República Federativa do Brasil
Ministério da Indústria, Comércio Exterior e Serviços
Instituto Nacional da Propriedade Industrial

(11) PI 0305138-2 B1

(22) Data do Depósito: 14/11/2003

(45) Data de Concessão: 07/03/2017

(54) Título: MÉTODO PARA A TRANSMISSÃO DE DADOS POR UM SERVIDOR DE E-MAIL E SISTEMA DE SERVIDOR DE E-MAIL

(51) Int.Cl.: G06F 15/167

(30) Prioridade Unionista: 20/11/2002 US 60/428.153, 21/05/2003 US 10/442.380

(73) Titular(es): MICROSOFT TECHNOLOGY LICENSING, LLC

(72) Inventor(es): JOSEPH R. WARREN; KARL FROELICH; NICOLE A. BONILLA; REMI A. LEMARCHAND; RONALD E. GRAY; ALEC DUN; AARON HARTWELL; STEVEN F. GODDARD; BRENT CURTIS; BRENDAN POWER

"MÉTODO PARA A TRANSMISSÃO DE DADOS POR UM SERVIDOR DE E-MAIL E SISTEMA DE SERVIDOR DE E-MAIL".

REFERÊNCIA A PEDIDO RELACIONADO

[001] Este pedido reivindica o benefício da data de depósito de pedido provisório dos Estados Unidos número de série 60/428153, depositado em 20 de novembro de 2002 e incorporado aqui dentro por referência.

CAMPO TÉCNICO DA INVENÇÃO

[002] Esta invenção relaciona-se em geral com redes de computadores e, mais particularmente, com métodos para comunicação entre aplicações cliente e servidor tal como aplicações de correio eletrônico.

FUNDAMENTO DA INVENÇÃO

[003] O correio eletrônico tornou-se um método importante para comunicação. Os sistemas de correio eletrônico incluem um componente servidor (por exemplo, o Microsoft Exchange Server) e um componente cliente (por exemplo, o Microsoft Outlook ou o Microsoft Outlook Express). Estes componentes tipicamente são aplicações de software que são configuradas para executar nos dispositivos de computação (por exemplo, servidores, PCs, laptops e PDAs).

[004] Alguns tipos de servidores de correio eletrônico são configurados para permitir que o correio eletrônico seja acessado via um cliente de navegador da Internet (por exemplo, o Microsoft Internet Explorer) ao invés do que por um cliente de correio eletrônico dedicado. Nestes sistemas, o navegador interage com o servidor de correio eletrônico e quaisquer funções requeridas de serem executadas no sistema cliente são executadas através do navegador (por exemplo, por se transferir o Javascript) ou através do uso de Páginas Ativas do Servidor.

[005] Desde que os clientes e os servidores frequentemente

estão conectados por redes que possuem uma largura de banda pequena e uma alta latência (por exemplo, conexão de discagem lenta), vários clientes e servidores de correio eletrônico são configurados para armazenar as instruções pendentes e então enviar várias instruções juntas. Por exemplo, ao invés de enviar um comando de pasta aberto e enviar um comando de calendário aberto, um cliente pode armazenar a primeira instrução e combinar a mesma com a segunda instrução e então enviar as duas instruções juntas. Este esquema de armazenamento, combinação e envio tende a permitir um uso mais eficiente dos recursos da rede e do servidor, desde que existe algumas atividades de suporte associadas com cada transmissão.

[006] Alguns sistemas de técnica anterior contavam com uma única memória intermediária alocada em cada um dentre o cliente e o servidor para atuar como uma área de armazenamento de dados para instruções e/ou dados que são aguardados de serem enviados juntos. Em um exemplo de tal sistema, o cliente utiliza uma memória intermediária para armazenar instruções e dados que são para ser enviados para o servidor. Uma vez que a memória intermediária esteja cheia ou próxima de estar cheia, o cliente envia o conteúdo da memória intermediária para o servidor. O servidor armazena o conteúdo recebido em uma memória intermediária e começa a analisar e executar as instruções. Um ponteiro pode ser utilizado para designar a próxima requisição a ser servida.

[007] O servidor monta suas respostas em sua memória intermediária e garante que o conteúdo de sua memória intermediária não exceda ao tamanho da memória intermediária do cliente. Se o servidor não estiver apto a completar qualquer requisição em sua memória intermediária (por exemplo, porque não existe espaço suficiente na memória intermediária), o servidor grava as requisições

incompletas na memória intermediária e envia as mesmas de volta para o cliente com as respostas completas.

[008] Em alguns sistemas, o cliente pode ser configurado para especificar quanta memória o cliente está desejando para alocar para sua memória intermediária. Por exemplo, o cliente pode indicar para o servidor que somente 32 KB serão devotados para sua memória intermediária. Em resposta, o servidor irá garantir que ele não envie para o cliente mais do que 32 KB ao mesmo tempo.

[009] Dada a natureza de largura de banda pequena e alta latência das conexões utilizadas entre vários clientes e servidores de correio eletrônico, é necessário um sistema e método para aperfeiçoar o desempenho.

SUMÁRIO DA INVENÇÃO

[0010] O dito a seguir apresenta um sumário simplificado da invenção de modo a proporcionar um entendimento básico de alguns aspectos da invenção. Este sumário não é uma vista geral extensiva da invenção. Ele não é pretendido para identificar os elementos chave/críticos da invenção ou para delinear o escopo da invenção. Seu único propósito é apresentar alguns conceitos da invenção de uma forma simplificada como um prelúdio para a descrição mais detalhada que é apresentada posteriormente.

[0011] É revelado um método para requisitar respostas. Em uma modalidade, o método é otimizado para uso entre um cliente de correio eletrônico e um servidor de correio eletrônico. O método pode incluir alocar uma primeira memória intermediária em um cliente e então utilizar a memória intermediária para montar uma ou mais requisições para um servidor. O cliente também pode ser configurado para anexar um cabeçalho junto ao conteúdo da primeira memória intermediária e o cabeçalho pode ser configurado para incluir um indicador de se a resposta para as requisições pelo servidor estão ou não compactadas

antes que elas seja retornadas para o cliente.

[0012] Outra opção para o servidor pode ser ofuscar ou criptografar as requisições antes que elas sejam enviadas para o cliente. Bits indicadores correspondentes para estes aspetos também podem ser incluídos no cabeçalho.

[0013] Em algumas implementações, o cliente pode ser configurado para utilizar RPCs (Chamadas de Procedimento Remoto) para implementar as requisições. Em algumas destas implementações, o cabeçalho pode ser um cabeçalho de chamada de procedimento remoto com comprimento fixo. Em algumas modalidades, o cabeçalho pode adicionalmente incluir um indicador do tamanho descompactado de um conjunto de respostas que o cliente está configurado para processar.

[0014] Um método para transferir dados de um servidor para um cliente também é revelado. O método pode incluir receber um lote de requisições a parir de um cliente, onde uma das requisições é uma requisição que o servidor enviar as respostas para as requisições utilizando o encadeamento. Em resposta, o servidor pode montar um primeiro conjunto de respostas para o cliente, compactar o conjunto e anexar um cabeçalho proporcionando informação a cerca do primeiro conjunto de respostas (por exemplo, seu tamanho). O servidor pode repetir este processo para uma série de conjuntos de respostas e então enviar os cabeçalhos e as respostas em um lote para o cliente. Cada cabeçalho pode incluir um ponteiro para o próximo cabeçalho, desse modo permitindo que o cliente decodifique de forma apropriada as respostas. O cabeçalho final no lote pode ser configurado com uma marca especial para indicar que ele corresponde à resposta final.

[0015] Em algumas implementações, o cliente pode ser configurado para comunicar o tamanho de sua memória intermediária para o servidor. O servidor então pode utilizar esta informação para

estabelecer o tamanho da sua própria memória intermediária, desse modo impedindo as resposta de transbordarem a memória intermediária do cliente quando o cliente recebe as mesmas. Em adição, o cliente pode ser configurado para comunicar o tamanho de um conjunto de respostas descompactado que ele está configurado para manipular. O servidor pode utilizar esta informação para criar conjuntos de respostas que são o tamanho coreto e pode ou não compactar os conjuntos de respostas. O servidor pode encadear os conjuntos de respostas e pode continuar a encadear os conjuntos, compactados ou não, até que a memória intermediária do servidor esteja cheia ou próxima de cheia. O conjunto de respostas encadeado pode então ser enviado para o cliente, o qual pode descompactar estes conjuntos (se aplicável) e pode processar cada um dos conjuntos de respostas individualmente.

[0016] Por compactar vários conjuntos de respostas no servidor e enviar estes em um único lote (isto é, um lote "encadeado" ou "empacotado") existe o potencial para aumentar o desempenho nas comunicações entre o cliente e o servidor. Enquanto os sistemas anteriores têm utilizado a compactação para reduzir o número total de bytes enviados entre o cliente e o servidor, por empacotar as memórias intermediárias antes que elas sejam enviadas, mais dados podem ser adicionados para a memória intermediária e enviados em cada sessão, assim reduzindo o número total de idas e volta para as redes com alta latência.

[0017] Ao mesmo tempo em que esta técnica pode ter ampla aplicabilidade, ela é especialmente bem adequada para operações entre clientes de correio eletrônico e servidores de correio eletrônico. Por exemplo, o método pode ser utilizado com o Outlook da Microsoft para as operações de Transferência Rápida tal como CopyMessages. Esta função copia os cabeçalhos de mensagem de um servidor para o

cliente.

[0018] Aspectos e vantagens adicionais da invenção serão expostos na descrição que se segue e em parte serão óbvios a partir da descrição, ou podem ser aprendidos pela prática da invenção. Os aspectos e vantagens da invenção podem ser realizados e obtidos por meio dos instrumentos e das combinações particularmente salientadas nas reivindicações anexas. Estes e outros aspectos da presente invenção irão se tornar mais totalmente aparentes a partir da descrição seguinte e das reivindicações anexas, ou podem ser aprendidos pela prática da invenção como exposta daqui para frente. Os tópicos incluídos abaixo na descrição detalhada são somente para propósito de organização e não são pretendidos para limitar ou modificar o escopo da invenção ou das reivindicações anexas.

[0019] Outras vantagens irão se tornar aparentes a partir da descrição detalhada seguinte quando considerada em conjunto com os desenhos, nos quais:

BREVE DESCRIÇÃO DOS DESENHOS

[0020] A Fig. 1 é um diagrama de blocos representando uma rede de computadores dentro da qual a presente invenção pode ser incorporada;

A Fig. 2 é um diagrama de blocos de uma arquitetura de um computador dentro do qual a presente invenção pode ser incorporada;

A Fig. 3 é um diagrama de blocos apresentando uma troca de requisição e resposta entre um cliente de correio eletrônico e um servidor de correio eletrônico de acordo com a presente invenção;

A Fig. 4A é uma representação de um processo do modo de transferência rápida com duas etapas de acordo com um aspecto da presente invenção;

A Fig. 4B é uma representação de um processo do modo de transferência rápida de uma etapa de acordo com um aspecto da

presente invenção;

A Fig. 5 é um diagrama de blocos representando uma requisição de acordo com uma modalidade da presente invenção;

A Fig. 6 é um fluxograma representando um método para um servidor de correio eletrônico manipular o processamento das requisições de acordo com uma modalidade da presente invenção;

A Fig. 7 é uma representação de compactações por um servidor de correio eletrônico e acordo com uma modalidade da presente invenção;

A Fig. 8 é uma representação da compactação e do encadeamento das respostas por um servidor de correio eletrônico de acordo com uma modalidade da presente invenção;

A Fig. 9 é uma representação do conteúdo de uma memória intermediária de respostas de um servidor de correio eletrônico de acordo com uma modalidade da presente invenção;

A Fig. 10 é um fluxograma geralmente representando as etapas executadas por um servidor de correio eletrônico para proporcionar quadros de respostas para um cliente de correio eletrônico dentro de uma memória intermediária que é maior do que os quadros de acordo com uma modalidade da presente invenção; e

A Fig. 11 é um fluxograma geralmente representando as etapas para induzir por astúcia um servidor adicionando respostas adicionais para uma memória intermediária de respostas de acordo com uma modalidade da presente invenção.

DESCRIÇÃO DETALHADA

[0021] Na descrição seguinte, vários aspectos da presente invenção serão descritos. Para propósito de explicação, configurações e detalhes específicos são expostos de modo a proporcionar um entendimento completo da presente invenção. Entretanto, será aparente para os com conhecimento na técnica que a presente

invenção pode ser praticada sem os detalhes específicos. Adicionalmente, aspectos bem conhecidos podem ser omitidos ou simplificados de modo a não obscurecer a presente invenção.

[0022] Antes de continuar com uma descrição das várias modalidades da invenção, uma descrição do computador e do ambiente em rede nos quais as várias modalidades da invenção podem ser praticadas será agora proporcionada. Apesar de não ser requerido, a presente invenção pode ser implementada por programas que são executados por um computador. Geralmente, tais programas incluem rotinas, objetos, componentes, estruturas de dados e semelhantes que executam tarefas particulares ou implementam tipos de dados abstratos particulares. O termo "programa" como utilizado aqui dentro pode sugerir um único módulo de programa ou vários módulos de programa atuando de comum acordo. O termo "computador" como utilizado aqui dentro inclui qualquer dispositivo que eletronicamente execute um ou mais programas, tal como computadores pessoais (PCs), dispositivos de mão, sistemas com vários processadores, dispositivos eletrônicos de consumidor programáveis baseados em microprocessador, PCs de rede, minicomputadores, computadores de grande porte, utensílios de consumidor possuindo um microprocessador ou microcontrolador, roteadores, conectores, centralizadores e semelhantes. A invenção também pode ser empregada em ambientes de computação distribuída, onde as tarefas são executadas por dispositivos de processamento remotos que estão ligados através de uma rede de comunicações. Em um ambiente de computação distribuída, os programas podem estar localizados tanto nos dispositivos de armazenamento de memória locais como remotos.

[0023] Um exemplo de um ambiente em rede no qual a invenção pode ser utilizada será agora descrito com referência à FIG. 1. No

exemplo, a rede inclui vários computadores 10 comunicando-se uns com os outros através de uma rede 11, representada por uma nuvem. A rede 11 pode incluir vários dispositivos bem conhecidos, tal como roteadores, conexões, centralizadores, etc. e permitir que os computadores 10 comuniquem-se via meio com fios e/ou sem fios. Quando interagindo uns com os outros através da rede 11, um ou mais dentre os computadores 10 podem atuar como clientes, servidores ou pares com respeito aos outros computadores 10. Por consequência, as várias modalidades da invenção podem ser praticadas nos clientes, servidores, pares ou em combinações dos mesmos, mesmo que no entanto os exemplos específicos contidos aqui dentro não refiram-se a todos estes tipos de computadores.

[0024] Referindo-se à FIG. 2, é apresentado um exemplo de uma configuração básica para um computador 10 no qual toda ou partes da invenção descrita aqui dentro podem ser implementadas. Na sua configuração mais básica, o computador 10 tipicamente inclui pelo menos uma unidade de processamento 14 e a memória 16. A unidade de processamento 14 executa instruções que realizam tarefas de acordo com as várias modalidades da invenção. Ao realizar tais tarefas, a unidade de processamento 14 pode transmitir sinais eletrônicos para outras partes do computador 10 e para dispositivos exteriores ao computador 10 para causar algum resultado. Dependendo da configuração exata e do tipo do computador 10, a memória 16 pode ser volátil (tal como a RAM), não volátil (tal como a ROM ou a memória flash) ou alguma combinação das duas. Esta configuração mais básica é ilustrada na FIG. 2 pela linha tracejada 18.

[0025] O computador 10 pode possuir aspectos e/ou funcionalidade adicionais. Por exemplo, o computador 10 também pode incluir armazenamento adicional (armazenamento removível 20 e/ou armazenamento não removível 22) incluindo, mas não limitado a

discos ou fita magnéticos ou óticos. O meio de armazenamento do computador inclui o meio volátil, não volátil, removível e não removível implementado em qualquer método ou tecnologia para armazenamento de informação, incluindo instruções executáveis por computador, estruturas de dados, módulos de programa, ou outros dados. O meio de armazenamento do computador inclui, mas não está limitado à RAM, ROM, EEPROM, memória flash, CD-ROM, disco versátil digital (DVD) ou outro armazenamento ótico, armazenamento em cassetes magnéticos, fita magnética, disco magnético ou outros dispositivos de armazenamento magnéticos, ou qualquer outro meio que possa ser utilizado para armazenar a informação desejada e que possa ser acessado pelo computador 10. Qualquer tal meio de armazenamento do computador pode ser parte do computador 10.

[0026] O computador 10 de preferência também contém as conexões de comunicações 24 que permitem o dispositivo comunicar-se com outros dispositivos. Uma conexão de comunicação (por exemplo, uma das conexões de comunicação 24) é um exemplo de um meio de comunicação. O meio de comunicação tipicamente incorpora instruções legíveis por computador, estruturas de dados, módulos de programa ou outros dados em um sinal de dados modulado tal como uma onda portadora ou outro mecanismo de transporte e inclui qualquer meio de distribuição de informação. A título de exemplo e não de limitação, o termo “meio de comunicação” inclui meio com fios tal como uma rede com fios ou conexão direta com fios e meio sem fios tal como um meio sem fios acústico, RF, infravermelho e outros. O termo “meio legível por computador” como utilizado aqui dentro inclui tanto meio de armazenamento do computador como o meio de comunicação.

[0027] O computador 10 também pode possuir dispositivos de entrada 26 tal como um teclado, mouse, caneta, dispositivo de entrada

de voz, dispositivo de entrada por toque, etc. Os dispositivos de saída 28n tal como um vídeo 30, alto-falantes, uma impressora, etc., também podem estar incluídos. Todos estes dispositivos são bem conhecidos na técnica e não precisam ser discutidos em maiores detalhes aqui dentro.

[0028] Empacotamento da Memória Intermediária

[0029] Voltando-se agora para a FIG. 3, é apresentado um exemplo de uma rede de correio eletrônico 100 na qual a presente invenção pode ser implementada. A rede de correio eletrônico 100 da presente invenção utiliza trocas de requisição e de resposta para passar consultas e dados entre os componentes cliente e servidor na rede de correio eletrônico 100. Na prática, o desempenho de um protocolo pode ser afetado pelo mecanismo de transporte de comunicações subjacente da rede utilizado para implementar as comunicações entre os clientes e servidores em uma rede de correio eletrônico, tal como a rede de correio eletrônico 100. Por exemplo, em uma rede de correio eletrônico que utiliza chamadas de procedimento remoto (RPCs) como o mecanismo de transporte de comunicações subjacente da rede, pode ser mais eficiente fazer uma única chamada de procedimento remoto de tamanho maior (por exemplo, 32 KB) do que fazer várias chamadas de procedimento remoto de tamanho menor (por exemplo, 2 KB). Um modo conhecido para aperfeiçoar o desempenho em tal rede de correio eletrônico é colocar na memória intermediária várias requisições e/ou respostas para transmissão em uma única chamada de procedimento remoto.

[0030] Como um exemplo, a FIG. 3 apresenta uma troca de requisição e resposta entre um cliente de correio eletrônico 102 e um servidor de correio eletrônico 106, um ou ambos podendo ser configurado tal como o computador 10. Neste exemplo, o cliente de correio eletrônico 102 aloca uma memória intermediária de envio 104 e

enche a mesma com requisições, as quais podem ser uma ou mais sub-requisições ou operações remotas ROPs) a serem envidas para o servidor de correio eletrônico 106. Quando a memória intermediária de envio 104 está cheia (ou quase cheia), o cliente de correio eletrônico 102 envia o conteúdo para o servidor de correio eletrônico 106, o qual armazena as mesmas em uma memória intermediária de requisições 108. O servidor de correio eletrônico 106 lê as requisições da memória intermediária de requisições 108 e processa as requisições. O processamento de cada requisição produz um resultado na forma de uma resposta. Estas respostas podem incluir dados requisitados pelo cliente de correio eletrônico 102 (por exemplo, uma mensagem de correio eletrônico particular). O servidor de correio eletrônico 106 armazena estas respostas em uma memória intermediária de respostas 110.

[0031] De acordo com uma modalidade da presente invenção, a media que o servidor de correio eletrônico 106 processa cada requisição, ele utiliza um ponteiro para rastrear qual requisição é a próxima requisição a ser processada a partir da memória intermediária de requisições 108. Quando o servidor de correio eletrônico 106 determina que a memória intermediária de respostas 110 está cheia (por exemplo, possui menos de 8 KB restantes dos 32 KB), então o servidor de correio eletrônico 106 para de processar as requisições na memória intermediária de requisições 108. Qualquer requisição restante que não tenha sido processada (isto é, requisições incompletas) é anexada para o conteúdo da memória intermediária de respostas 110. Estas requisições incompletas e as respostas para as requisições completas são enviadas para uma memória intermediária de recepção 112 no cliente de correio eletrônico 102.

[0032] Em uma modalidade da presente invenção, o cliente de correio eletrônico 102 é capaz de designar o amanho de qualquer uma

das memórias intermediárias 104, 108, 110, 112. O tamanho de uma resposta é tipicamente maior do que o tamanho de uma requisição. Por esta razão, o tamanho da memória intermediária de respostas 110 e da memória intermediária de recepção 112 (coletivamente, as "memórias intermediárias de respostas 110 e 112") podem ser designadas pelo cliente de correio eletrônico como sendo maior do que o tamanho da memória intermediária de envio 104 e da memória intermediária de requisições 108 (coletivamente, as "memórias intermediárias de requisições 104 e 108").

[0033] Os sistemas de rede de correio eletrônico da técnica anterior dos quais os inventores estão cientes não eram capazes desta função, pelo fato de que eles utilizam somente uma única memória intermediária no cliente de correio eletrônico e no servidor de correio eletrônico. Apesar da seção de fundamento do pedido provisório baseado no qual esta especificação reivindica benefício descrever uma rede de correio eletrônico na qual cada um dentre o cliente de correio eletrônico e o servidor de correio eletrônico possui duas memória intermediárias, o os requerentes não estão cientes de qualquer rede de correio eletrônico anterior à presente invenção que utilizasse mais do que uma única memória intermediária em cada um.

[0034] Algumas redes de correio eletrônico que utilizam memórias intermediárias, por exemplo, a rede de correio eletrônico 100 apresentada na FIG. 3, podem empregar uma modo de transferência rápida entre um cliente (por exemplo, o cliente de correio eletrônico 102) e um servidor (por exemplo, o servidor de correio eletrônico 106). O modo de transferência rápida inclui requisições, tal como os ROPs, por um cliente que são divididas em pelo menos duas categorias: as requisições que resultam em uma inicialização de uma fonte de dados de transferência rápida no servidor e as requisições que resultam na transferência eficiente de dados a partir da fonte de dados de

transferência rápida para o cliente. A fonte de dados de transferência rápida pode ser, por exemplo, uma tabela da base de dados. A fonte de dados de transferência rápida serve como um armazenamento temporário pronto de dados que permite que requisições posteriores para os dados sejam servidas com menos demora do que de outro modo seria possível. Algumas vezes, a segunda categoria de requisição do modo de transferência rápida procura alcançar a transferência eficiente de dados por explicitamente especificar o tamanho da resposta. Como um exemplo, o tamanho da resposta pode ser estabelecido para o tamanho de toda a memória temporária de recepção 112, menos as atividades de suporte da resposta.

[0035] A FIG. 4A apresenta uma operação de transferência rápida possuindo pelo menos dois ciclos de requisição – resposta. Em uma primeira requisição 401, um ROP (por exemplo, FXPrepare) inicializa uma fonte de dados de transferência rápida no servidor de correio eletrônico 106. NO servidor de correio eletrônico 106, somente o FXPrepare é processador (isto é, a fonte de dados de transferência rápida é inicializada) e seu resultado é retornado em uma primeira resposta 402. Em uma segunda requisição 403, um ROP (por exemplo, FXGetBuffer) requisita ao servidor de correio eletrônico 106 para encher a memória intermediária de respostas 110 a partir da fonte de dados rápida. O servidor de correio eletrônico 106 esvazia a fonte de dados rápida dentro da memória intermediária de respostas 110 e retorna o resultado em uma segunda resposta 404. Se a memória intermediária de respostas 110 para o servidor de correio eletrônico 106 encher antes que a fonte de dados rápida seja esvaziada, ROPs FXGetBuffer adicionais podem ser requeridas.

[0036] A FIG. 4B apresenta uma operação de transferência rápida possuindo somente um único ciclo de requisição – resposta. Em uma primeira requisição 405, tanto o FXPrepare quanto o FXGetBuffer são

processados pelo servidor de correio eletrônico e os resultados de ambas operações são retornadas em uma primeira resposta 406. O resultado de FXPrepare fica disponível; para o FXGetBuffer no servidor de correio eletrônico 106 porque parte de cada memória intermediária é explicitamente definida como uma tabela de dados compartilhada.

[0037] É desejável reduzir o número de ciclos de requisição – resposta porque tal redução resulta em uma transferência mais eficiente dos dados. Uma operação de transferência rápida possuindo mais do que somente um único ciclo de requisição – resposta pode ocorrer quando a memória intermediária de respostas 110 estiver muito cheia para manter o resultado de um ROP FXGetBuffer.

[0038] Voltando-se agora para a FIG. 5, é apresentado um exemplo de conteúdo 120 da memória intermediária de envio do cliente 104. Neste exemplo, a memória temporária de envio 104 contém um cabeçalho da chamada de procedimento remota (RPC) 122 e uma série de requisições 124.

[0039] De acordo com um aspecto da presente invenção, o cabeçalho da RPC 122 pode incluir um bit de compactação 126 e um bit de ofuscação 128. O bit de compactação 126 indica se o servidor de correio eletrônico 106 é ou não para compactar as respostas para as requisições. Outras informações podem ser proporcionadas dentro do conteúdo 120 para indicar que o servidor de correio eletrônico 106 é para compactar as respostas. A compactação pode nem sempre ser desejada. Por exemplo, se o cliente tiver uma conexão de alta velocidade com baixa latência e não tiver potência de processamento de reserva suficiente para de forma eficiente executar a descompactação, o cliente pode enviar a requisição com uma indicação de que a compactação não é desejada. Alternativamente, se o cliente tiver potência de processamento suficiente e a conexão com o servidor tiver largura de banda pequena, o cliente pode indicar para o

servidor que ele deseja a compactação (por exemplo, estabelecer o bit indicador de compactação no cabeçalho).

[0040] O bit de ofuscação 128 indica se o servidor de correio eletrônico 106 é ou não para ofuscar as requisições. A ofuscação é uma operação simples executada para impedir os dados de serem enviados como texto legível claramente através da rede. Um exemplo de ofuscação é executar o XOR (um método de ofuscação conhecido) junto às requisições antes que elas sejam enviadas. Em algumas modalidades, a criptografia pode ser utilizada em vez da ofuscação. Novamente, outras informações podem ser incluídas dentro do conteúdo 120, as quais indicam que as requisições são para ser ofuscadas ou criptografadas.

[0041] Como apresentado na FIG. 5, em algumas modalidades, o cliente de correio eletrônico 102 pode ser configurado para incluir uma requisição especial 130 dentro do conteúdo 120 que instrui o servidor de correio eletrônico 106 para responder à requisição do cliente utilizando o encadeamento, descrito abaixo.

[0042] Voltando-se agora para a FIG. 6, um fluxograma é proporcionado ilustrando um método para transferir dados entre um cliente e um servidor de acordo com uma modalidade da presente invenção. Começando na etapa 600, o servidor de correio eletrônico 106 recebe uma pluralidade de requisições a partir de um cliente (por exemplo, as requisições 124).

[0043] De acordo com uma modalidade da presente invenção, o cliente de correio eletrônico 102 pode requisitar o encadeamento ou o não encadeamento para o servidor de correio eletrônico 106 para enviar as respostas. Na etapa 602, o servidor de correio eletrônico 106 examina as requisições 124 para determinar se as requisições incluem uma requisição para encadeamento (por exemplo, a requisição especial 130). Se não, então, a etapa 602 ramifica-se para a etapa

604, onde o servidor de correio eletrônico 106 começa a construir as respostas para as requisições 124. Um exemplo de um processo para construir uma resposta não utilizando encadeamento é apresentado na FIG. 8 e as etapas na FIG. 7 serão aplicadas para este exemplo nesta descrição.

[0044] Na etapa 604 (FIG. 6), o servidor de correio eletrônico 106 cria um cabeçalho 140. Na etapa 606, as respostas 142 (FIG. 6) para as requisições 124 são recuperadas e são armazenadas na memória intermediária de respostas 110. Uma vez que o servidor de correio eletrônico 106 tenha gerado repostas suficientes de modo que as respostas 142 e o cabeçalho 140 encham ou quase encham a memória intermediária de respostas 110, o servidor de correio eletrônico 106 para de processar as requisições. Se a memória intermediária de respostas 110 está cheia ou quase cheia pode ser definido pelo servidor de correio eletrônico 106 e/ou pelo cliente de correio eletrônico 102. Como um exemplo, a memória intermediária de respostas 110 pode ser considerada cheia quando ela tiver menos do que 8 k restantes de uma memória intermediária inicial com 32 K.

[0045] Se o cliente de correio eletrônico 102 indicou que ele suporta a compactação (por exemplo, por de forma apropriada estabelecer o bit de compactação 126), o servidor de correio eletrônico compacta as respostas dentro da memória intermediária de respostas 110 em um conjunto compactado 144 de respostas 142 (FIG. 7) na etapa 608. De forma similar, também na etapa 608, se o cliente de correio eletrônico 102 indicou que ele suporta a ofuscação (por exemplo, por de forma apropriada estabelecer o bit de ofuscação 128), então, o servidor de correio eletrônico 106 pode ofuscar ou criptografar as respostas 142 como instruído).

[0046] Quaisquer requisições que não tenham sido processadas são anexadas para as respostas na memória intermediária de

requisições 108 na etapa 610. Estas respostas não processadas podem ser colocadas na memória intermediária não utilizada após a compactação, geralmente apresentada na memória 146 na FIG. 6. O servidor de correio eletrônico 106 então envia as respostas e as requisições não completadas para o cliente de correio eletrônico 102 na etapa 612.

[0047] Como pode ser visto pelo exemplo descrito acima e apresentado na FIG. 7, a memória não utilizada após a compactação (isto é, a memória 146) em uma resposta que não é encadeada, pode ser substancial. De acordo com um aspecto da presente invenção, a quantidade de memória não utilizada pode ser minimizada utilizando-se um processo de encadeamento. Entretanto, o método que não é de encadeamento até aqui descrito pode ser útil, por exemplo, onde um cliente de correio eletrônico 102 não deseja o encadeamento, por exemplo, em uma ROP que não requisita o modo de transferência rápida.

[0048] Se o cliente de correio eletrônico 102 indicar que o servidor de correio eletrônico 106 deve utilizar o encadeamento, então a etapa 602 ramifica-se para a etapa 614, onde o servidor de correio eletrônico 106 cria um primeiro cabeçalho 150 (FIG. 7). A FIG. 8 apresenta um exemplo de um processo para construir uma resposta para o encadeamento e é utilizada com a descrição das etapas 614 até 620.

[0049] Na etapa 616, o servidor de correio eletrônico 106 recupera e enche a memória intermediária de respostas 110 com as respostas 152. Novamente, a memória intermediária de respostas 110 pode ser considerada cheia uma vez que ela alcance um limite predefinido. Ela pode pegar somente uma resposta para encher a memória intermediária de respostas 110, entretanto, como utilizado aqui dentro, um “conjunto de respostas” significa uma ou mais respostas. Na etapa 618, uma vez que a memória intermediária de respostas 110 esteja

cheia ou quase cheia, o servidor de correio eletrônico 106 compacta e/ou ofusca as respostas dentro da memória intermediária de respostas 110 de acordo com as instruções do cliente de correio eletrônico 102 (por exemplo, pelo bit de compactação 126 e/ou pelo bit de ofuscação 128). Um conjunto compactado 154 de respostas é criado, deixando uma grande seção 156 de memória não utilizada na memória intermediária de requisições 108.

[0050] Após a compactação e/ou a ofuscação, na etapa 620, é feita uma determinação de se respostas adicionais podem encaixar-se dentro da memória intermediária de respostas 110. Novamente, se respostas adicionais encaixam-se pode ser definido pelo cliente de correio eletrônico 102 ou pelo servidor de correio eletrônico 106. Entretanto, após a primeira compactação, é antecipado que espaço adicional ficará disponível. Se espaço adicional ficar disponível, então o processo retornar para a etapa 614, onde o servidor de correio eletrônico 106 cria e anexa um segundo cabeçalho 158 (FIG. 8) e começa a processar as requisições mais uma vez (etapa 616).

[0051] Uma vez que a memória intermediária de respostas 110 esteja cheia ou quase cheia com respostas, o servidor de correio eletrônico 106 compacta e/ou ofusca a respostas mais recentemente adicionadas 160 na etapa 618. Novamente é feita uma determinação na etapa 620 quanto a se existe espaço deixado para respostas adicionais. Se existir, o processo novamente retorna para a etapa 614, onde um terceiro cabeçalho é anexado e o servidor de correio eletrônico 106 novamente enche a memória intermediária de respostas 110 com respostas e compacta e/ou ofusca as respostas (etapas 616 e 618). Este processo é repetido até que todas as requisições tenham sido completadas ou a memória intermediária de respostas 110 esteja cheia ou quase cheia de cabeçalhos e de respostas compactadas correspondentes. Uma vez que a memória intermediária de respostas

110 esteja cheia ou quase cheia de respostas e cabeçalhos compactados (apresentados na parte de baixo da FIG. 8), a etapa 620 ramifica-se para a etapa 610, onde o servidor de correio eletrônico 106 anexa quaisquer requisições não completadas (se houverem) e envia o conteúdo da memória intermediária de respostas 110 para o cliente de correio eletrônico 102.

[0052] Um cliente de correio eletrônico 102 que recebe o conteúdo da memória intermediária de respostas 110 em sua memória intermediária de recepção 112 pode então processar cada um dos conjuntos de respostas entre os cabeçalhos. Se os conjuntos de respostas estiverem compactados e/ou ofuscados, então o cliente de correio eletrônico 102 pode descompactar ou reverter a ofuscação. Em tal caso, o cliente de correio eletrônico 102 ainda possui uma pluralidade de conjuntos de respostas que ele pode então processar.

[0053] Como pode ser visto pelas diferenças entre os dados enviados no processo que não é de encadeamento na FIG. 7 e o processo de encadeamento da FIG. 6, o encadeamento permite que vários pares de cabeçalho/resposta sejam encadeados ou empacotados juntos e que sejam enviados em um “lote”, desse modo potencialmente reduzindo o número de idas e voltas entre o cliente de correio eletrônico 102 e o servidor de correio eletrônico 106. Este processo é referido aqui dentro como “encadeamento” ou “empacotamento” das respostas. O encadeamento pode ser muito mais eficiente para uma rede, especialmente em um ambiente com largura de banda pequena. De acordo com uma modalidade da invenção, o servidor de correio eletrônico 106 pode proporcionar o encadeamento com as requisições do modo de transferência rápida e pode não proporcionar o encadeamento com uma requisição que não é do modo de transferência rápida.

[0054] Voltando-se agora para a FIG. 9, um exemplo mais

detalhado de uma memória intermediária de respostas 159 é apresentado. Neste exemplo, cada cabeçalho 161_1, 161_2, ..., 162_N inclui um ponteiro 162_1, ..., 162_N para o próximo cabeçalho na memória intermediária. Alternativamente, o cabeçalho 161 pode incluir o tamanho compactado da resposta correspondentes. Em qualquer caso, este aspecto permite que o cliente de correio eletrônico 102 mais facilmente decodifique o lote compactado quando recebido, porque o cliente de correio eletrônico 102 irá saber o tamanho de cada respostas e a localização do começo da próxima resposta.

[0055] Cada cabeçalho 161 também pode incluir a informação 164_1, ..., 164_N, por exemplo, na forma de um arquivo de bits, que indica se o cabeçalho 161 corresponde à última resposta na memória intermediária. O cabeçalho 161 também pode incluir outras informações, tal como o tamanho descompactado da informação de resposta correspondente.

[0056] Observe que o servidor de correio eletrônico 106 pode receber e processar requisições a partir de vários clientes de correio eletrônico 102 em paralelo. Para este fim, um único cliente de correio eletrônico 102 é apresentado simplesmente para simplificar as figuras e a explicação.

[0057] Utilizando Memórias Intermediárias de Respostas Maiores

[0058] Como descrito acima, o cliente de correio eletrônico 102 pode ser configurado para informar para o servidor de correio eletrônico qual tamanho das memórias intermediárias de requisições e/ou de respostas será utilizado. Por exemplo, em uma modalidade da presente invenção, as memórias intermediárias de requisições 104 e 108 são cada uma com 32 KB e o tamanho ótimo das memórias intermediárias de respostas 110 e 112 é cada uma com 92 KB, uma proporção de 3 para 1.

[0059] Apesar do cliente de correio eletrônico 102 poder

especificar memórias intermediárias de respostas maiores 110 e 112, o cliente de correio eletrônico 102 pode ser configurado para trabalhar com pedaços de dados de respostas que são menores do que o tamanho real das memórias intermediárias de respostas 110 e 112. Por exemplo, memórias intermediárias de 96 K podem ser especificadas para as memórias intermediárias de respostas 110 e 112, mas o cliente de correio eletrônico 102 pode desejar que todos os pedaços de dados de respostas sejam 32 K ou menos. O empacotamento ou encadeamento da presente invenção permite que tal sistema seja operacional.

[0060] Uma modalidade de um método para permitir esta função é apresentada no fluxograma na Fig. 10. Começando na etapa 1000, o cliente de correio eletrônico envia um conjunto de requisições para o servidor de correio eletrônico 106, junto com a informação definindo um tamanho da memória intermediária de respostas (por exemplo, 96 k) e a informação a cerca do tamanho de um pedaço de dados que o cliente está configurado para processar. Na etapa 1002, o servidor de correio eletrônico 106 cria um quadro dentro da memória intermediária de respostas 110 igual ao tamanho do pedaço de dados definido pelo cliente. O servidor de correio eletrônico 106 então grava, na etapa 1004, um cabeçalho para o quadro dentro do quadro na memória intermediária de respostas 110. Na etapa 1006, o servidor de correio eletrônico 106 começa a processar as respostas até que elas encham ou quase encham o quadro. O conjunto de respostas pode ou não ser compactado ou ofuscado na etapa 1008.

[0061] Então. é feita uma determinação na etapa 1010 de se a memória intermediária de respostas 110 está cheia ou não. Tipicamente, a memória intermediária de respostas 110 não estará cheia após o primeiro processamento das respostas. Se a memória intermediária de respostas 110 não estiver cheia, o processo retorna

para a etapa 1002, onde o servidor de correio eletrônico 106 cria um novo quadro começando no fim do conjunto de respostas recém-processado. Um ponteiro pode ser utilizado de modo que o servidor de correio eletrônico 106 sabe onde começar este próximo quadro. O novo quadro também terá o tamanho de um pedaço de dados que o cliente de correio eletrônico 102 pode manipular, se existir espaço suficiente dentro da memória intermediária de respostas 110. Na etapa 1004, o servidor de correio eletrônico 106 grava o próximo cabeçalho no novo quadro. O processo então continua para a etapa 1010.

[0062] Uma vez que a memória intermediária de respostas 110 esteja cheia (ou todas as requisições tenham sido processadas, o que acontecer primeiro).o processo ramifica-se para a etapa 1012, onde o servidor de correio eletrônico 106 copia as requisições não processadas restantes na memória intermediária de respostas 110. Na etapa 1014, o servidor de correio eletrônico 106 envia o conteúdo da memória intermediária de respostas 110 para o cliente de correio eletrônico 102.

[0063] Um cliente de correio eletrônico 102 que recebe o conteúdo da memória intermediária de respostas 110 em sua memória intermediária de recepção 112 pode então processar cada um dos pedaços de dados (conjuntos de respostas) entre os cabeçalhos. Se as respostas não estiverem compactadas ou ofuscadas, então o cliente de correio eletrônico 102 pode processar cada um dos conjuntos de respostas entre os cabeçalhos como eles são. Os conjuntos de respostas serão iguais ou menores do que os pedaços de dados definidos pelo cliente de correio eletrônico 102, de modo que o cliente de correio eletrônico 102 deve estar apto a de forma apropriada manipular os conjuntos de dados. Se os conjuntos de respostas estiverem compactados e/ou ofuscados, então o cliente de correio eletrônico 102 pode descompactar ou reverter a ofuscação. Em tal

caso, o cliente de correio eletrônico 102 ainda possui uma pluralidade de conjuntos de respostas que são cada um menor ou igual ao tamanho dos pedaços de dados que ele pode manipular.

[0064] Enganando o Servidor para Processar mais Requisições

[0065] Quando o servidor de correio eletrônico 106 tiver completado o processamento de um conjunto de requisições, o servidor de correio eletrônico 106 pode ter que ser "enganado" para continuar a processar requisições adicionais. Por exemplo, os servidores de correio eletrônico existentes tipicamente são configurados para processar as requisições e proporcionar as respostas até um certo tamanho (por exemplo, 32 KB), o tamanho normalmente sendo ditado pelo cliente de correio eletrônico 102. Após este processamento, os servidores de correio eletrônico existentes são configurados para enviar uma resposta indicando que as respostas estão prontas (uma resposta FXPrepare), ou para automaticamente enviar as respostas (uma resposta FXGetBuffer). Entretanto, utilizando a compactação revelada aqui dentro, pode ser desejável para o servidor de correio eletrônico igualmente processar mais requisições em uma resposta FXGetBuffer de modo a preencher espaço adicional dentro da memória intermediária de respostas 110. O espaço adicional pode ser criado pela compactação. Alternativamente, a modalidade com a memória intermediária maior descrita acima pode ter espaço adicional após o processamento de um de seus quadros.

[0066] Uma modalidade de um método para manipular esta situação é apresentada na FIG. 11. Começando na etapa 1100, é feita uma determinação de se existe espaço para mais respostas e se existe mais requisições para processar. Se não, então a etapa 1100 ramifica-se para a etapa 1102, onde o servidor de correio eletrônico 106 envia as respostas para o cliente de correio eletrônico 102. Se a condição do servidor de correio eletrônico 106 após proporcionar um conjunto de

respostas indicar que existe mais para processar e espaço para processar as mesmas, então a etapa 1100 ramifica para a etapa 1104, onde o servidor de correio eletrônico 106 gera uma requisição dentro do limite "falsa" (por exemplo, na requisição FXGetBuffer "falsa). Esta pseudo requisição RPC (chamada de procedimento remoto) dentro do limite é então processada pelo servidor de correio eletrônico 106 como se ela tivesse sido recebida a partir do cliente de correio eletrônico 102. A RPC é "pseudo" pelo fato de que ela não é realmente enviada a partir de um computador remoto, mas ao invés disso é enviada a partir de dentro do servidor. A memória intermediária fora do limite para esta pseudo RPC é estabelecida para ser a parte restante da memória intermediária fora do limite original após a compactação (a qual pode ser limitada pelos quadros definidos acima). O servidor de correio eletrônico 106 então processa as novas respostas na etapa 1106, como descrito acima.

[0067] O servidor de correio eletrônico 106 continua a repetir este processo até que ele atinja um dos seguintes critérios: não exista nada deixado na requisição dentro do limite para processar, o tamanho da memória intermediária fora do limite restante é menor do que um limite predeterminado (por exemplo 8 K), um número máximo de memórias intermediárias estão encadeadas (por exemplo, 64),ou existe um erro grave.

[0068] Cada conjunto de respostas empacotado possui seu próprio cabeçalho com seus próprios indicadores. Um pacote poderia ser compactado e o próximo ofuscado, ou, alternativamente, nenhum pacote pode ser compactado ou ofuscado. Para cada nova memória intermediária na resposta encadeada, os indicadores são honrados.

[0069] Exemplo de Conteúdo de uma Memória Intermediária

[0070] Abaixo está uma vista detalhada de um exemplo com duas memórias intermediárias encadeadas na memória intermediária de

respostas dentro do limite:

Memória Intermediária 1			Memória Intermediária 2		
	Dados Compactados			Dados Compactados	
RpcHeaderExt wFlags frehComp wSize WSizeActual	dados resp rop	Tabela HSOT	RpcHeaderExt wFlags frheComp frheLast	dados resp rop	Tabela HSOT
	wSize			wSize	

[0071] Como ilustrado acima, cada memória intermediária de respostas empacotadas possui sua própria Tabela HSOT (Tabela de Operação de Manipulação para Armazenamento). Uma HSOT é um identificador em cada requisição da rede de sobre qual objeto no servidor de correio eletrônico 106 ela está atuando. Isto é uma tabela de HSOT's, cujas entradas correspondem aos objetos internos de armazenamento. Cada operação na memória intermediária de requisições 108 contém um índice dentro desta tabela para identificar junto a qual objeto de armazenamento a operação se aplica. O servidor de correio eletrônico 106 também irá adicionar entradas para esta tabela se um novo objeto de armazenamento interno for criado durante uma operação. A tabela HSOT não deve diferir em valor da primeira na lista, mas somente irá conter as entradas HSOT para todos as HSOT's até e incluindo a utilizada na ROP FXGetBuffer, descrita acima. Isto é primariamente uma decisão de implementação a medida que ela irá reduzir a complexidade no servidor de correio eletrônico 106. A tabela HSOT geralmente não é mais do que um par de DWORDS, de modo que isto não deve consumir largura de banda significativa.

[0072] Observe que isto é simplesmente um exemplo e os detalhes de uma implementação diferente irão variar. Por exemplo, alguns dos indicadores podem ser interpretados de forma diferente baseado em

quais aspectos o cliente de correio eletrônico 102 e/ou o servidor de correio eletrônico 106 suporta. Por exemplo, se o cliente de correio eletrônico 102 indicar que ele suporta as memórias intermediárias empacotadas e/ou compactadas, mas o servidor de correio eletrônico 106 não suportar uma ou ambas estas, o servidor de correio eletrônico 106 pode ignorar os indicadores correspondentes.

[0073] Como descrito acima, duas novas chaves de registro podem ser adicionadas no cliente de correio eletrônico 102 para alternar se o cliente de correio eletrônico 102 deve ou não deve utilizar as respostas encadeadas e qual deve ser o tamanho da memória intermediária fora do limite (por exemplo 32 K <= tamanho <= 128 K). Isto também poderia ser utilizado para requisitar a compactação. Estes aspectos poderiam ser habilitados por valor preestabelecido, ou desativados por valor preestabelecido dependendo da implementação.

[0074] Em algumas implementações, as memórias intermediárias dentro do limite indo para o servidor de correio eletrônico podem ser empacotadas. Estas implementações podem implementar um processo similar no cliente de correio eletrônico 102 para manipular o empacotamento como descrito acima para o servidor de correio eletrônico 106. Entretanto, desde que a baixa de itens tipicamente é mais consumidora de tempo do que o levantamento, algumas implementações podem enviar memórias intermediárias dentro do limite para o servidor de correio eletrônico 106 sem empacotamento.

[0075] Ao mesmo tempo que esta revelação focou as implementações de cliente e de servidor de correio eletrônico, os sistemas e métodos revelados aqui dentro também podem ser aplicáveis junto a outros tipos de aplicações. Por exemplo, as técnicas reveladas aqui dentro podem ser aplicadas para sincronizar a informação entre um cliente e um servidor de mapeamento em uma aplicação de mapeamento tal como o MapPoint da Microsoft.

Adicionalmente, ao mesmo tempo que as modalidades dos sistemas e métodos foram descritas em termos de aplicações de software, os com conhecimento na técnica irão apreciar que os sistemas podem ser implementados em hardware, software, ou em uma combinação de hardware e software. Ao mesmo tempo que os exemplos focaram conexões discadas, outros tipos de conexões de rede também são contempladas (por exemplo, sem limitação, LAN, sem fios, ISDN e DSL).

[0076] Portanto, pode ser visto que um novo e útil sistema e método para comunicação entre aplicações cliente e servidor utilizando o empacotamento de memória intermediária foi proporcionado. Em vista das várias modalidades possíveis junto as quais os princípios desta invenção podem ser aplicados, deve ser reconhecido que as modalidades descritas aqui dentro com respeito às figuras do desenho são tencionadas para serem somente ilustrativas e não devem ser consideradas como limitando o escopo da invenção. Por exemplo, os com conhecimento na técnica irão reconhecer que os elementos das modalidades ilustradas apresentados em software podem ser implementados em hardware e vice versa ou que as modalidades ilustradas podem ser modificadas em disposição e detalhe sem sair do espírito da invenção. Portanto, a invenção como descrita aqui dentro contempla todas tais modalidades como podem surgir dentro do escopo das reivindicações e equivalentes seguintes da mesma.

REIVINDICAÇÕES

1. Método para a transmissão de dados por um servidor de e-mail caracterizado pelo fato de que compreende:

a) receber uma pluralidade de pedidos (124) incluindo um pedido de um encadeamento (130) de respostas aos pedidos;

b) montar um primeiro conjunto de respostas (152) para um cliente de e-mail (102), enquanto o tamanho das respostas for inferior a um limite pré-definido de um armazenador temporário de resposta (110);

c) comprimir e ofuscar o primeiro conjunto de respostas dentro do armazenador temporário de resposta (110);

d) anexar um cabeçalho (150) ao primeiro conjunto de respostas (152);

e) repetir as etapas (b), (c) e (d) para um ou mais conjuntos de respostas enquanto o tamanho dos conjuntos de respostas compactados e ofuscados e dos cabeçalhos forem menores do que o limite pré-definido do armazenador temporário de resposta (110); e

f) enviar os conjuntos de respostas (154) compactados e ofuscados e os cabeçalhos (150) juntos.

2. Método, de acordo com a reivindicação 1, caracterizado pelo fato de que compreende ainda, antes da etapa (e), gerar um pedido de entrada para o processamento de pedidos.

3. Método, de acordo com o a reivindicação 2,

caracterizado pelo fato de que o pedido de entrada é uma pseudo chamada de procedimento remoto.

4. Método, de acordo com qualquer uma das reivindicações 1 a 3, caracterizado pelo fato de que a transmissão de dados pelo servidor de e-mail (106) é realizada através da transferência de dados entre um servidor de e-mail (106) e um cliente de correio (102).

5. Método, de acordo com qualquer uma das reivindicações 1 a 4, caracterizado pelo fato de que a pluralidade de pedidos (124) é recebida a partir de um cliente de e-mail (102).

6. Método, de acordo com qualquer uma das reivindicações 1 a 5, caracterizado pelo fato de que os conjuntos de respostas e cabeçalhos são enviados em conjunto para um cliente de e-mail (102).

7. Método, de acordo com qualquer uma das reivindicações 1 a 6, caracterizado pelo fato de que a informação é recebida a definir um tamanho de um armazenador temporário.

8. Método, de acordo com qualquer uma das reivindicações 1 a 7, caracterizado pelo fato de que a informação é recebida para criar conjuntos de respostas com um dado tamanho.

9. Método, de acordo com qualquer uma das reivindicações 1 a 8, caracterizado pelo fato de que compreende ainda a etapa de:

receber informações sobre o tamanho de uma fatia de dados que o cliente de e-mail é configurado para

processar e criar um quadro dentro de um armazenador temporário de resposta igual ao tamanho da fatia de dados definida pelo cliente de e-mail.

10. Método, de acordo com qualquer uma das reivindicações 1 a 9, caracterizado pelo fato de que cada cabeçalho (161) inclui um ponteiro (162) para o próximo cabeçalho (161).

11. Método, de acordo com qualquer uma das reivindicações 1 a 10, caracterizado pelo fato de que compreende a etapa de ofuscar ou encriptar os pedidos antes de eles serem enviados, em que, opcionalmente, bits indicadores para estas características podem ser incluídas no cabeçalho.

12. Método, de acordo com qualquer uma das reivindicações 1 a 11, caracterizado pelo fato de que compreende armazenar as respostas em um armazenador temporário de resposta (110).

13. Sistema de servidor de e-mail caracterizado pelo fato de que compreende uma memória (16) e uma unidade de processamento (14) configurada para executar instruções que, quando executadas, fazem com que o sistema servidor de e-mail execute qualquer o método conforme definido e qualquer uma das reivindicações 1 a 12.

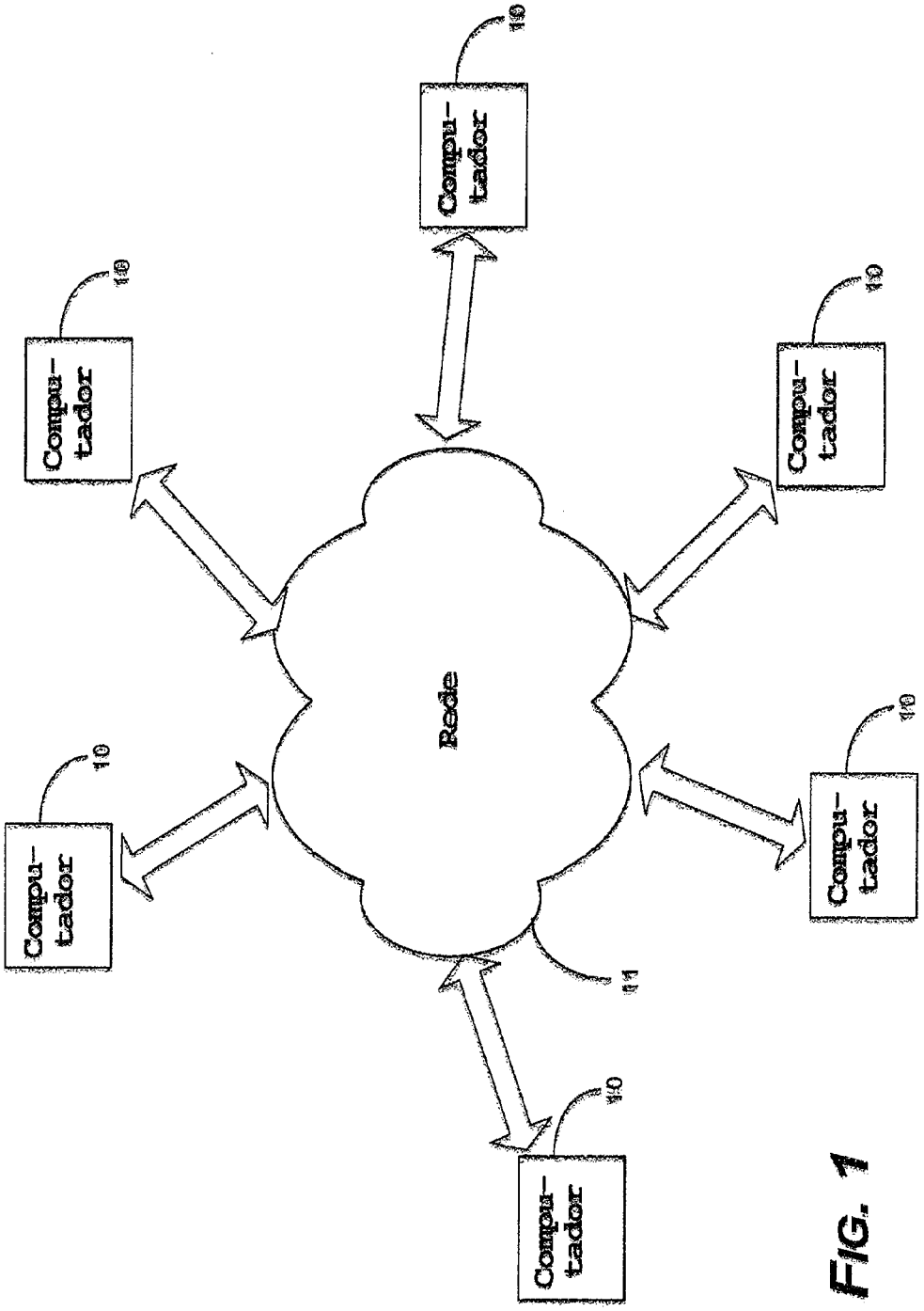

Compu-
tador
10
Compu-
tador
10
Compu-
tador
10
Compu-
tador
10
Compu-
tador
10
Compu-
tador
10
Rede
11

FIG. 1

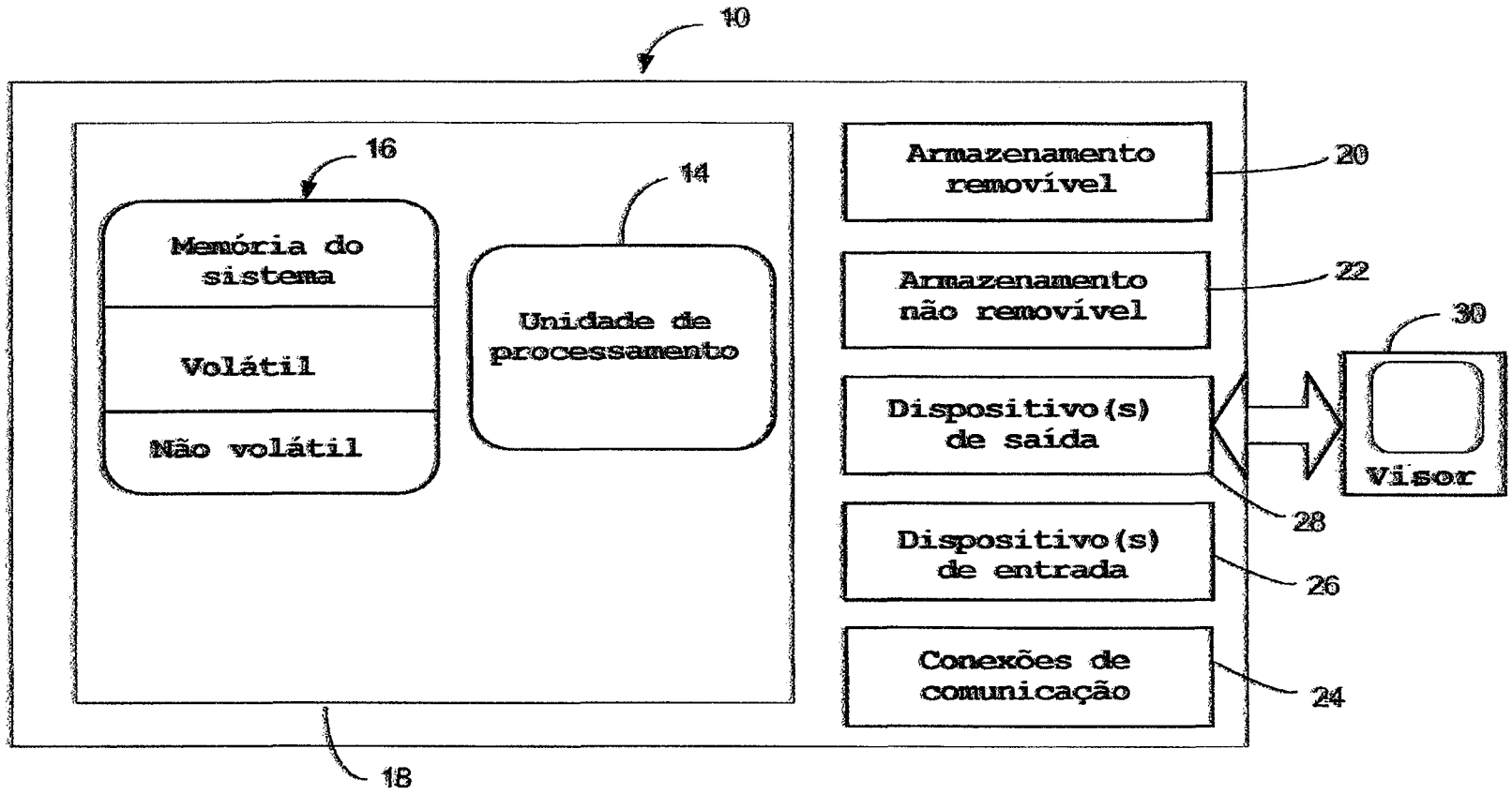

10
16
Memória do sistema
Volátil
Não volátil
14
Unidade de processamento
Armazenamento removível
20
Armazenamento não removível
22
Dispositivo(s) de saída
30
Visor
28
Dispositivo(s) de entrada
26
Conexões de comunicação
24
18

FIG. 2

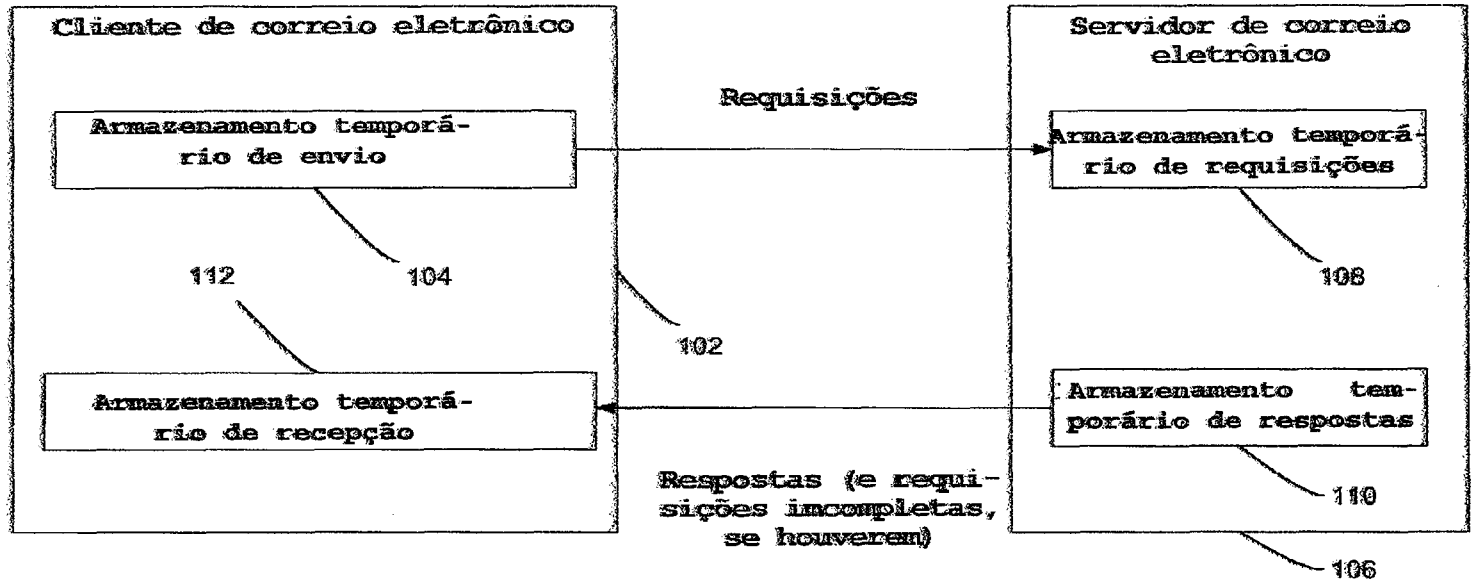

Cliente de correio eletrônico
Servidor de correio eletrônico
Requisições
Armazenamento temporário de envio
Armazenamento temporário de requisições
104
108
112
102
Armazenamento temporário de recepção
Armazenamento temporário de respostas
110
Respostas (e requisições incompletas, se houverem)
106

FIG. 3

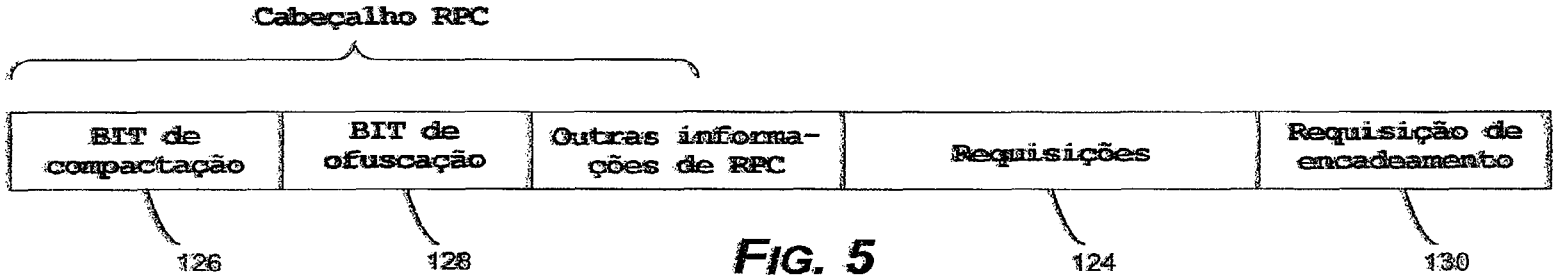

Cabeçalho RPC
BIT de compactação
BIT de ofuscação
Outras informações de RPC
Requisições
Requisição de encadeamento
126
128
124
130

FIG. 5

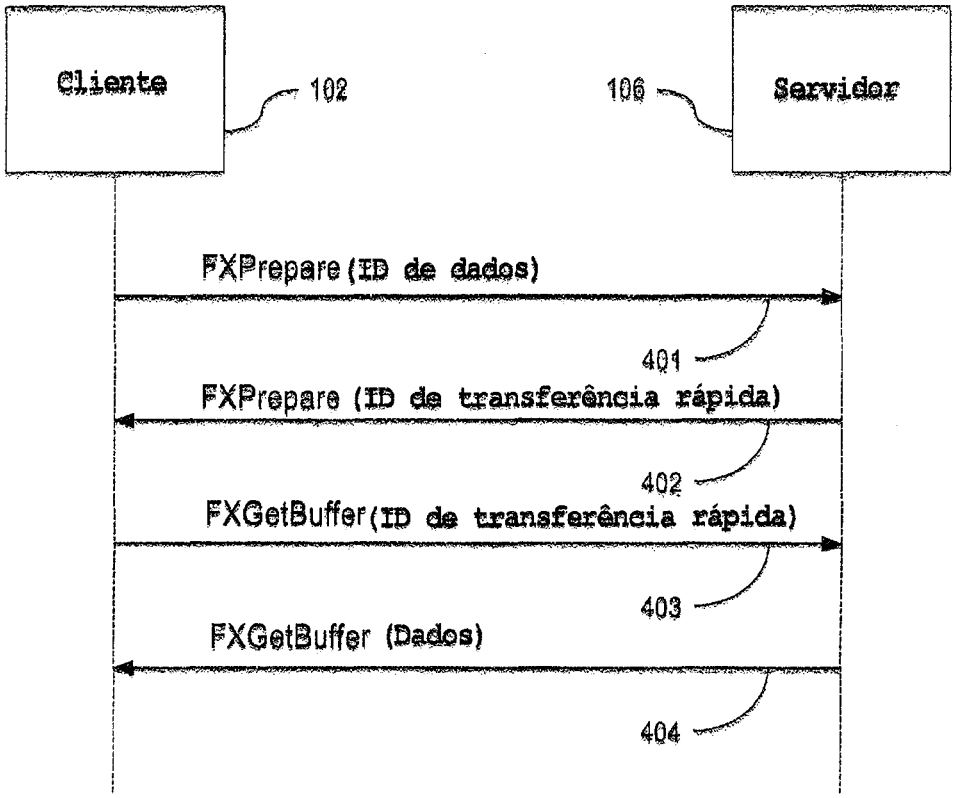

Cliente
102
106
Servidor
FXPrepare (ID de dados)
401
FXPrepare (ID de transferência rápida)
402
FXGetBuffer (ID de transferência rápida)
403
FXGetBuffer (Dados)
404

FIG. 4A

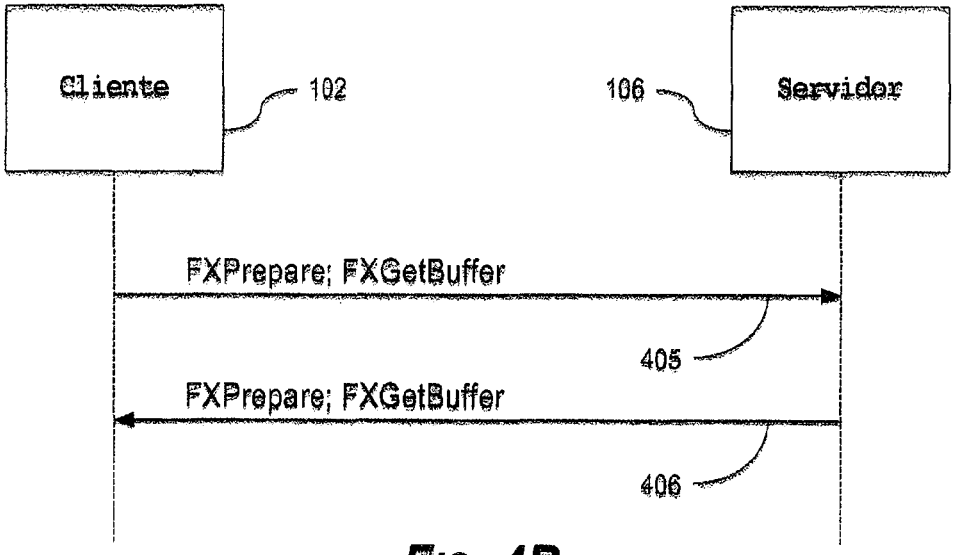

Cliente
102
106
Servidor
FXPrepare; FXGetBuffer
405
FXPrepare; FXGetBuffer
406

FIG. 4B

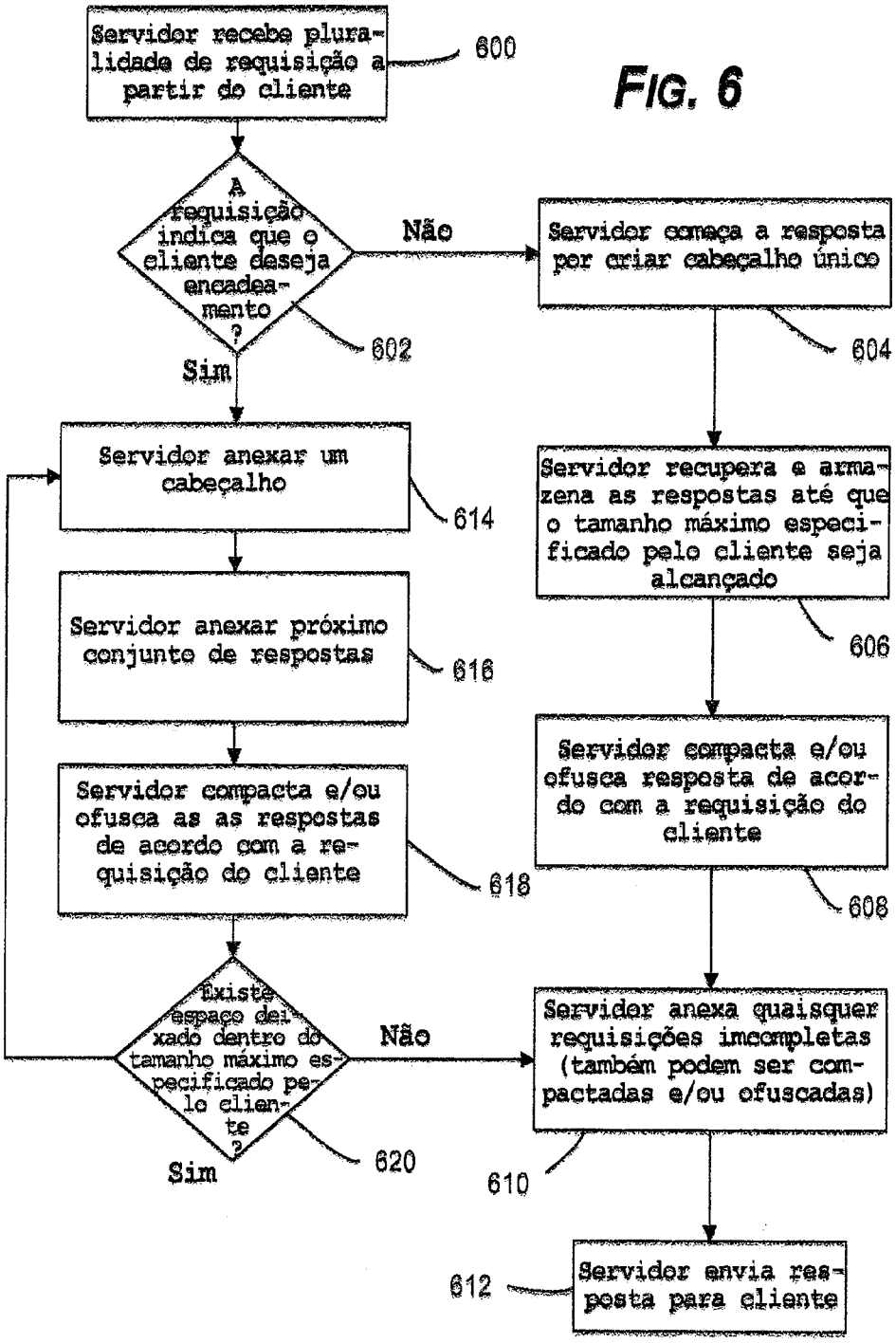

Servidor recebe pluralidade de requisição a partir do cliente
600
A requisição indica que o cliente deseja encadeamento ?
602
Não
Sim
Servidor começa a resposta por criar cabeçalho único
604
Servidor recupera e armazena as respostas até que o tamanho máximo especificado pelo cliente seja alcançado
606
Servidor compacta e/ou ofusca resposta de acordo com a requisição do cliente
608
Servidor anexa quaisquer requisições incompletas (também podem ser compactadas e/ou ofuscadas)
610
Servidor envia resposta para cliente
612
Servidor anexar um cabeçalho
614
Servidor anexar próximo conjunto de respostas
616
Servidor compacta e/ou ofusca as as respostas de acordo com a requisição do cliente
618
Existe espaço deixado dentro do tamanho máximo especificado pelo cliente ?
620
Não
Sim

FIG. 6

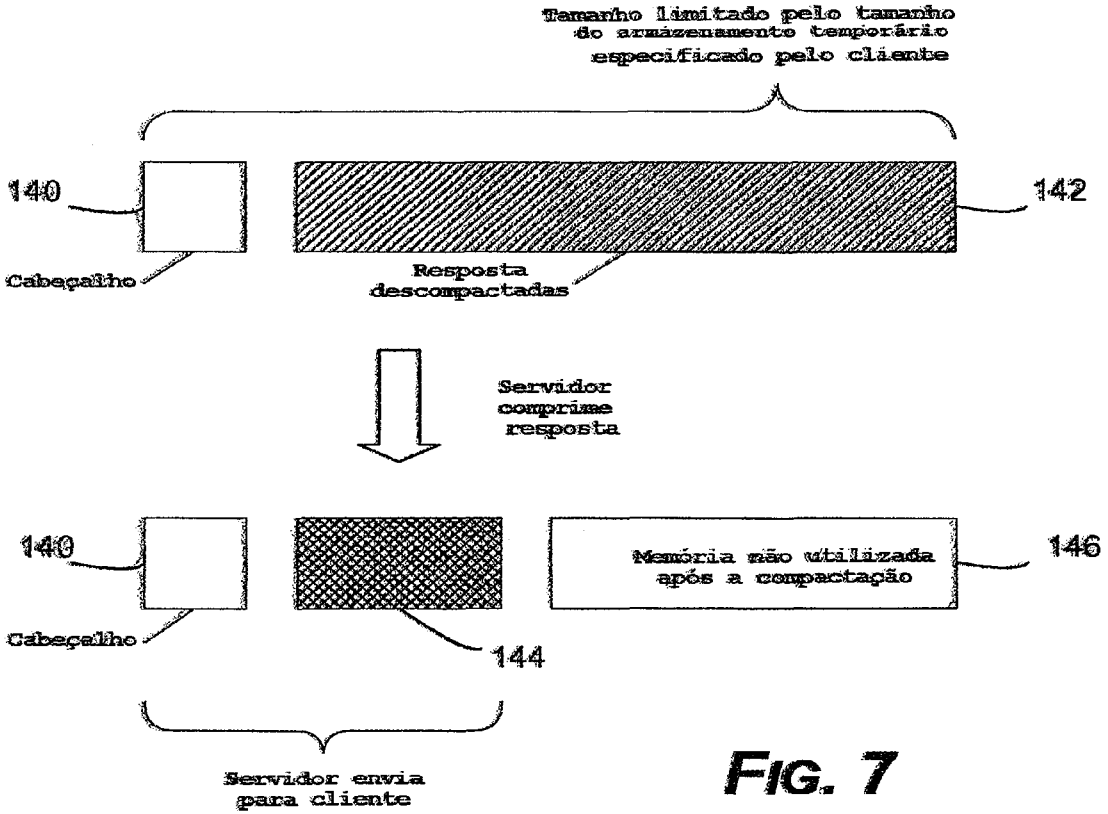

Tamanho limitado pelo tamanho
do armazenamento temporário
especificado pelo cliente
140
142
Cabeçalho
Resposta
descompactadas
Servidor
comprime
resposta
140
Memória não utilizada
após a compactação
146
Cabeçalho
144
Servidor envia
para cliente

FIG. 7

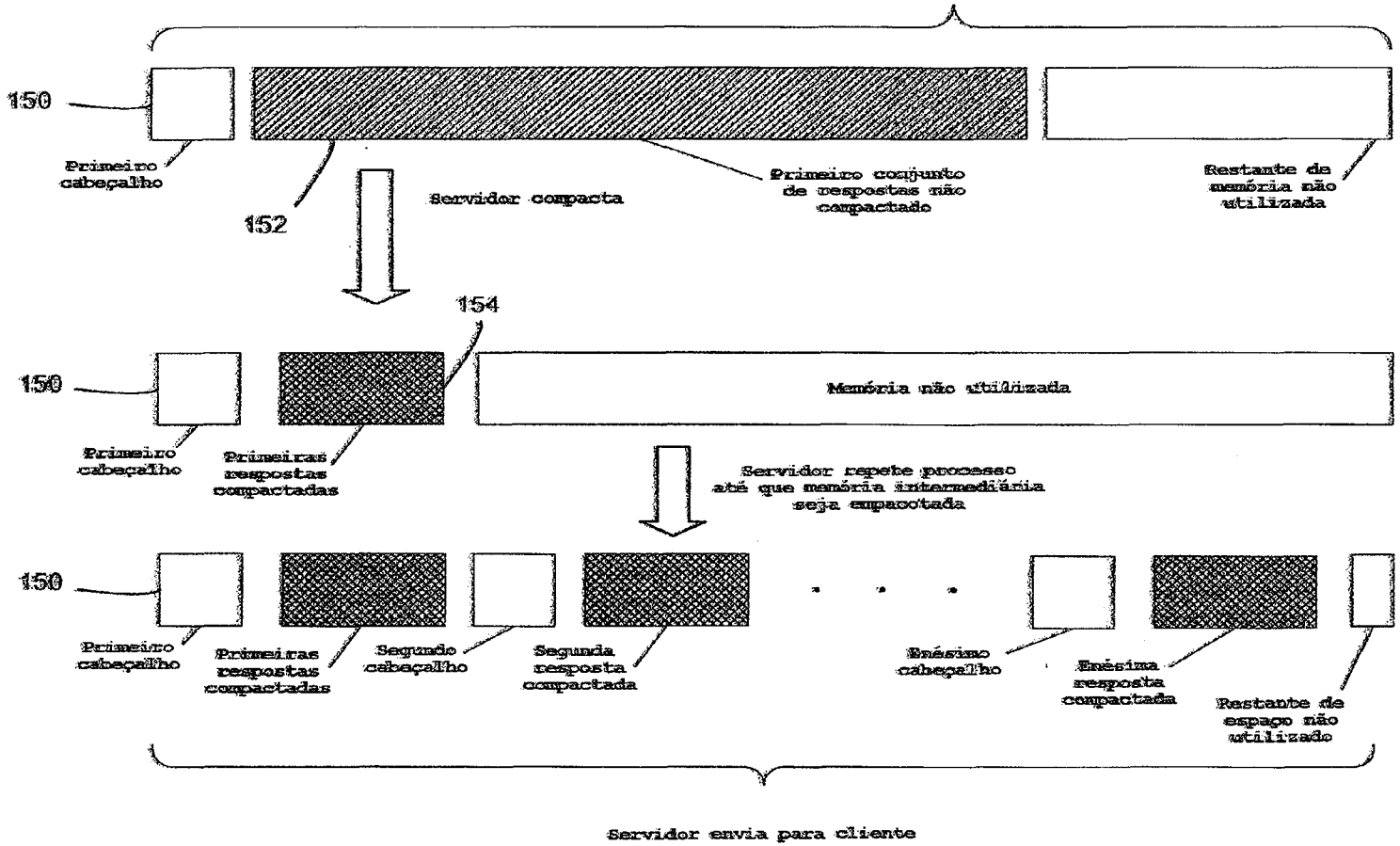

150
Primeiro cabeçalho
152
Servidor compacta
Primeiro conjunto de respostas não compactado
Restante de memória não utilizada
154
150
Primeiro cabeçalho
Primeiras respostas compactadas
Memória não utilizada
Servidor repete processo até que memória intermediária seja empacotada
150
Primeiro cabeçalho
Primeiras respostas compactadas
Segundo cabeçalho
Segunda resposta compactada
Enésimo cabeçalho
Enésima resposta compactada
Restante de espaço não utilizado
Servidor envia para cliente

FIG. 8

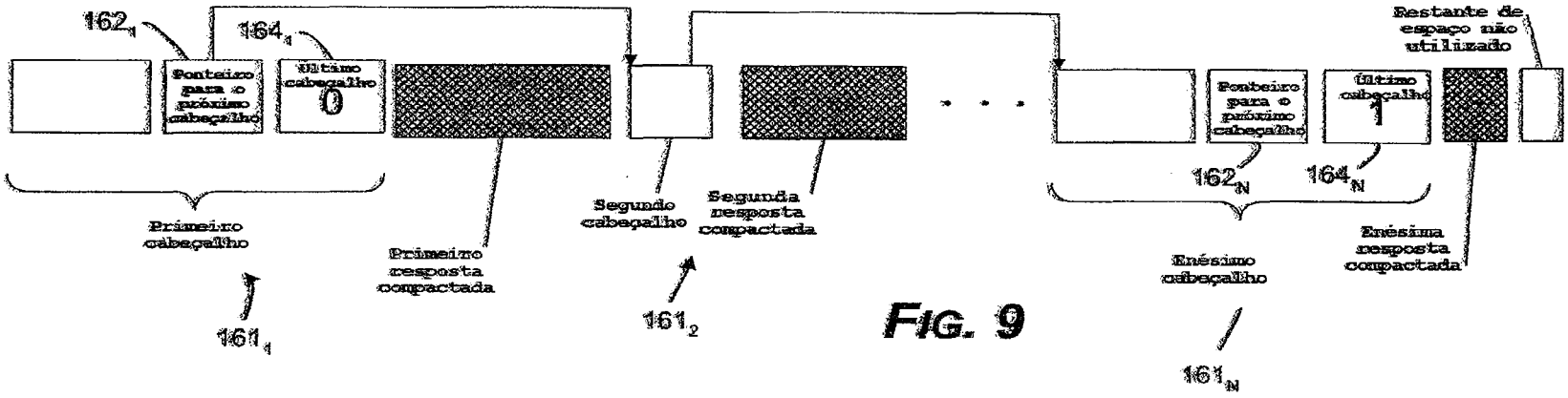

162_1
164_1
Ponteiro para o próximo cabeçalho
Último cabeçalho
0
Restante de espaço não utilizado
Ponteiro para o próximo cabeçalho
Último cabeçalho
1
162_N
164_N
Primeiro cabeçalho
Primeiro resposta compactada
Segundo cabeçalho
Segunda resposta compactada
Enésimo cabeçalho
Enésima resposta compactada
161_1
161_2
161_N

FIG. 9

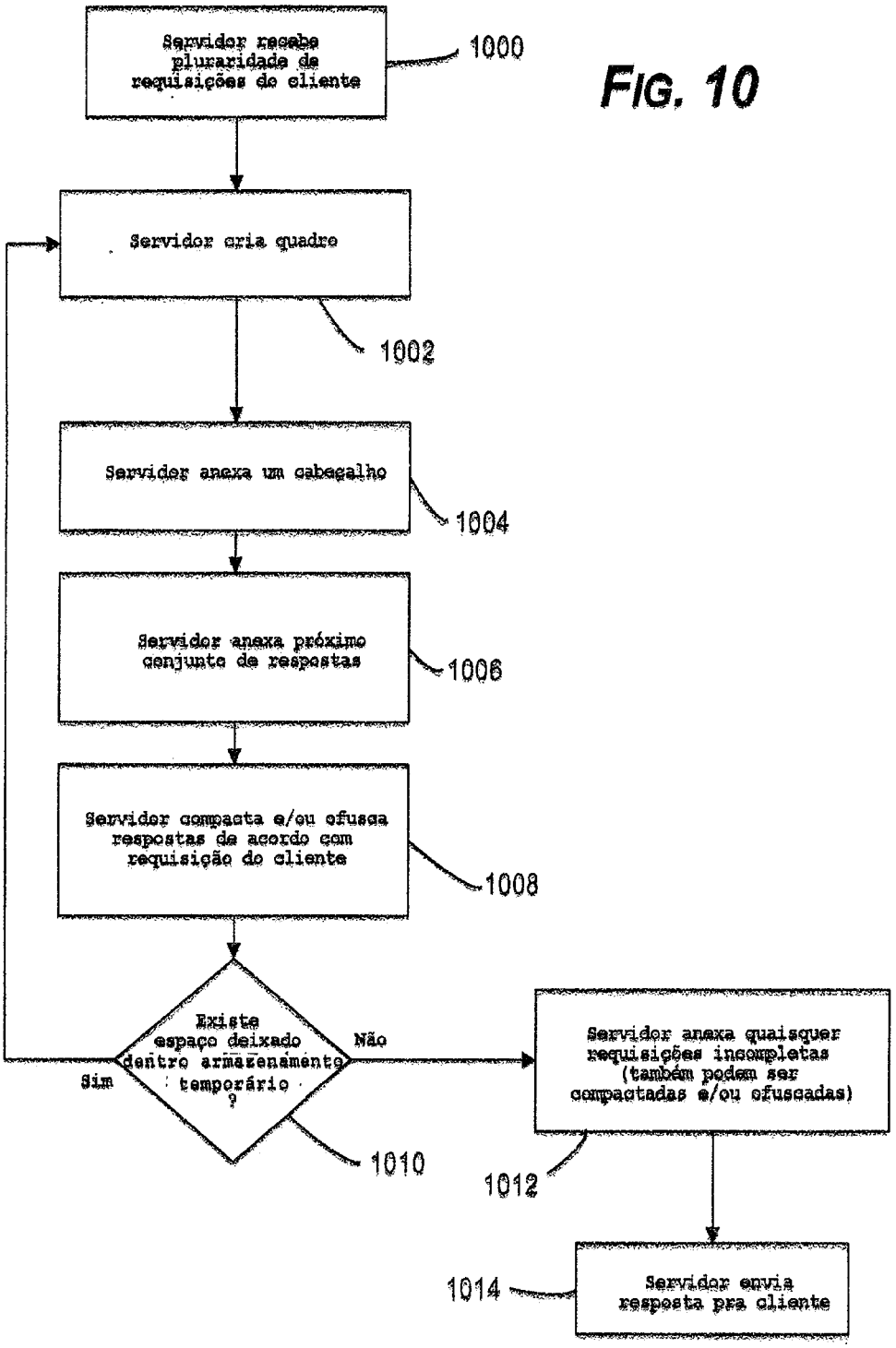

Servidor recebe pluraridade de requisições do cliente
1000
Servidor cria quadro
1002
Servidor anexa um cabeçalho
1004
Servidor anexa próximo conjunto de respostas
1006
Servidor compacta e/ou ofusca respostas de acordo com requisição do cliente
1008
Existe espaço deixado dentro armazenamento temporário ?
Não
Sim
1010
Servidor anexa quaisquer requisições incompletas (também podem ser compactadas e/ou ofuscadas)
1012
Servidor envia resposta pra cliente
1014

FIG. 10

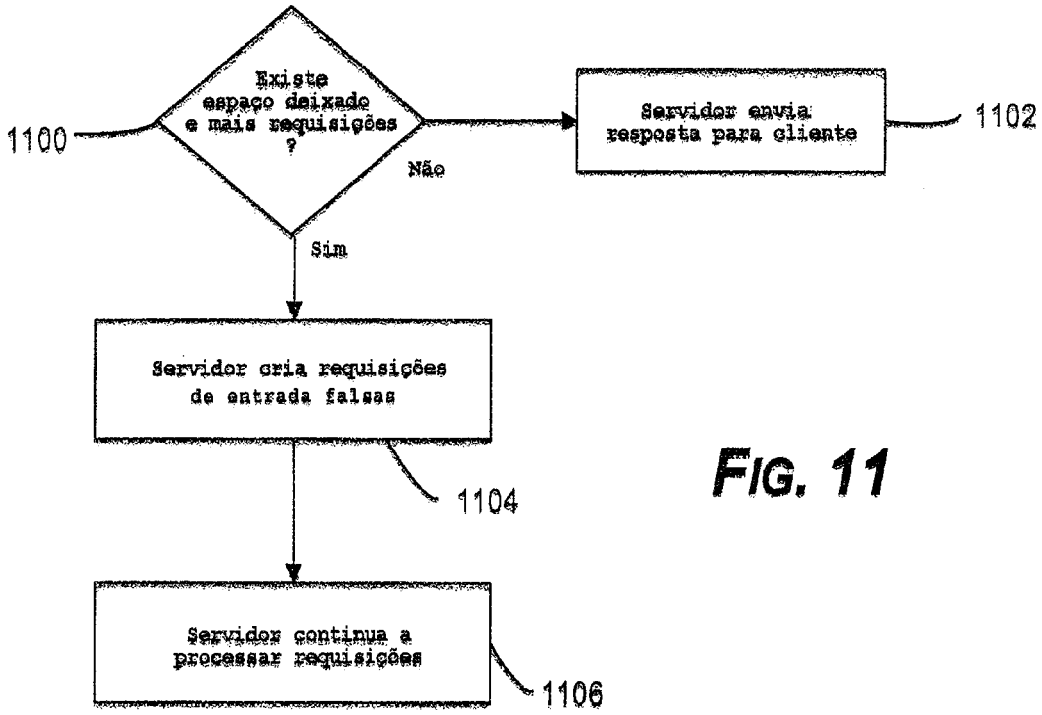

Existe espaço deixado e mais requisições ?
1100
Não
Sim
Servidor envia resposta para cliente
1102
Servidor cria requisições de entrada falsas
1104
Servidor continua a processar requisições
1106

FIG. 11