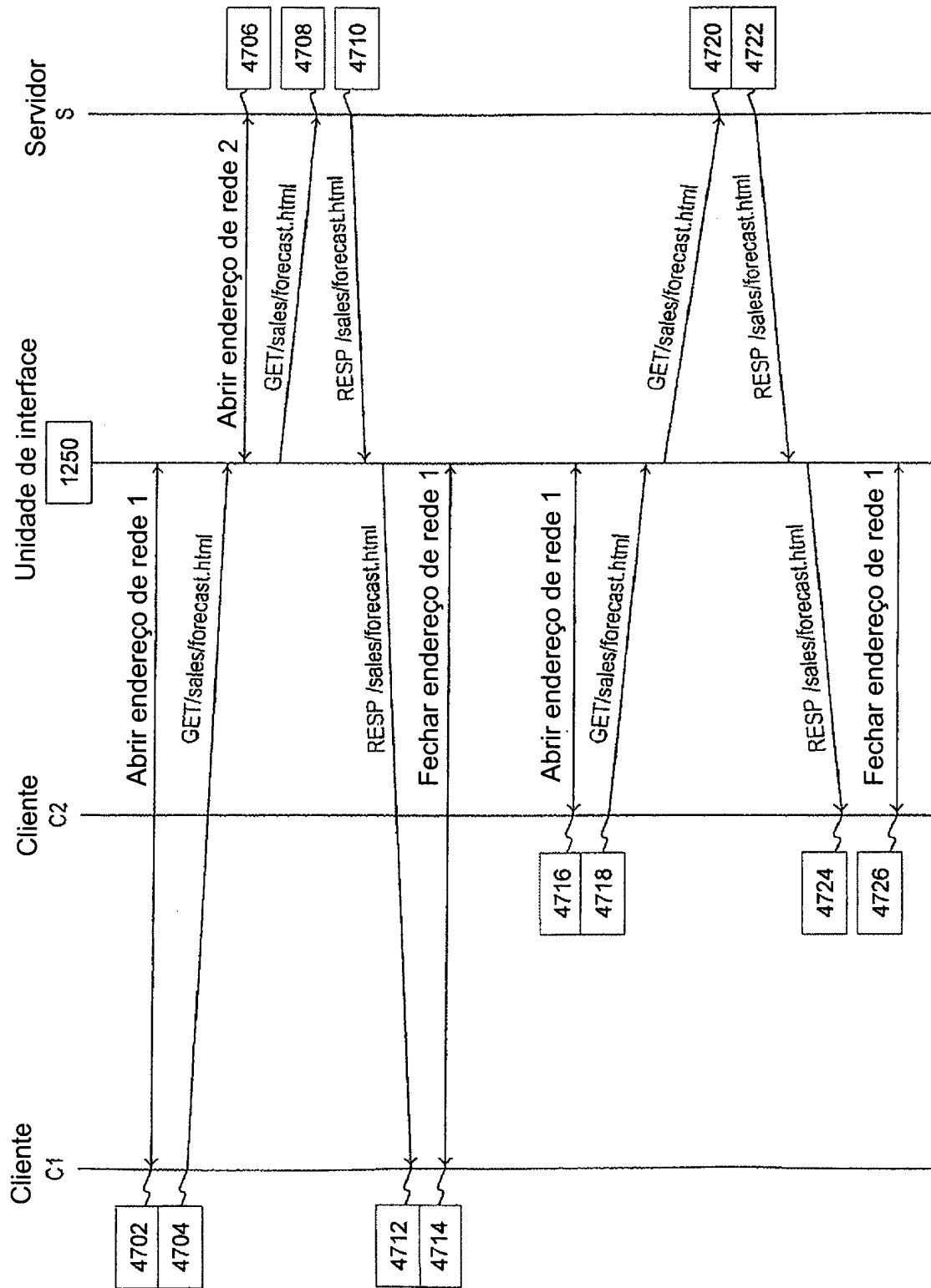


FIG. 34

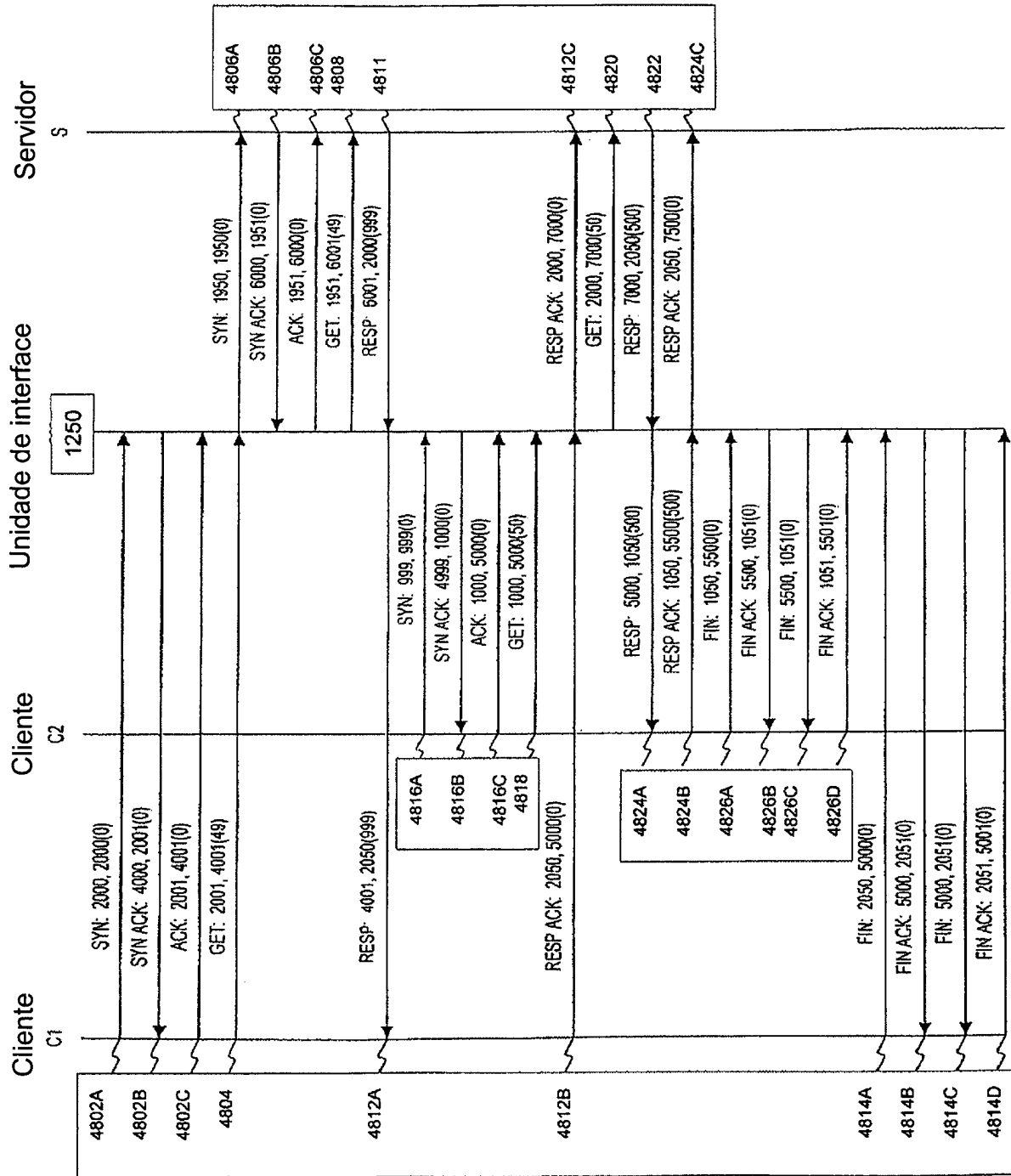


FIG. 35

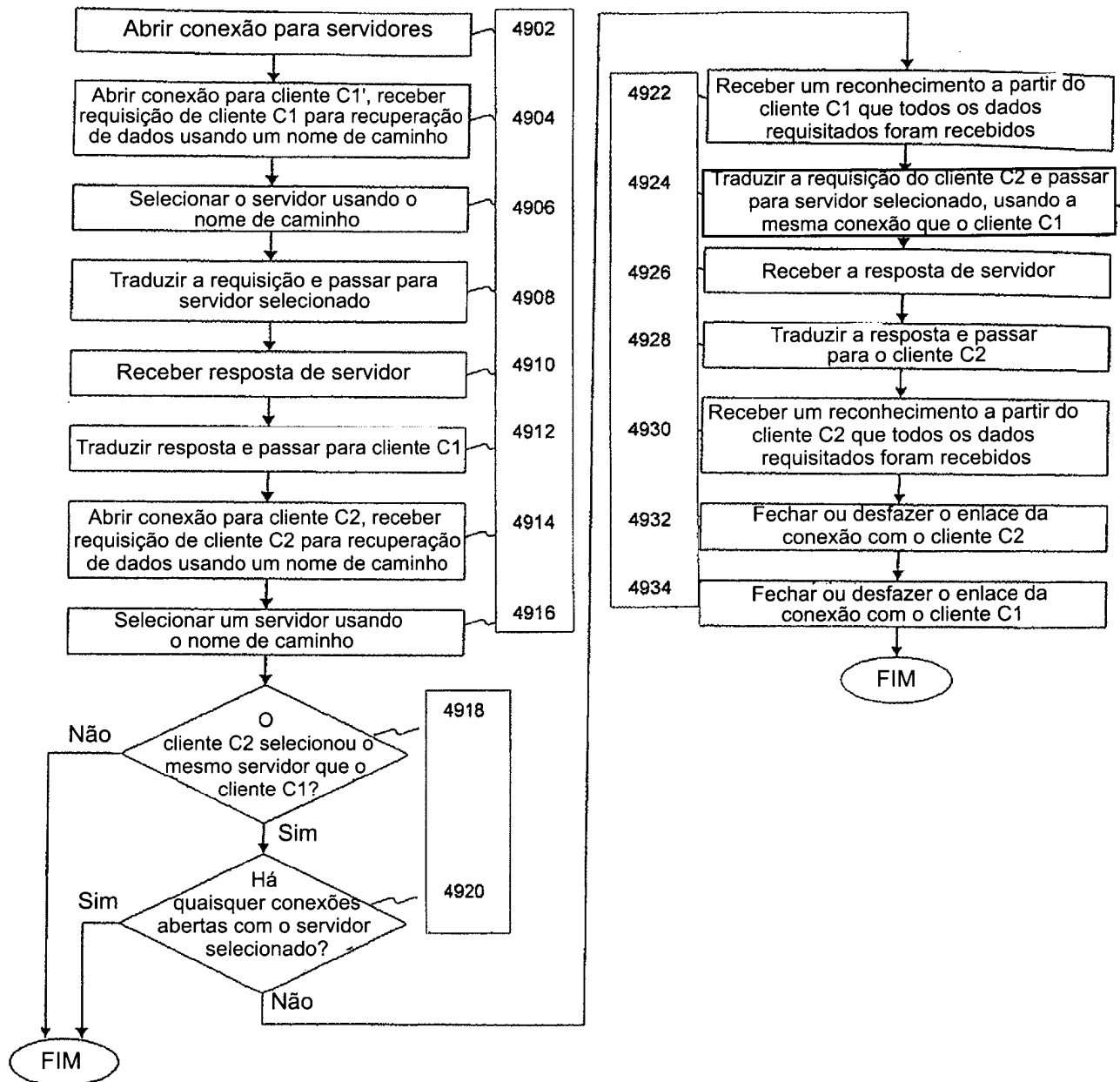


FIG. 36

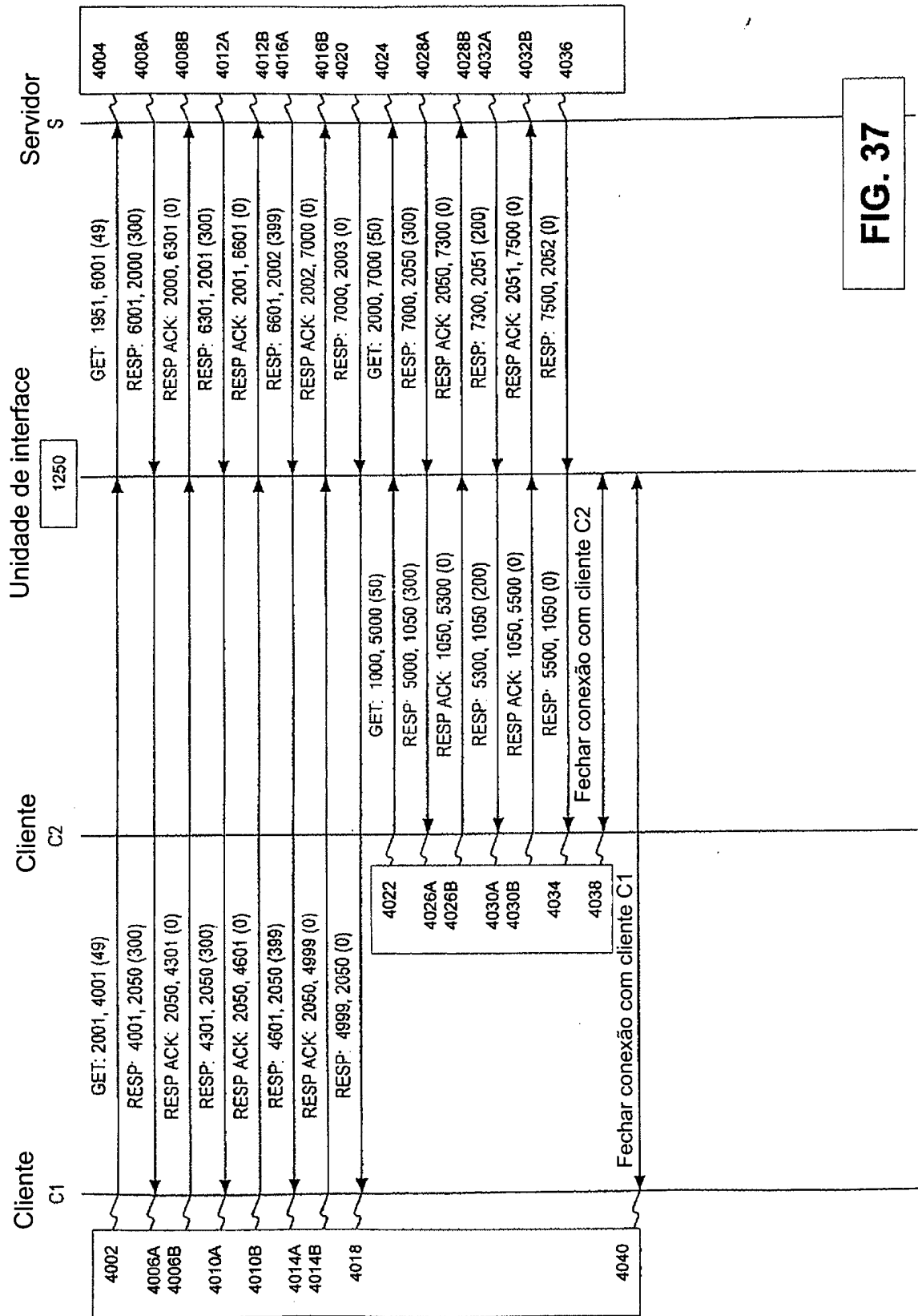


FIG. 37

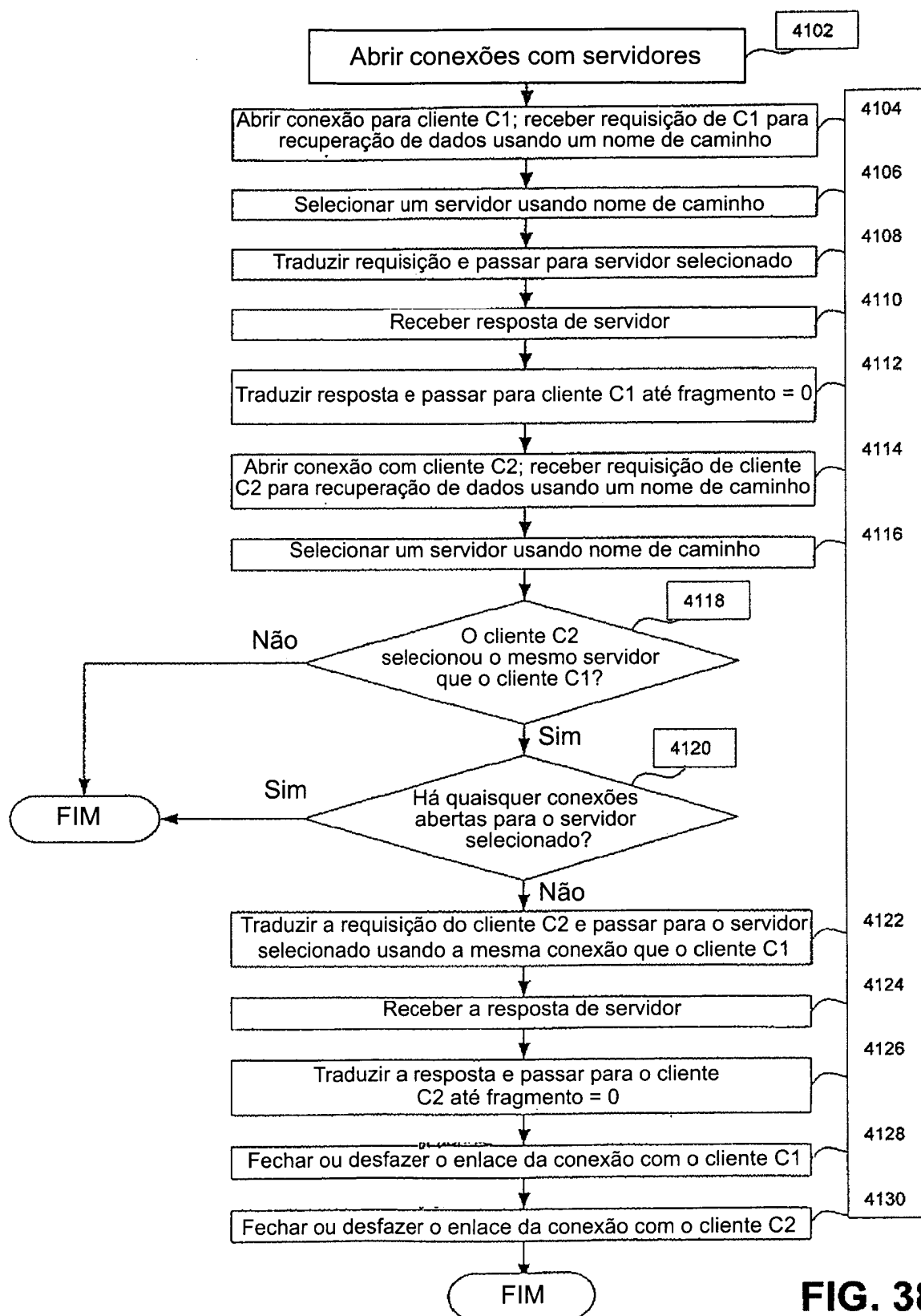


FIG. 38

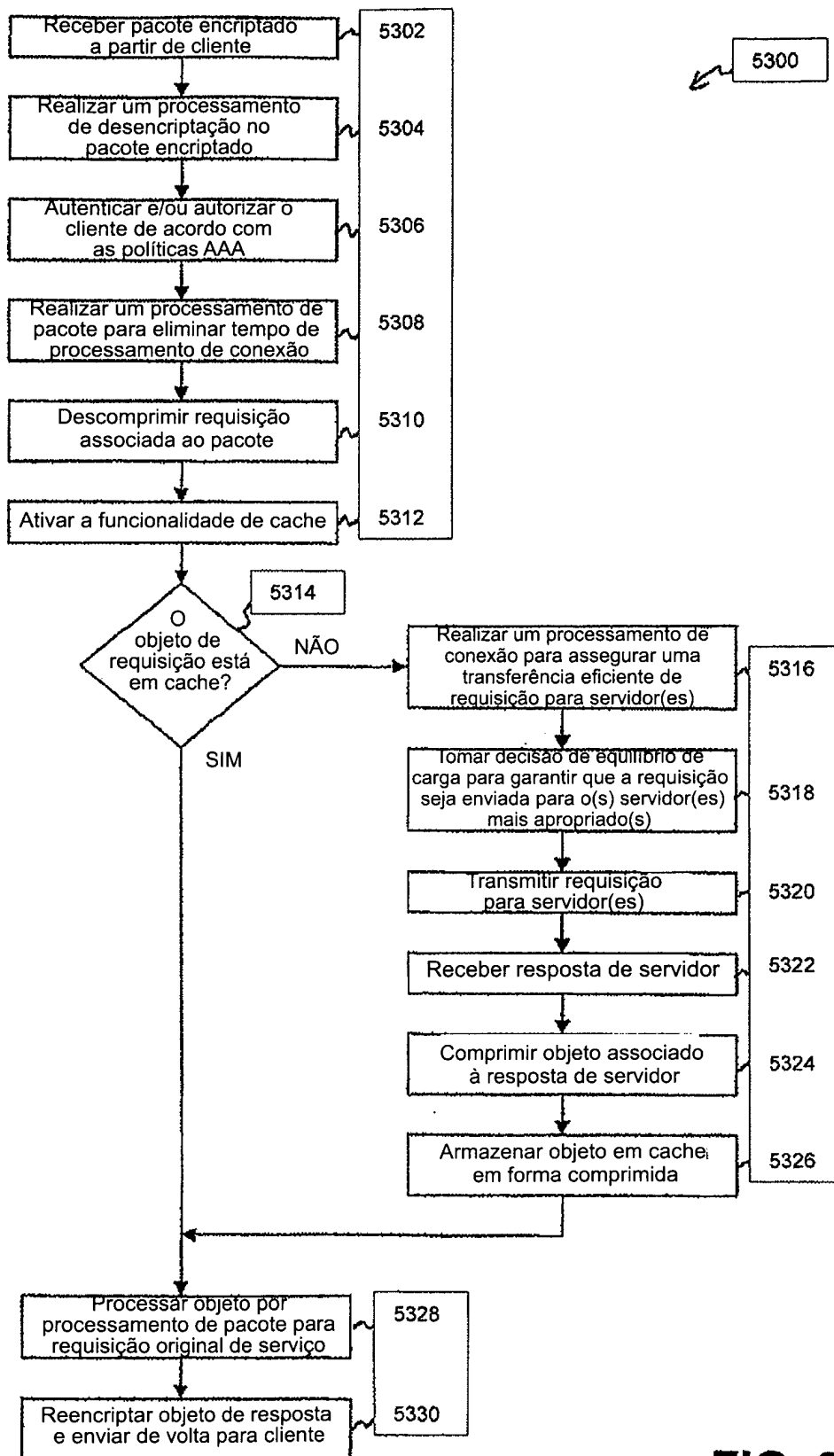
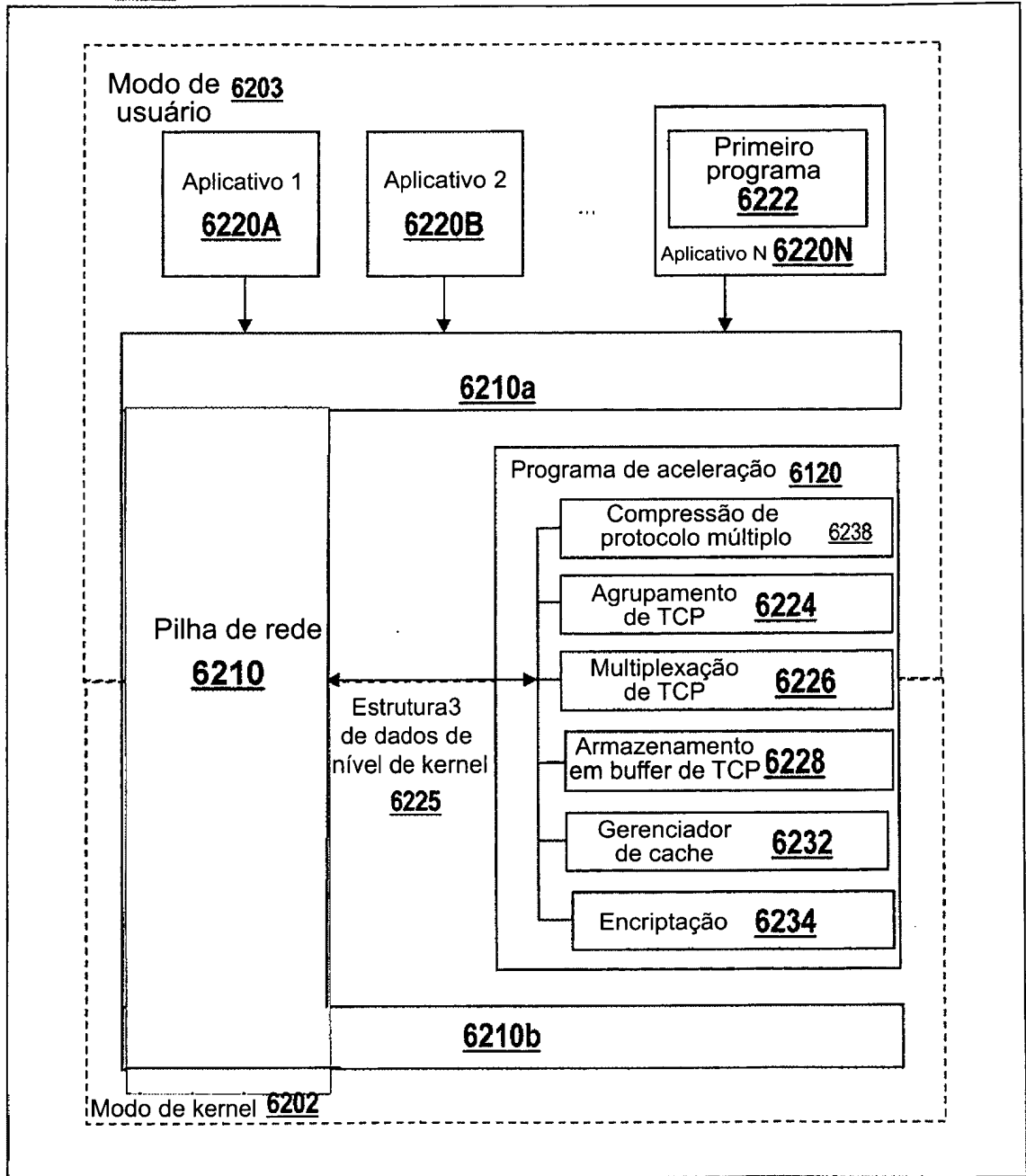


FIG. 39

Cliente **6205***Fig. 40A*

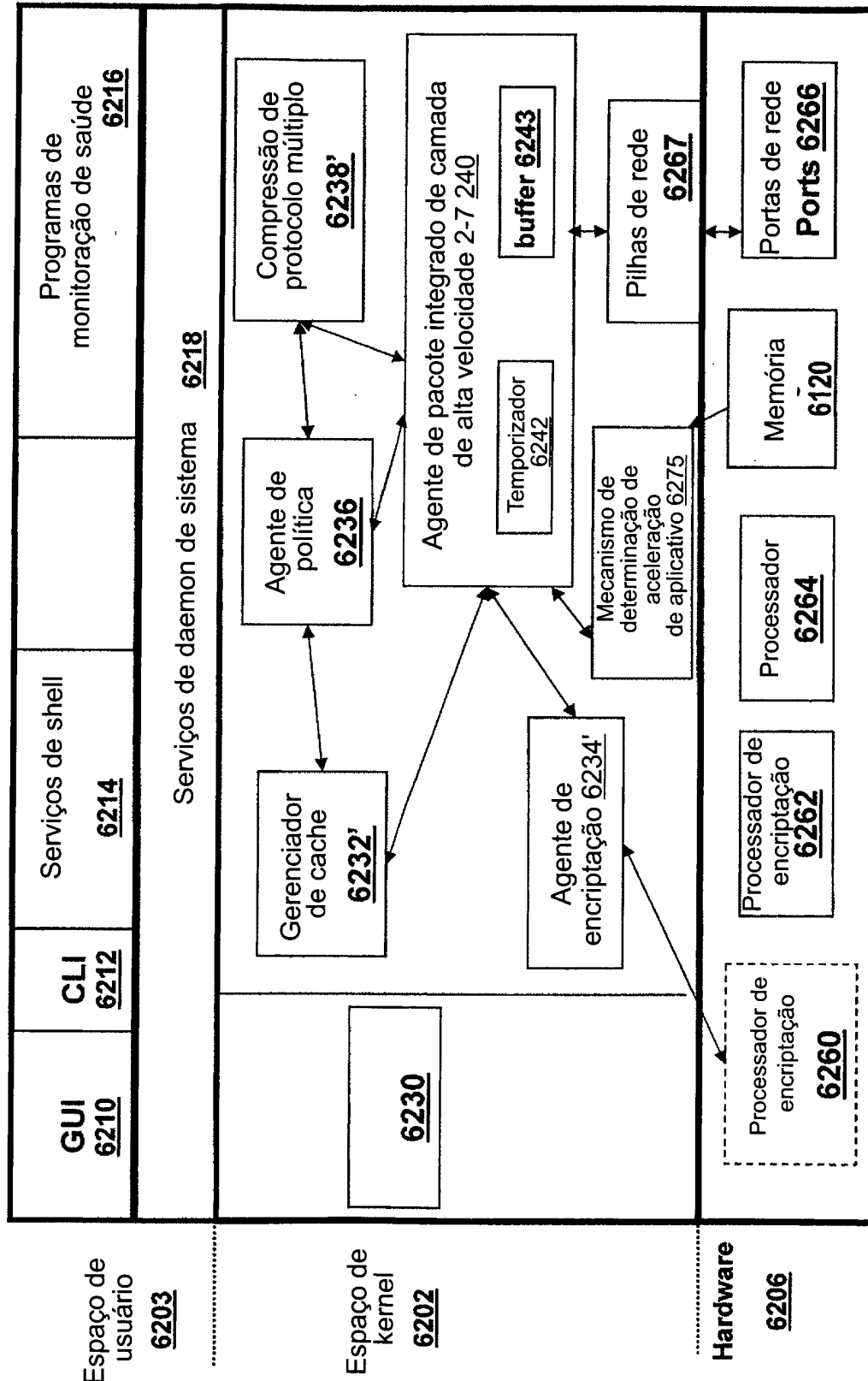
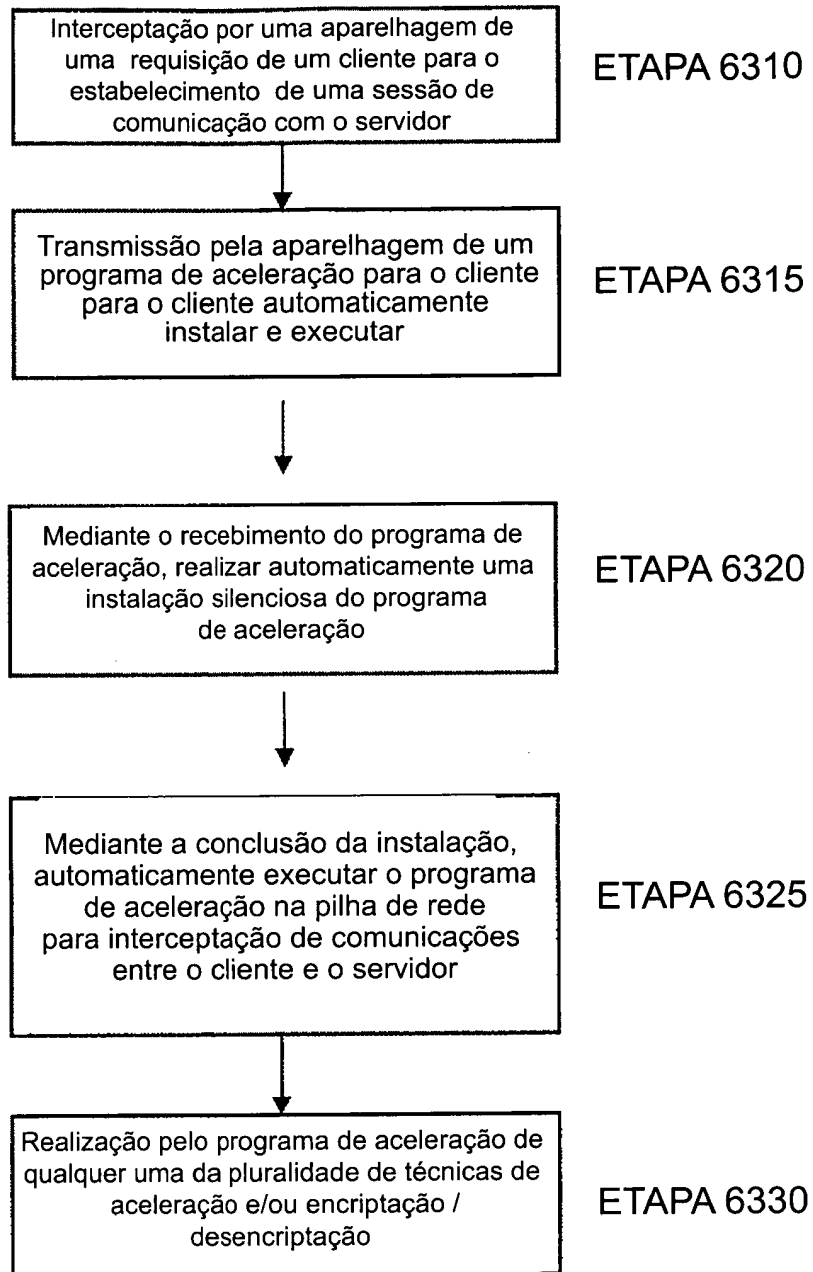


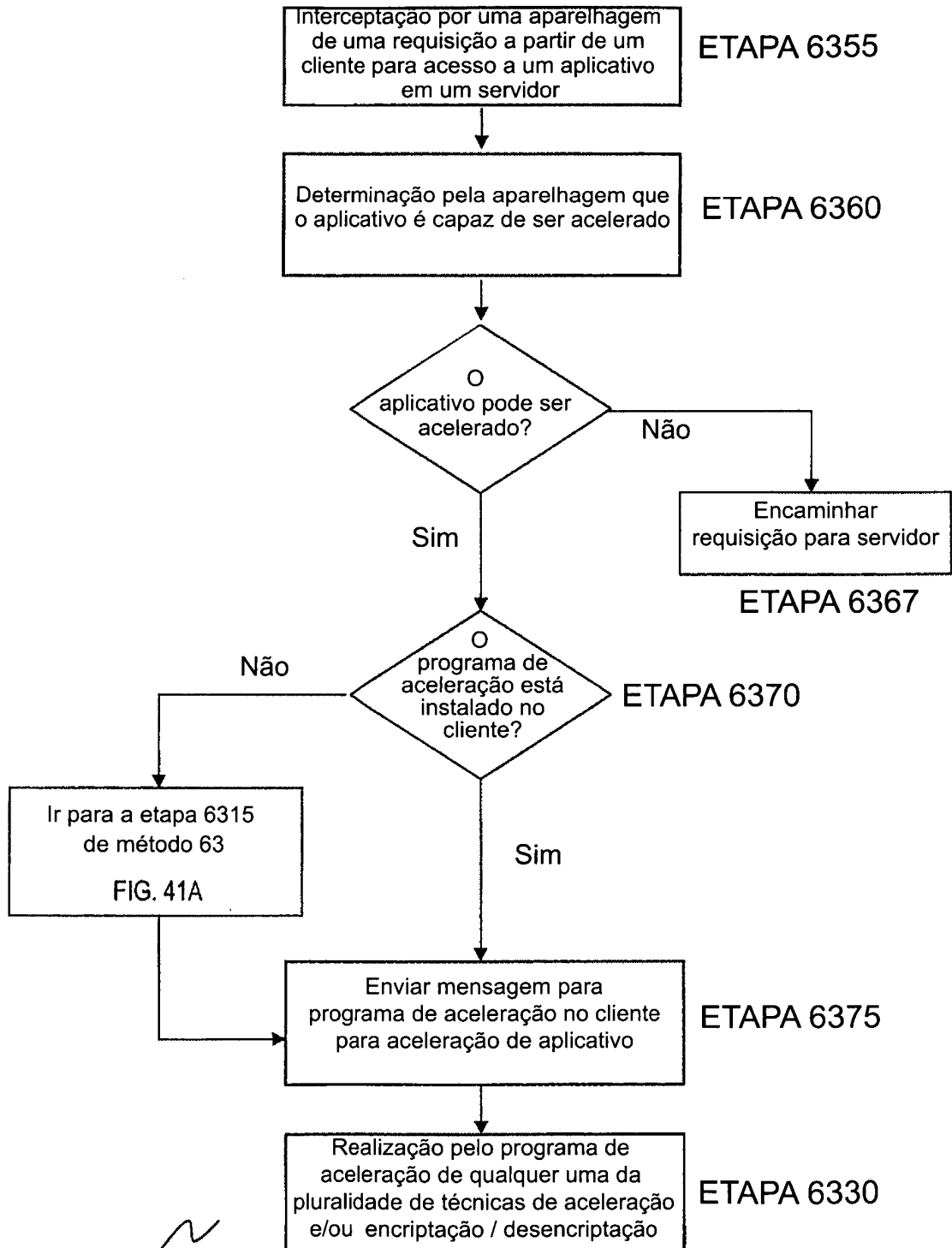
Fig. 40B

58/71



6300

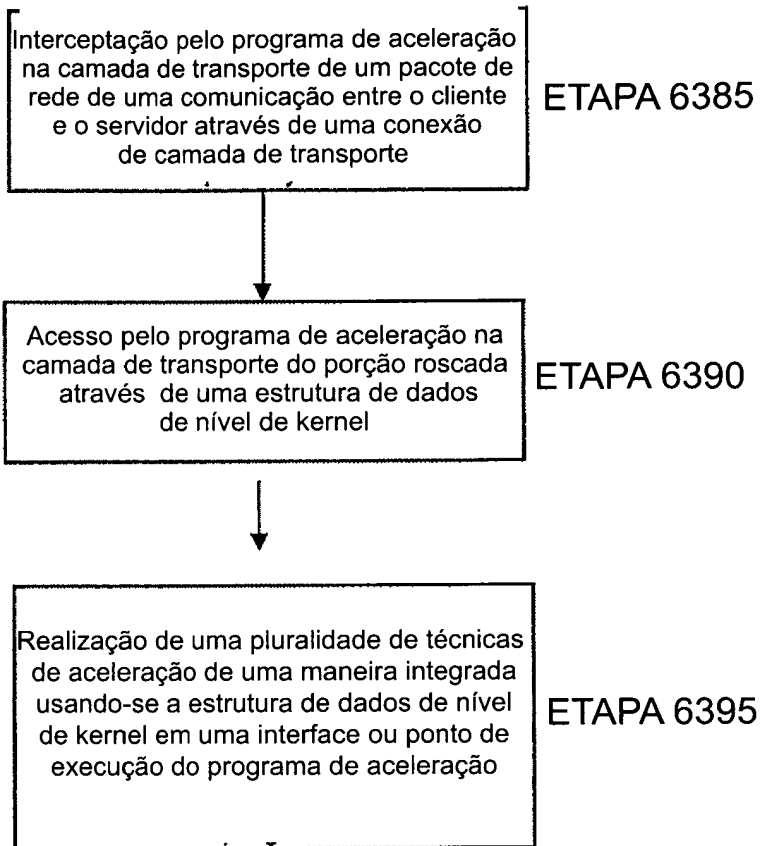
Fig. 41A



6350

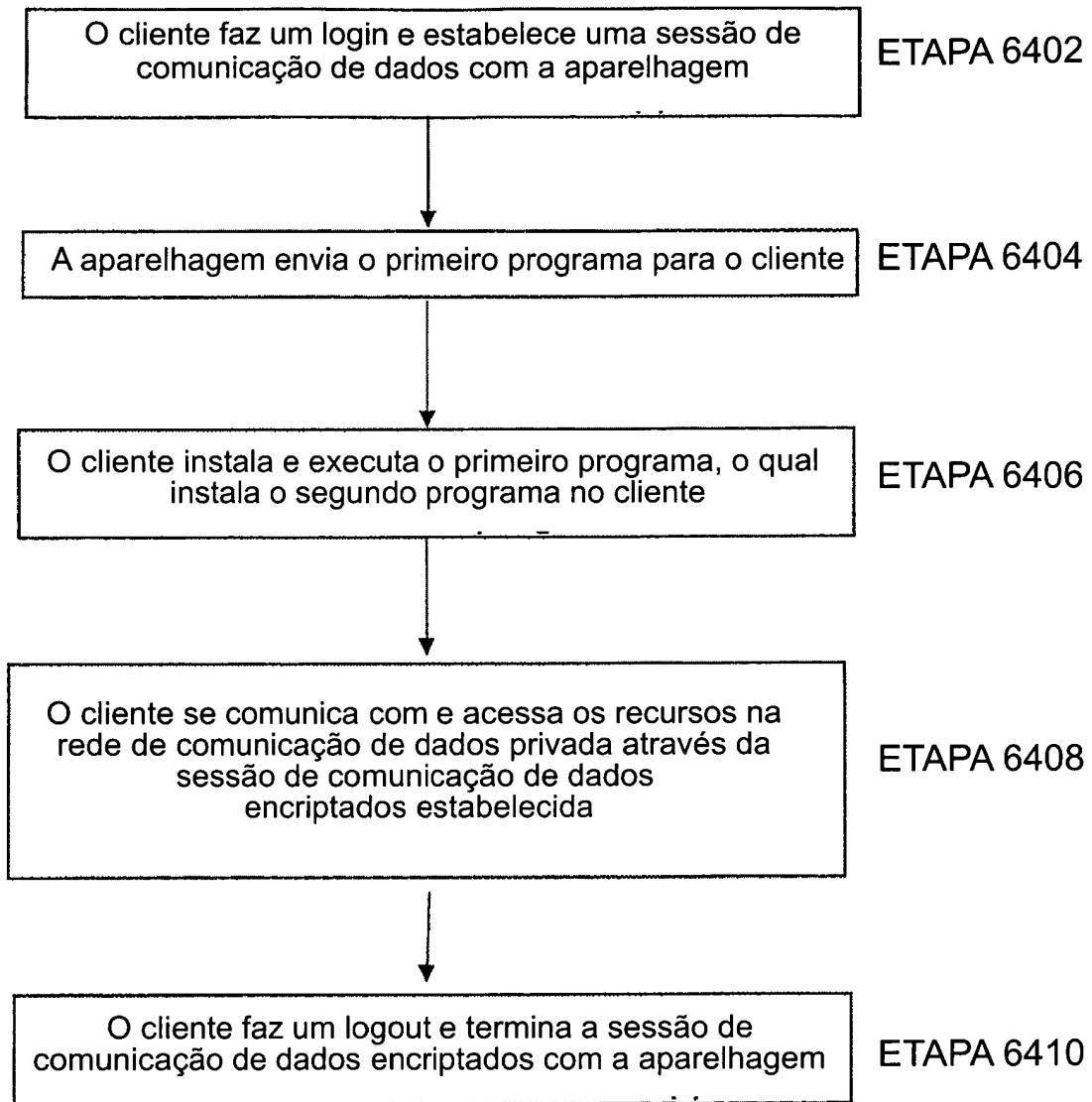
Fig. 41B

60/71



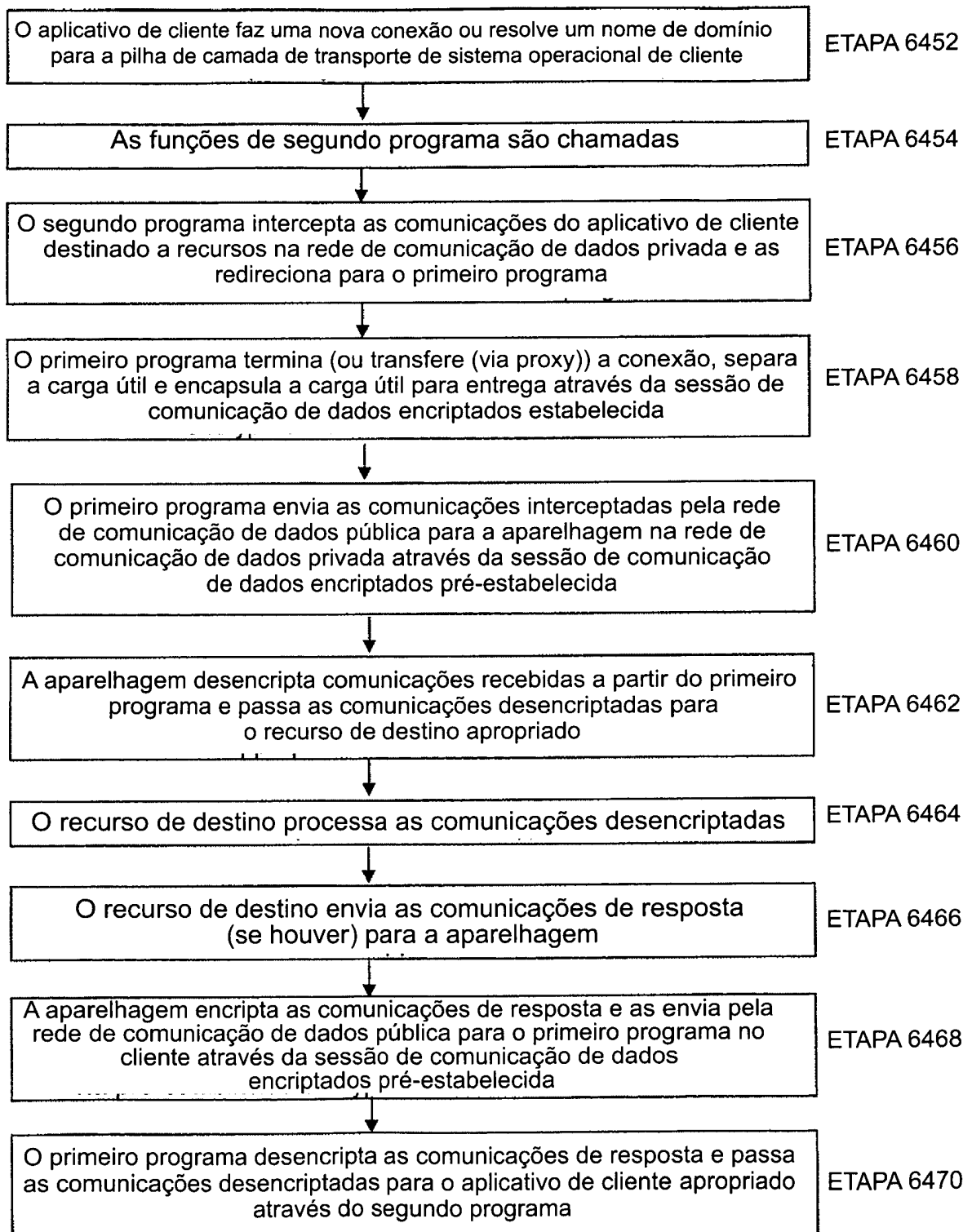
~
6380

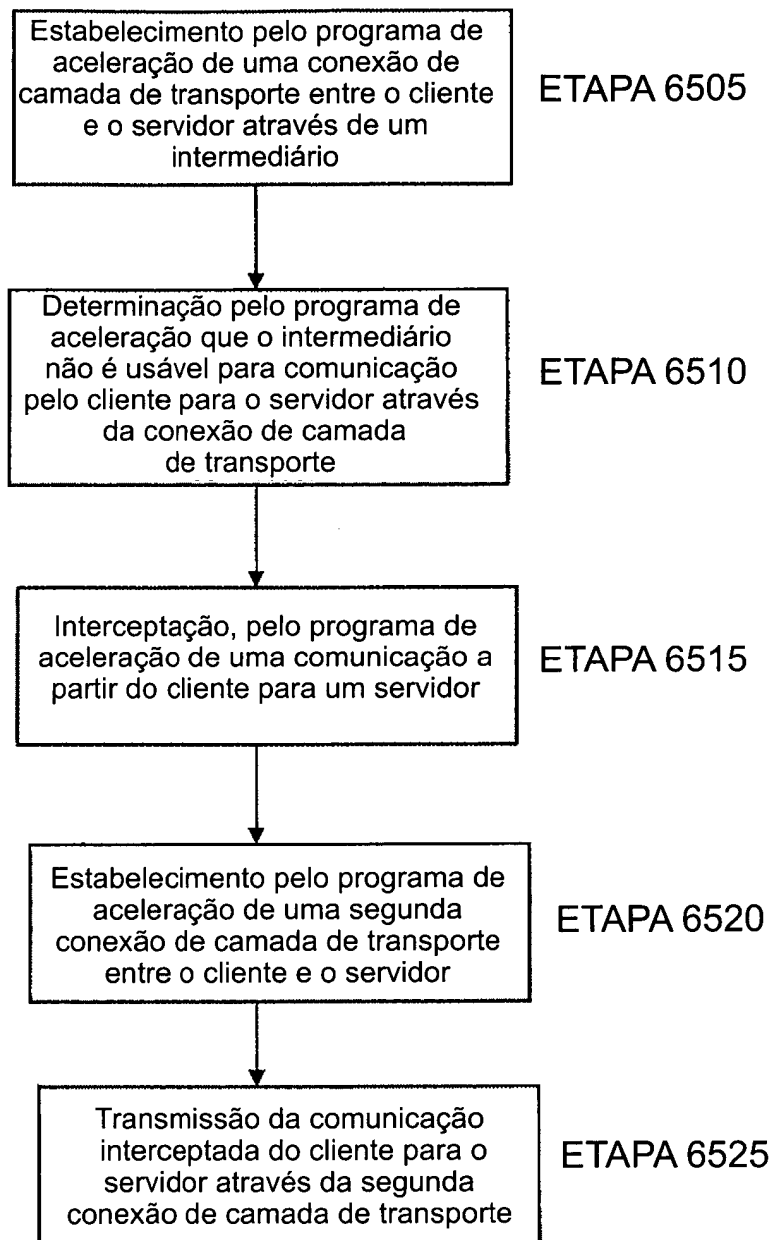
Fig. 41C



6400

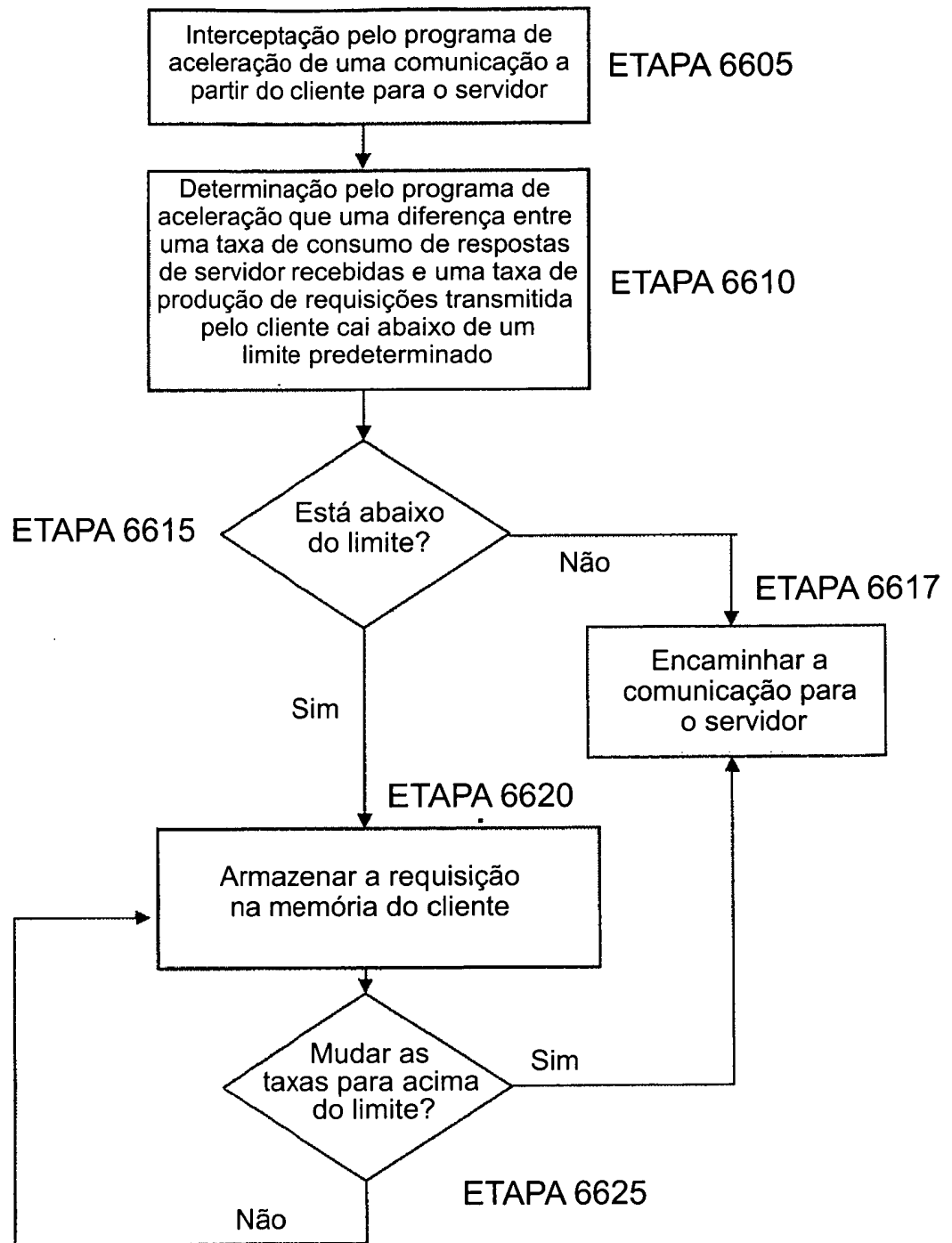
Fig. 42A

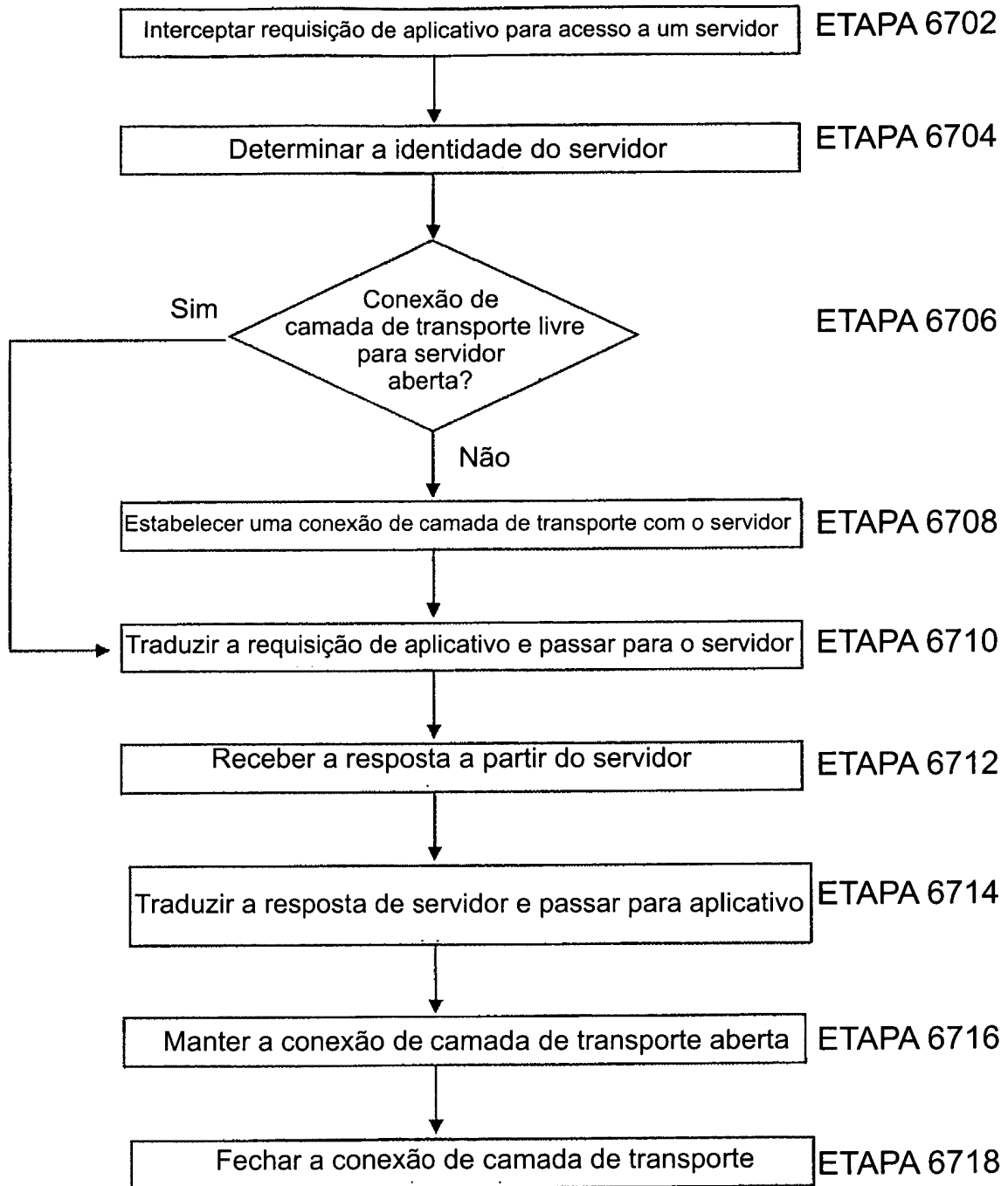




~
6500

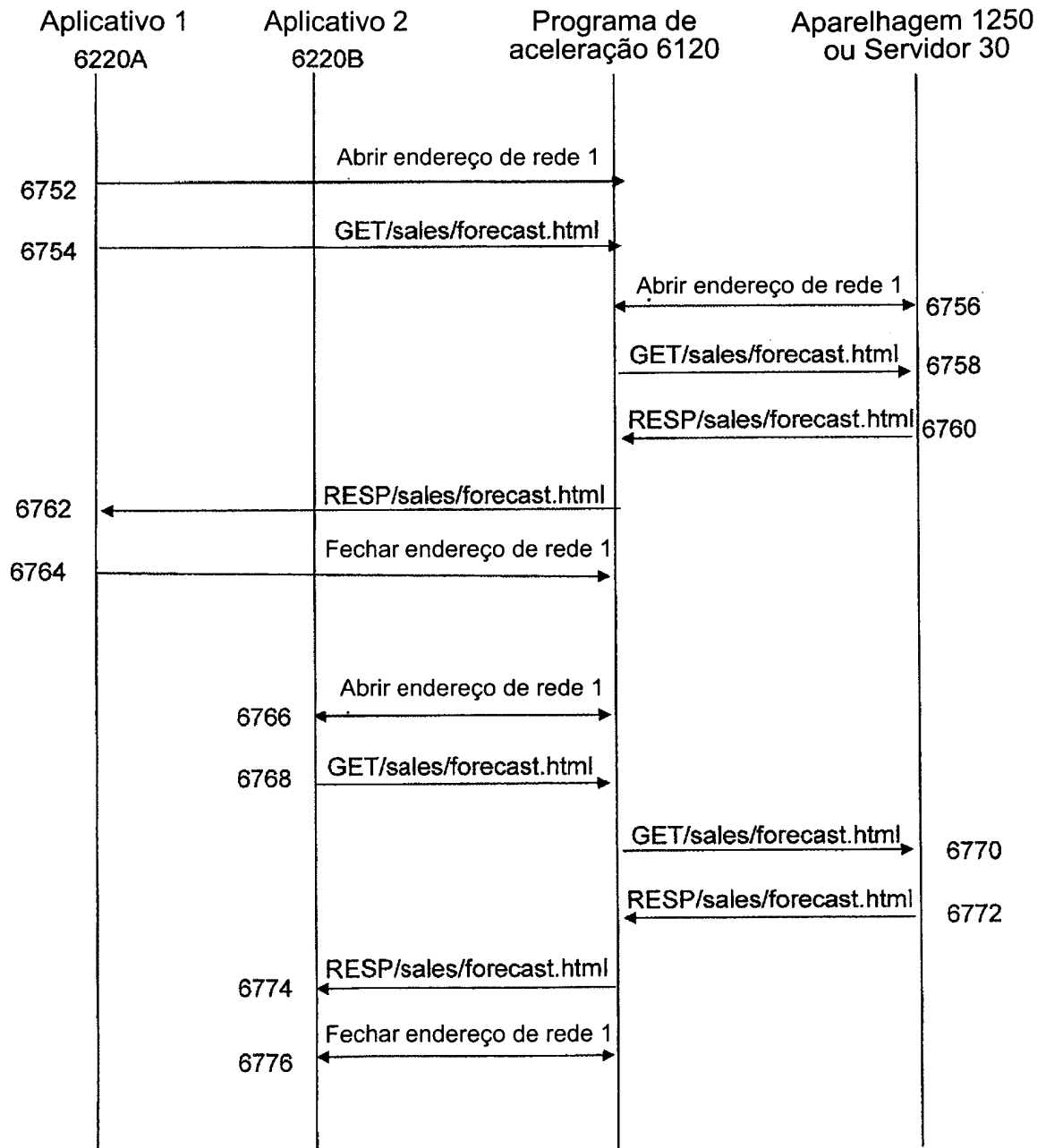
Fig. 43



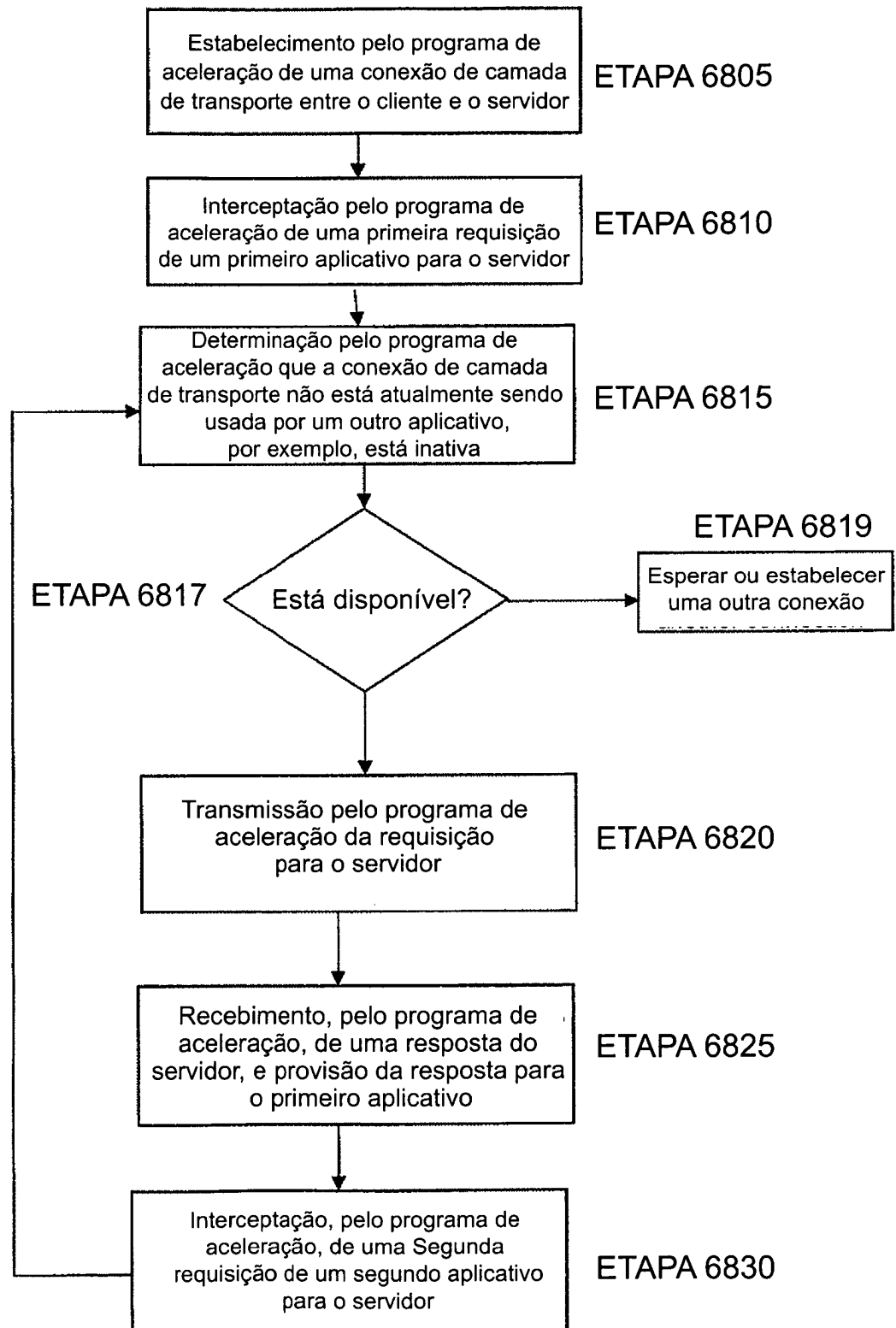


6700

Fig. 45A

*Fig. 45B*

67/71



6800

Fig. 46

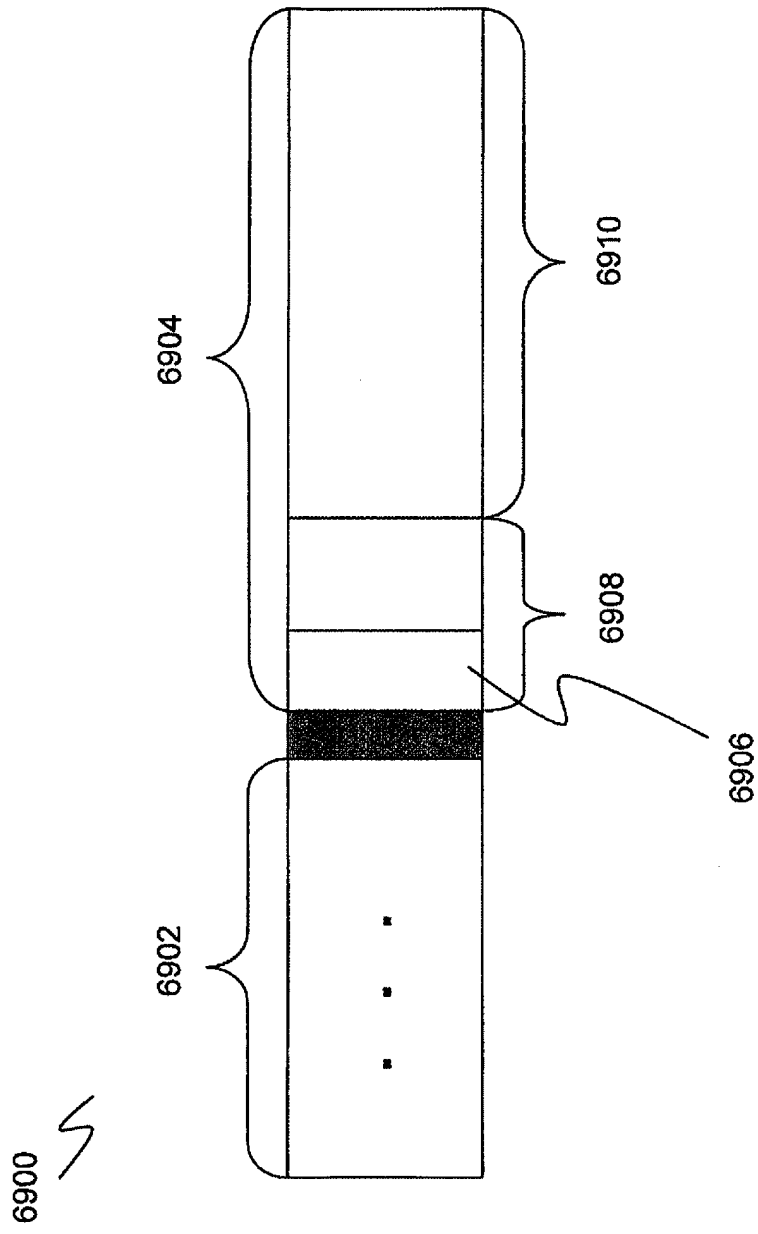



Fig. 47

7600 

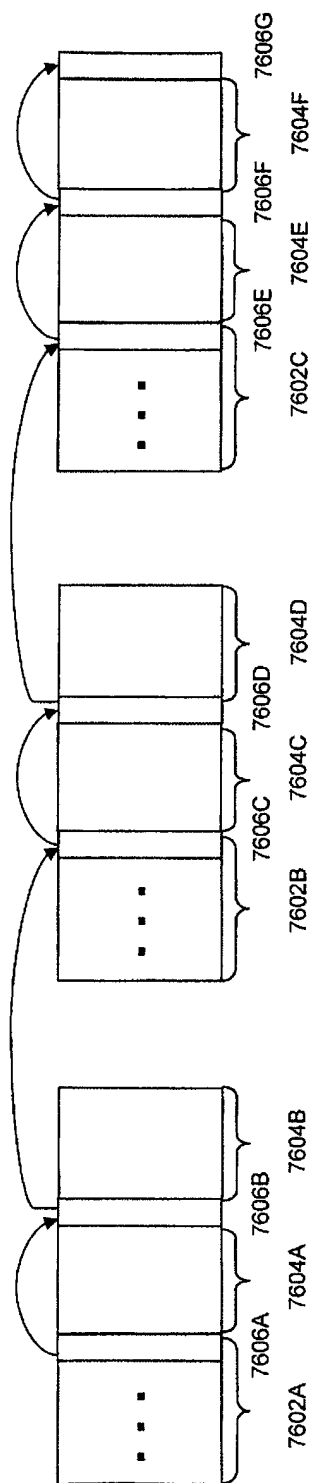


Fig. 48

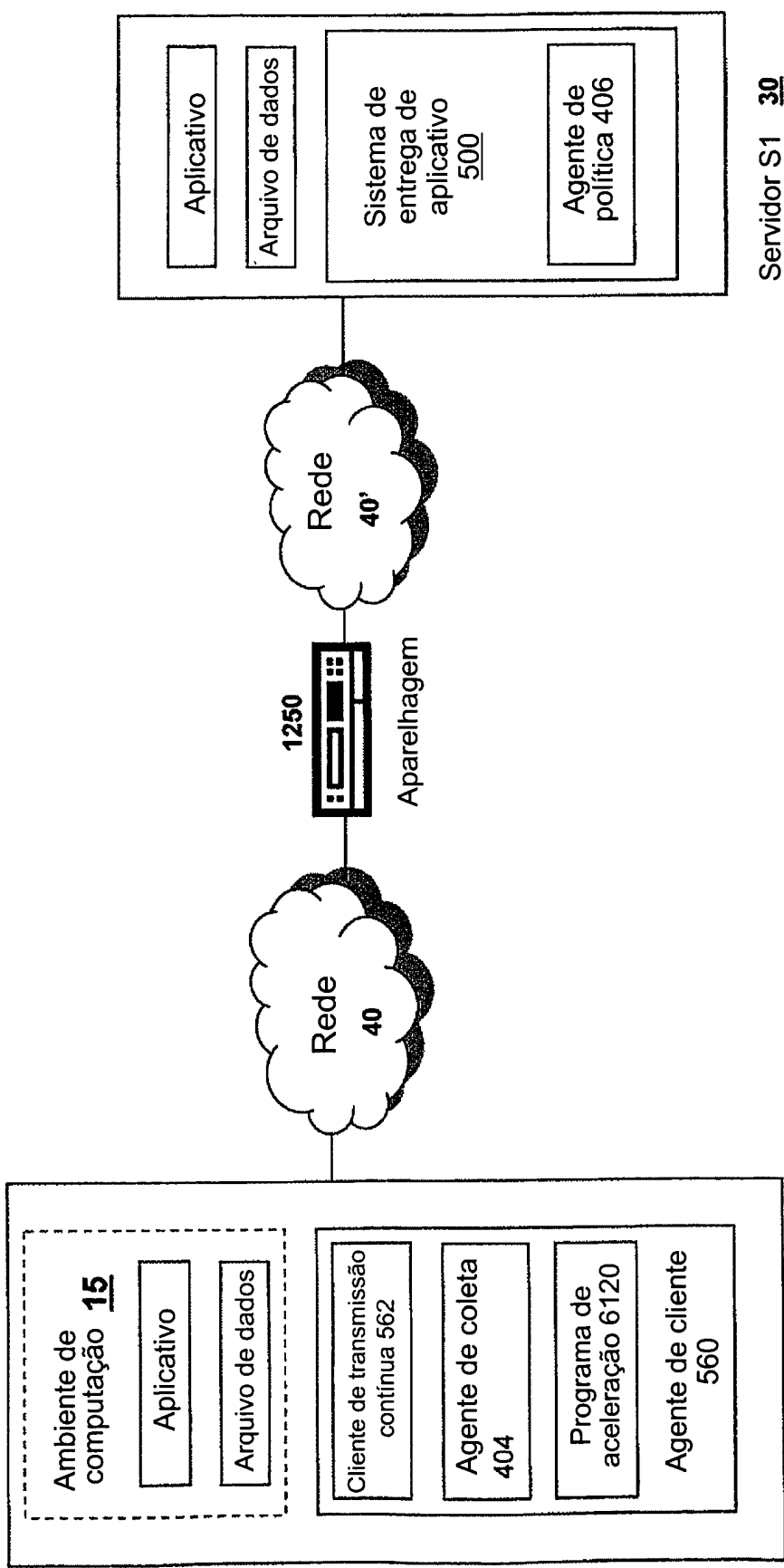
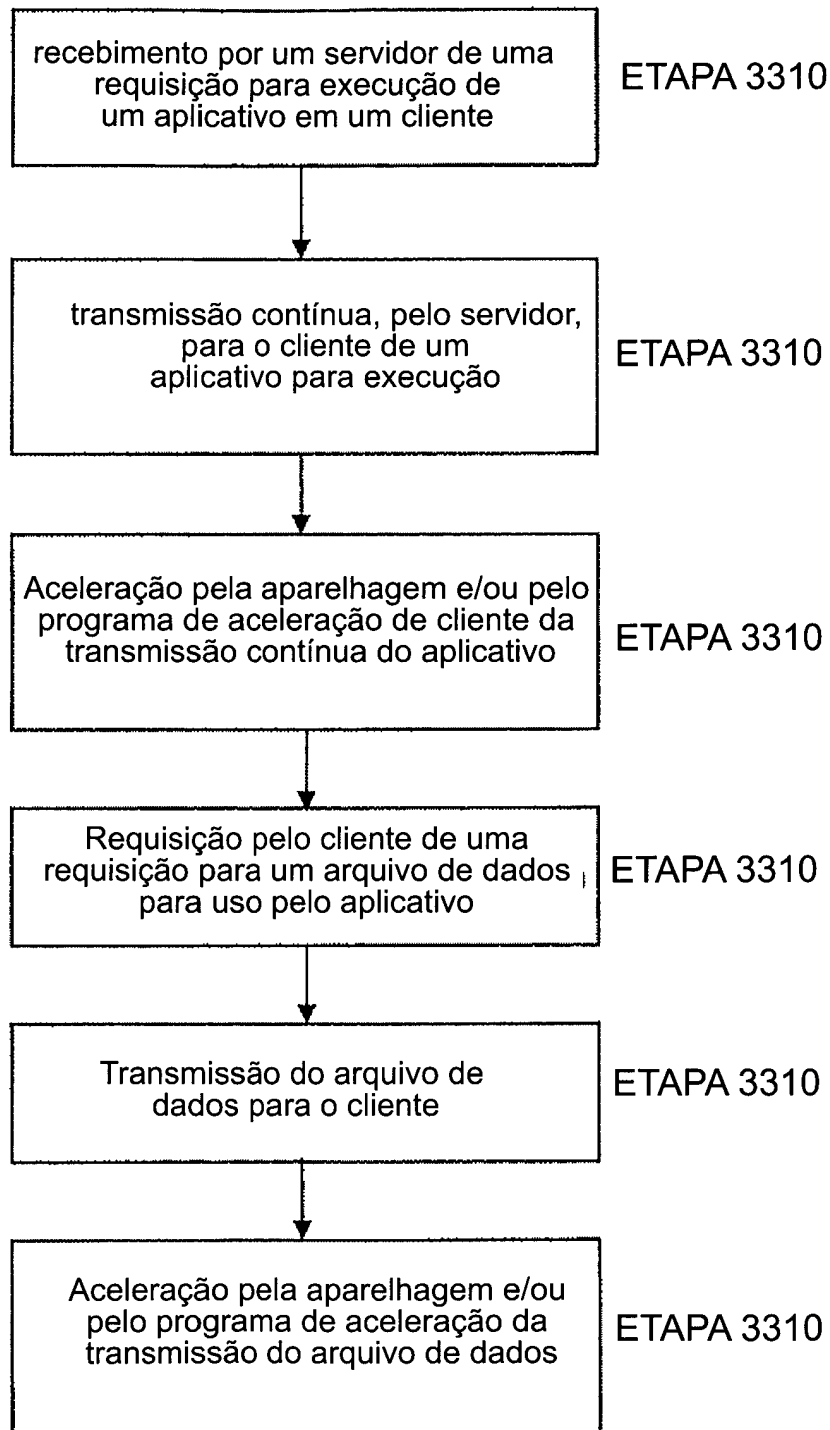


FIG. 49A



8000

FIG. 49B