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

(21) **PI 0622122-0 A2**

* B R P I 0 6 2 2 1 2 2 A 2 *

(22) Data de Depósito: 20/11/2006
(43) Data da Publicação: 27/12/2011
(RPI 2138)

(51) Int.Cl.:
H04L 12/28
H04L 12/66

(54) Título: MÉTODO PARA ELIMINAR CONEXÕES REDUNDANTES

(73) Titular(es): Thomson Licensing

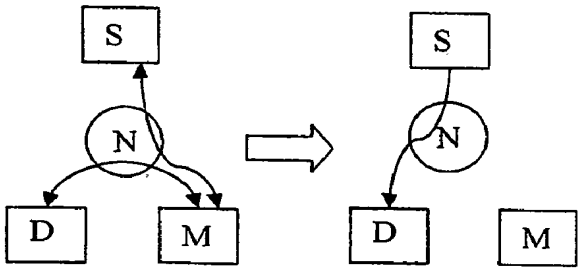
(72) Inventor(es): Ashwin Kashyap, Dekai Li, Jun Gu, Saurabh Mathur

(74) Procurador(es): Nellie Anne Daniel-Shores

(86) Pedido Internacional: PCT US2006044941 de 20/11/2006

(87) Publicação Internacional: WO 2008/063166de 29/05/2008

(57) Resumo: MÉTODO PARA ELIMINAR CONEXÕES REDUNDANTES Um método e sistema para eliminar conexões redundantes, incluindo transmitir uma solicitação para fundir duas conexões, terminar uma primeira porção de uma primeira conexão, terminar uma primeira porção de uma segunda conexão, e fundir uma segunda porção de cada uma das primeira e segunda conexões, são descritos. Um método e sistema para fundir conexões, incluindo receber uma solicitação para fundir duas conexões, terminar uma primeira porção de uma primeira conexão, terminar uma primeira porção de uma segunda conexão, e fundir uma segunda porção de cada uma das primeira e segunda conexões.



PI0622122-0

"MÉTODO PARA ELIMINAR CONEXÕES REDUNDANTES"

Campo da Invenção

A presente invenção refere-se à gestão de redes e especificamente a dispositivos usando conexões TCP/UDP.

Antecedentes da Invenção

O desenvolvimento de tecnologias de gestão de redes facilita a comunicação de vários dispositivos e torna possível para uma aplicação envolver múltiplos dispositivos. Estes dispositivos, usualmente destinados para diferentes propósitos, têm capacidades de computação, capacidade de memória e consumo de potência diferentes. Como resultado, há requerimentos diferentes para estes dispositivos quando eles estiverem envolvidos em uma aplicação, tal como compartilhamento de arquivo e transferência de arquivo.

Fig. 1 ilustra um ambiente de gestão de rede doméstica exemplar em que um ponto de acesso sem fios é conectado à Internet (por meio de um roteador (dispositivo de tradução de rede) com base em Tradução de Endereços de Rede (NAT)). Há também um dispositivo de armazenamento ligado à rede (NAS) (dispositivo de armazenamento de destino/dispositivo de armazenamento), um computador de mesa e vários clientes sem fios, por exemplo, laptop, assistente digital pessoal (PDA). Como aqui usado "/" deveria ser interpretado como denotando nomes alternativos para um componente. Uma aplicação de compartilhamento de arquivo, em um ambiente exemplar, como descrito na Fig. 1, envolve quatro dispositivos: o servidor de arquivo remoto (dados) (servidor de dados fonte/dispositivo fonte), o armazenamento ligado à rede (NAS), um ponto de acesso e pelo menos um cliente móvel (por exemplo, laptop ou PDA). Embora o laptop e PDA sejam equipados com CPUs poderosas e interfaces amigáveis de usuário, eles não são projetados para ficarem ligados todo o tempo. Por outro lado, o ponto de acesso sem fios e o NAS são sistemas embebidos usualmente pequenos com CPUs mais lentas e com menos memória. Eles ficam ligados na maior parte do tempo.

O protocolo de comunicação usado por estes dispositivos é usualmente TCP. Porém, a natureza de ponto-a-ponto do TCP amarra todos os dispositivos participantes juntos e requer que eles fiquem ativos pela duração de uma sessão. Ainda, nem todos os dispositivos são requeridos toda hora. Por exemplo, um PDA pode ser usado para iniciar uma transferência de arquivo. O que realmente é desejado é que o arquivo seja transferido diretamente do servidor de arquivo remoto (dados) (dispositivo fonte) para o disco de NAS (dispositivo de armazenamento) de modo que o PDA possa ser suspenso quando a transferência começar. Pode não ser possível instruir que o NAS o faça diretamente por várias razões. Primeiro, o NAS poderia ser um dispositivo burro, que não pode ser controlado para iniciar conexões. Segundo, o processo de configuração de conexão inicial poderia requerer autenticação complexa e, desse modo, interação humana, por exemplo, preencher nome de usuário

e senha, interpretar um manuscrito (por exemplo, Manuscrito de Java), ou solucionar um quebra-cabeça de gráfico.

Convencionalmente, conexões múltiplas são estabelecidas e mantidas para que a aplicação trabalhe corretamente. Porém, há várias desvantagens com este método. Primeiro, o requerimento que todos os participantes estejam ativos durante uma sessão limita a mobilidade dos dispositivos móveis. Nenhum dispositivo pode ser desconectado ou suspenso porque a função apropriada da aplicação conta com os pontos/dispositivos finais de TCP ativos. Porém, não é usualmente desejável manter aqueles dispositivos ligados todo o tempo. Por exemplo, poderia ser necessário ou desejável levar o notebook para trabalhar ou o PDA poderia ter vida de bateria limitada. Segundo, as conexões redundantes consomem mais recursos. Potência é consumida mantendo aqueles dispositivos ativos; largura de banda extra é requerida para transmitir os dados entre os dispositivos; ciclos de CPU são requeridos para executar cálculo de soma de verificação; memória é requerida para armazenamento temporário dos pacotes. Por fim, as conexões redundantes usualmente não aumentam a confiança e desempenho da aplicação mas ao contrário, as conexões redundantes introduzem mais pontos de falha, mais elos gargalos e latência mais alta. Por exemplo, uma conexão sem fios é frequentemente suscetível à perda de pacote e queda de conexão. Quanto mais ligações sem fios uma aplicação envolve, menos segura a comunicação de dados na verdade é.

Em métodos da técnica anterior, proxies foram desenvolvidas no dispositivo de tradução de endereços de rede. Uma conexão com o exterior pode ser feita e mantida pela proxy. Dispositivos múltiplos podem conectar-se primeiro ao dispositivo de tradução de endereços de rede e requerer a mesma conexão. Assim um dispositivo móvel pode estabelecer uma conexão e instruir outro dispositivo a usar esta conexão. Porém, este método apenas permite compartilhamento de uma conexão entre certos hospedeiros; não diminui o número de conexões. Além disso, alguns dispositivos burros (por exemplo, NAS) podem nem mesmo ser capazes de iniciar uma conexão com o dispositivo de tradução de endereços de rede. Além disso, devido ao número extra de conexões no dispositivo de tradução de endereços de rede, o dispositivo de tradução de rede requer mais recursos, por exemplo cópia de dados e cálculo de soma de verificação, armazenamento temporário de pacote, etc. Divisão de TCP é um método similar da técnica anterior, mas seu foco está em melhorar o desempenho de proxy para uma conexão entre um cliente e um servidor. Isto apenas envolve três dispositivos: cliente, servidor e proxy. Msocks, outro método da técnica anterior, faz uso de divisão de TCP para solucionar o problema de mobilidade do mesmo dispositivo, isto é, quando um hospedeiro move-se, sua conexão é mantida em nível de transporte. Msocks também requer alterar a aplicação no dispositivo móvel para ligar com uma biblioteca especial.

PET020090097031
14/10/09

Sumário da Invenção

A presente invenção é direcionada a um método para eliminar conexões TCP/UDP redundantes quando dispositivos múltiplos estiverem envolvidos em uma aplicação, que permite a migração transparente de uma sessão de comunicação de um dispositivo para outro e facilita a mobilidade dos dispositivos móveis.

O método de acordo com a presente invenção tem muitas vantagens incluindo nenhuma restrição na implementação de TCP do dispositivo móvel. A fim de equiparar duas conexões de modo que elas possam ser fundidas, as conexões deveriam ter características de extensão similares, por exemplo escala de janelas, reconhecimento seletivo, selo de tempo. É possível implementar algumas destas características no dispositivo de tradução de endereços de rede com algum custo indireto extra.

Um método e sistema para eliminar conexões redundantes, incluindo: transmitir uma solicitação para fundir duas conexões, terminar uma primeira porção de uma primeira conexão, terminar uma primeira porção de uma segunda conexão, e fundir uma segunda porção de cada uma das primeira e segunda conexões, são descritos. Um método e sistema para fundir conexões, incluindo receber uma solicitação para fundir duas conexões, terminar uma primeira porção de uma primeira conexão, terminar uma primeira porção de uma segunda conexão, e fundir uma segunda porção de cada uma das primeira e segunda conexões, são também descritos. Deveria ser observado que a presente invenção é descrita em termos das porções de conexões sendo metades de conexão.

Um método e sistema para fundir conexões, incluindo terminar um primeiro mapeamento entre um primeiro dispositivo e um segundo dispositivo, terminar um segundo mapeamento entre o primeiro dispositivo e um terceiro dispositivo e criar um mapeamento entre o segundo dispositivo e o terceiro dispositivo, são também descritos.

Breve Descrição dos Desenhos

A presente invenção é melhor entendida a partir da descrição detalhada a seguir quando lida juntamente com os desenhos em anexo. Os desenhos incluem as figuras a seguir brevemente descritas abaixo:

Fig. 1 ilustra um ambiente de gestão de rede doméstica exemplar.

Fig. 2 é um diagrama esquemático do processo de fusão de conexão de acordo com os princípios da presente invenção.

Descrição Detalhada das Modalidades Preferidas

Fig. 1 ilustra um ambiente de gestão de rede doméstica exemplar em que um ponto de acesso sem fios é conectado à Internet. O ponto de acesso pode ser conectado diretamente à Internet ou pode ser conectado por meio de um roteador (dispositivo de tradução de endereços de rede) com base em Tradução de Endereços de Rede (NAT). Há também um dispositivo de armazenamento ligado à rede (NAS) (dispositivo de armazenamento de destino), um computador de mesa e vários clientes sem fios, por exemplo laptop, PDA. Uma aplicação de compartilhamento de arquivo, em um ambiente exemplar como descrito na Fig. 1, envolve quatro dispositivos: o servidor de arquivo remoto (dados), o armazenamento ligado ao dispositivo de rede (NAS), um ponto de acesso e pelo menos um cliente móvel (por exemplo, laptop ou assistente digital pessoal (PDA)). Embora o laptop e PDA sejam requeridos com CPUs poderosas e interfaces amigáveis ao usuário, eles não são projetados para ser ligados toda hora. Por outro lado, o ponto de acesso sem fios e o NAS são sistemas embutidos usualmente pequenos com CPUs mais lentas e menos memória. Eles são ligados na maior parte do tempo.

A presente invenção é direcionada às aplicações do tipo de compartilhamento de dados/arquivos ou de transferência de dados/arquivos tendo conexões redundantes que tipicamente ocorrem com protocolos de comunicação TCP e UDP. Para a maioria das aplicações de compartilhamento ou transferência de dados/arquivos, dados não são urgentemente necessários como em aplicações interativas, embora a presente invenção possa também ser usada com aplicações de fluxo de dados e de VoIP. É observado que embora a presente invenção seja descrita em termos de transferência de arquivos/dados, a presente invenção trabalhará bem para aplicações de transferência de arquivos/dados, em cujo caso os papéis dos dispositivos serão invertidos. O dispositivo fonte se torna o dispositivo destino e vice-versa.

Em uma modalidade, a presente invenção trata de aplicações com base em TCP. Referindo à Fig. 2 onde um dispositivo móvel inicia uma conexão com um servidor de dados fonte (dispositivo fonte) S usando uma primeira conexão e salva o conteúdo em um servidor de armazenamento destino/dispositivo de armazenamento D usando uma segunda conexão. Ambas as conexões passam por meio do dispositivo de tradução de endereços de rede. Ou seja, a conexão entre o dispositivo móvel e o dispositivo fonte passa por meio do dispositivo de tradução de endereços de rede como também a conexão entre o dispositivo móvel e o dispositivo de armazenamento. Após estabelecimento de conexão, o dispositivo móvel requer para o dispositivo de tradução de endereços de rede fundir estes duas conexões de modo que o dispositivo fonte S transfere dados diretamente ao dispositivo de armazenamento D. Deveria ser observado que as conexões na Fig. 2 têm setas com pontas duplas que

indicam que a comunicação é em ambas as direções. Os símbolos a seguir são usados na descrição do método da presente invenção.

S: dispositivo fonte / servidor de dados fonte

M: dispositivo móvel

D: servidor de armazenamento destino / dispositivo de armazenamento

N: hospedeiro de tradução de endereços de rede/dispositivo de tradução de endereços de rede

Uma tupla é definida para registrar informação de estado de protocolo para cada direção de uma conexão. A tupla consiste em dois campos:

T(snd_max, ack_max)

A tupla registra o número de sequência máximo e número de reconhecimento máximo visto pelo dispositivo de tradução de endereços de rede para uma direção de uma conexão. Cada tupla pode ser pensada como um espelho atrasado dos estados de protocolo do dispositivo correspondente. Porém, diferente de um dispositivo terminal que tem que manter a exatidão dos estados, o dispositivo de tradução de endereços de rede apenas mantém um registro. Por exemplo, o dispositivo terminal deve armazenar temporariamente pacotes de fora-de-sequência, enquanto o dispositivo de tradução de endereços de rede não armazena temporariamente nenhum pacote durante operação normal. Portanto, pode ser dito que o estado é atrasado porque ainda há pacotes dentro da rede entre os dispositivos terminais e o dispositivo de tradução de endereços de rede. A vantagem deste método é que a complexidade do protocolo é mantida no dispositivo terminal sem requerer uma proxy no dispositivo de tradução de endereços de rede.

Há quatro tuplas usadas no dispositivo de tradução de endereços de rede na configuração de acordo com os princípios da presente invenção.

T_{sm}: Tupla para meia-conexão de hospedeiro fonte para dispositivo móvel

T_{ms}: Tupla para meia-conexão de hospedeiro móvel para dispositivo fonte

T_{md}: Tupla para meia-conexão de hospedeiro móvel para dispositivo destino

T_{dm}: Tupla para meia-conexão de hospedeiro destino para dispositivo móvel

Estas quatro tuplas correspondem às duas conexões, (T_{ms}, T_{sm}) para conexão entre o dispositivo fonte e o dispositivo móvel por meio do dispositivo de tradução de endereços de rede e (T_{md}, T_{dm}) entre o dispositivo móvel e o dispositivo destino/armazenamento por meio do dispositivo de tradução de endereços de rede. Desse modo, durante a configuração, o dispositivo móvel inicia uma primeira conexão no dispositivo fonte (servidor de dados fonte ou servidor de dados). Depois o dispositivo móvel inicia uma segunda conexão no dispositivo de armazenamento. Estas duas etapas podem ser trocadas. O dispositivo móvel depois obtém o arquivo/dados do dispositivo fonte. Por fim, o dispositivo móvel remete o arquivo/dados ao dispositivo de armazenamento.

Quando o dispositivo móvel decide fundir estas duas conexões, o dispositivo móvel primeiro envia uma mensagem ao dispositivo de tradução de endereços de rede indicando as duas conexões a serem fundidas e o modo de fundir, neste caso, eliminando as conexões que envolvem o dispositivo móvel. O dispositivo de tradução de endereços de rede aguarda por dados não respondidos serem reconhecidos e depois termina o lado do dispositivo móvel da conexão entre o dispositivo móvel e o dispositivo fonte. Após processar os dados recebidos armazenados temporariamente, o dispositivo móvel envia outra mensagem ao dispositivo de tradução de endereços de rede, indicando o término de comunicação com o dispositivo de armazenamento. Ao receber a segunda mensagem, o dispositivo de tradução de endereços de rede pode iniciar a fusão das conexões. As conexões são fundidas em estágios como segue:

Estágio 1: Operação normal

No dispositivo de tradução de endereços de rede, a informação de protocolo necessita ser rastreada para toda conexão que poderia ser fundida com outra conexão. A partir do número de sequência do pacote com conjunto de flag SYN, o número de sequência mais alto e o número de reconhecimento que o dispositivo de tradução de endereços de rede viu é registrado, embora o outro lado possa não ter recebido todos os pacotes precedentes devido à perda ou reordem dos pacotes. O dispositivo terminal manipulará a retransmissão dos pacotes perdidos. Por exemplo, um pacote P_{sm} enviado do dispositivo fonte ao dispositivo móvel com número de sequência P_{sm}-seq e número de reconhecimento P_{sm}-ack atualiza a tupla correspondente como segue:

If (T_{sm}.snd_max < P_{sm}-seq)

T_{sm}.snd_max = P_{sm}.seq;

If (T_{sm}.ack_max < P_{sm}.ack)

T_{sm}.ack_max = P_{sm}.ack;

Estágio 2: "Terminando" a primeira conexão

Quando o dispositivo móvel requerer fusão das duas conexões, ele envia uma mensagem para notificar o dispositivo de tradução de endereços de rede. O dispositivo de tradução de endereços de rede depois congela as duas tuplas para a primeira conexão entre o dispositivo móvel e o dispositivo fonte. Deste ponto de tempo adiante, apenas pacotes com número de sequência e número de reconhecimento inferiores podem ser enviados entre eles. Estes pacotes ou estão atrasados dentro da rede ou retransmitidos por causa de erros. Se o número de sequência de certa carga útil for além de "snd_max" da tupla correspondente, a carga útil é truncada. Porque tanto o hospedeiro/dispositivo móvel como o hospedeiro/dispositivo fonte não podem ver números de sequência maiores que o número de sequência máximo visto pelo dispositivo de tradução de endereços de rede, pacotes com número de reconhecimento maior que "ack_max" da tupla correspondente são deixados.

Porque o tempo que leva para executar a operação de fusão deveria ser menor que o valor de tempo de aguarda de TCP, os dispositivos terminais deveriam ser capazes de detectar os pacotes deixados como perdidos e executar retransmissão. Como uma melhoria de implementação, o dispositivo de tradução de endereços de rede poderia selecionar armazenar temporariamente alguns pacotes com carga útil além de "snd_max" dependendo da memória disponível do dispositivo de tradução de endereços de rede.

Após pacotes serem recebidos corretamente, o dispositivo de tradução de endereços de rede inicia um fechamento ativo e envia um FIN ao dispositivo móvel. Do ponto de vista do dispositivo móvel, a conexão é terminada. O dispositivo de tradução de endereços de rede depois aguarda pelo FIN, ACK do dispositivo móvel e depois o dispositivo de tradução de endereços de rede envia o ACK final. Ou seja, a metade da conexão entre o dispositivo móvel e o dispositivo fonte, onde o dispositivo móvel recebe os dados do dispositivo fonte, é terminada.

Estágio 3: "Terminando" a segunda conexão

O dispositivo móvel continua processando os dados armazenados temporariamente do dispositivo fonte e depois envia uma mensagem ao dispositivo de tradução de endereços de rede indicando um término de conexão com o dispositivo de armazenamento. Esta mensagem é definida pela aplicação e enviada após todos os dados enviados ao dispositivo de armazenamento terem sido reconhecidos. Ou seja, a metade da conexão entre o dispositivo móvel e o dispositivo de armazenamento, onde o dispositivo de armazenamento recebe dados do dispositivo móvel, é terminada.

Em outra modalidade o dispositivo móvel envia um pacote com FIN para o dispositivo de armazenamento. Nesta modalidade, o dispositivo de tradução de endereços de rede intercepta a mensagem de FIN sem remetê-la ao destino. Isto é essencialmente um fim passivo e o dispositivo de tradução de endereços de rede responde ao dispositivo móvel com FIN e ACK e aguarda pelo ACK final do dispositivo móvel. Se a aplicação envia uma mensagem sem usar uma mensagem de FIN, o dispositivo de tradução de endereços de rede termina a meia-conexão com o dispositivo móvel como no estágio anterior.

Neste estágio, as tuplas para a segunda conexão são atualizadas até a mensagem de término ou um pacote com uma mensagem de FIN ser recebido. Depois as tuplas para a segunda conexão são congeladas. Deste ponto de tempo, apenas pacotes com um número de sequência menor que snd_max e um número de reconhecimento menor que ack_max são aceitos como no estágio anterior.

Após terminar a metade da conexão do dispositivo móvel entre o dispositivo móvel e o dispositivo de armazenamento, o dispositivo de tradução de endereços de rede cria um mapeamento novo entre o dispositivo de armazenamento e o dispositivo fonte e prepara para enviar os pacotes entre eles. Se uma mensagem de FIN é usada para terminar a cone-

xão, snd_max é especificado como o número de sequência de FIN menor.

Estágio 4: Conclusão da fundição

O número de sequência e desse modo o número de reconhecimento destas duas conexões são independentes. Pacotes não podem meramente ser mandados diretamente entre o dispositivo fonte e o dispositivo de armazenamento. Os pacotes necessitam ser modificados de forma que eles pareçam ser do dispositivo móvel, embora não estejam agora envolvidos nestas duas conexões. Por exemplo, para pacotes que vêm do dispositivo fonte, o dispositivo fonte enviando espaço de sequência é mapeado para o espaço de sequência de envio usado pelo dispositivo móvel para enviar pacotes ao dispositivo de armazenamento. Os reconhecimentos são também modificados de forma que eles enquadram-se no espaço de envio usado pelo dispositivo de armazenamento. O número de sequência é modificado como segue:

P_{sm}.seq = P_{sm}-seq - T_{sm}.snd_max + T_{md}.snd_max;

O ACK é modificado como segue:

P_{sm}.ack = P_{sm}.ack - T_{ms}.snd_max + T_{dm}.snd_max;

As operações detalhadas para pacotes do dispositivo fonte para o dispositivo móvel são como segue:

P_{sm}-dst_ip = D.ip (altera o ip de destino para o destino (dispositivo de armazenamento))

P_{sm}.dst_port = D.port

P_{sm}.src_ip = M.ip

P_{sm}.src_port = M.port (a porta usada para comunicar-se com D)

P_{sm}-seq = P_{sm}.seq - T_{sm}.snd_max + T_{md}.snd_max

P_{sm}.ack = P_{sm}-ack - T_{ms}.snd_max + T_{dm}.snd_max

As terceira e quarta instruções junto enganam o destino (dispositivo de armazenamento) para pensar que os dados/arquivo são do dispositivo móvel mas os dados/arquivo estão vindo de fato do dispositivo fonte.

As operações detalhadas para pacotes do dispositivo de armazenamento para o dispositivo móvel são como segue:

P_{dm}.dst_ip = S.ip

P_{dm}-dst_port = S.port

P_{dm}.src_ip = N.ip

P_{dm}.dst_port = N.port

P_{dm}.seq = P_{dm}-seq - T_{dm}.snd_max + T_{ms}.snd_max

P_{dm}-ack = P_{dm}-ack - T_{md}.snd_max + T_{ms}.snd_max

O que as operações acima realizam é a fusão da outra (segunda) metade de cada uma das primeira e segunda conexões. Ou seja, o dispositivo fonte envia os da-

dos/arquivo/pacotes ao dispositivo de armazenamento por meio do dispositivo de tradução de endereços de rede que modifica os cabeçalhos de pacote de forma que parece para o dispositivo de armazenamento que os dados/pacotes estão vindo do dispositivo móvel que já está envolvido. Isto eficazmente funde ou termina a comunicação do dispositivo móvel com o dispositivo fonte e do dispositivo de armazenamento com o dispositivo móvel.

É observado que N.port é a porta visível para o servidor fonte de dados remoto (dispositivo fonte remoto). Dependendo da implementação do dispositivo de tradução de endereços de rede, esta poderia ser diferente da porta usada pelo dispositivo móvel para conectar ao dispositivo fonte. M.port é a porta visível para o dispositivo de armazenamento. No exemplo acima, o dispositivo de armazenamento está dentro da mesma rede que o dispositivo móvel, assim M.port é a mesma porta usada pelo dispositivo móvel. Após as modificações acima, a soma de verificação de cabeçalho de TCP e cabeçalho de IP é calculada ou incrementalmente recalculada.

Em outra modalidade, aplicações usam UDP como o protocolo de transporte. O dispositivo móvel conecta primeiro ao dispositivo fonte e o dispositivo de tradução de endereços de rede mantém um mapeamento entre a porta UDP usada pelo dispositivo móvel e a porta usada pelo dispositivo fonte. Quando o dispositivo móvel decidir migrar uma sessão para o dispositivo de armazenamento, o dispositivo móvel ativa primeiro uma sessão UDP para o dispositivo de armazenamento (o dispositivo móvel ou o dispositivo de armazenamento pode iniciar a sessão dependendo da implementação). Depois o dispositivo móvel instrói o dispositivo de tradução de endereços de rede para alterar o mapeamento entre o dispositivo móvel M e o dispositivo fonte S para um mapeamento entre o dispositivo de armazenamento D e o dispositivo fonte S.

Para pacotes que são transmitidos do dispositivo fonte para o dispositivo móvel, o dispositivo de tradução de endereços de rede executa as operações a seguir:

P_{sm}.src_ip = M.ip

P_{sm}.src_port = M.port (a porta usada para comunicar com D)

P_{sm}dst_port = D.port

P_{sm}.dst_ip = D.ip

Para pacotes que são transmitidos do dispositivo de armazenamento para o dispositivo móvel, o dispositivo de tradução de endereços de rede executa as operações a seguir:

P_{dm}.dst_ip = S.ip

P_{dm}-dst_port = S.port

P_{dm}.dst_port = N.port

Além da aplicação de compartilhamento de arquivo ilustrada no método de acordo com a presente invenção, o método da presente invenção tem outras aplicações, por exemplo, como segue:

1. Transparentemente troca uma sessão de fluxo de um dispositivo para o outro. Usando o método da presente invenção, um cliente de fluxo que permite os usuários iniciar em assistir um vídeo de fluxo a partir de um computador pessoal (PC) ou PDA pode depois migrá-lo para uma TV (ou TV com caixa de sinal de frequência) que também usa TCP/UDP. Se a aplicação de fluxo apenas usar o TCP, (por exemplo RTP em RTSP), o dispositivo móvel transfere primeiro o estado de aplicação para o dispositivo de armazenamento e depois migra a conexão de TCP para o dispositivo de armazenamento. Se a aplicação de fluxo usar o TCP (por exemplo RTSP) e UDP (RTP), o dispositivo móvel transfere primeiro o estado da aplicação (estado de RTSP) e estado de RTP para o dispositivo de armazenamento e depois migra as conexões TCP e UDP para o dispositivo de armazenamento.

2. Transparentemente troca de "visualizando" uma sessão de fluxo para "armazenando" o fluxo para o armazenamento local. Esta aplicação é similar à descrita acima exceto que o dispositivo de armazenamento armazena o fluxo em armazenamento local em vez de transmitir/exibir o mesmo em um dispositivo, por exemplo TV.

3. Transparentemente troca uma chamada VoIP de um dispositivo para o outro, ou do telefone de um usuário para o telefone de outro usuário. Esta aplicação é similar às duas aplicações acima e é também implementada transferindo primeiro o estado de aplicação para o dispositivo de armazenamento e depois migrando a conexão TCP/UDP.

Para aplicações que não usam estados adicionais ou apenas usam alguns estados no começo de uma sessão, as aplicações que correm no dispositivo móvel e no dispositivo de armazenamento podem permanecer inalteradas. Porém, para aplicações que têm que manter alguns estados durante a sessão, as aplicações devem ser modificadas para transferir aqueles estados e usar o método de acordo com a presente invenção como descrito aqui para migrar a conexão subjacente.

Para iniciar a fusão de conexões, o dispositivo de tradução de endereços de rede pode fornecer uma interface de rede com base em interface de portão comum (CGI). O cliente pode acessar os manuscritos de CGI para adquirir uma lista de conexões correntemente rastreadas e indicar as conexões que o dispositivo de tradução de endereços de rede deveria fundir. Outra modalidade é desenvolver um processo daemon no dispositivo de tradução de endereços de rede e definir uma interface de comunicação para o dispositivo móvel enviar as instruções de fusão.

É para ser entendido que a presente invenção pode ser implementada em várias formas de hardware, software, firmware, processadores de propósito especial, ou uma combinação dos mesmos. Preferivelmente, a presente invenção é implementada como uma combinação de hardware e software. Além disso, o software é preferivelmente implementado como um programa de aplicação tangivelmente incorporado em um dispositivo de armazenamento de programa. O programa de aplicação pode ser transferido, e executado, por

uma máquina compreendendo qualquer arquitetura adequada. Preferivelmente, a máquina é implementada em uma plataforma de computador tendo hardware tal como uma ou mais unidades de processamento central (CPU), uma memória de acesso aleatório (RAM), e interface(s) de entrada/saída (I/O). A plataforma de computador também inclui um sistema operacional e código de microinstrução. Os vários processos e funções descritas aqui ou podem fazer parte do código de microinstrução ou parte do programa de aplicação (ou uma combinação dos mesmos) que são executados por meio do sistema operacional. Além disso, vários outros dispositivos periféricos podem ser conectados à plataforma de computador tais como um dispositivo de armazenamento de dados adicional e um dispositivo de impressão.

É também para ser entendido que, porque alguns dos componentes constituintes do sistema e etapas metodológicas descritos nas figuras em anexo são preferivelmente implementados em software, as conexões atuais entre os componentes do sistema (ou as etapas processuais) podem diferir dependendo da maneira em que a presente invenção é programada. Dado os ensinamentos aqui, alguém de habilidade usual na técnica relacionada poderá contemplar estas e implementações ou configurações similares da presente invenção.

REIVINDICAÇÕES

1. Método para eliminar conexões redundantes, **CARACTERIZADO** pelo fato de que o dito método compreende:

transmitir uma solicitação para fundir duas conexões;

terminar uma primeira porção de uma primeira conexão;

terminar uma primeira porção de uma segunda conexão; e

fundir uma segunda porção de cada uma das ditas primeira e segunda conexões.

2. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que a dita terminação da dita primeira porção da dita primeira conexão adicionalmente compreende:

terminar as comunicações entre um primeiro dispositivo e um segundo dispositivo, em que a dita primeira conexão realizando comunicações entre o dito primeiro dispositivo e o dito segundo dispositivo, a dita segunda conexão realizando comunicações entre o dito primeiro dispositivo e um terceiro dispositivo, e também em que cada uma das primeira e segunda conexões é realizada através de um quarto dispositivo;

processar dados armazenados temporariamente pelo dito primeiro dispositivo recebidos do dito segundo dispositivo; e

enviar uma mensagem indicando um término de comunicação com o dito segundo dispositivo.

3. Método, de acordo com a reivindicação 2, **CARACTERIZADO** pelo fato de que a dita terminação da dita primeira porção da dita segunda conexão adicionalmente compreende:

enviar dados armazenados temporariamente pelo dito terceiro dispositivo; e

enviar uma mensagem indicando um término de comunicação com o dito terceiro dispositivo.

4. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que adicionalmente compreende estabelecer a dita primeira conexão entre o dito primeiro dispositivo e o dito segundo dispositivo e estabelecer a dita segunda conexão entre o dito primeiro dispositivo e o dito terceiro dispositivo.

5. Método, de acordo com a reivindicação 2, **CARACTERIZADO** pelo fato de que a dita terminação da dita primeira porção da dita segunda conexão adicionalmente compreende:

enviar um término de mensagem de conexão a um terceiro dispositivo;

receber um reconhecimento para o dito término de mensagem de conexão; e

enviar um reconhecimento final.

6. Método para fundir conexões, **CARACTERIZADO** pelo fato de que o dito método compreende:

terminar uma primeira porção de uma primeira conexão;

terminar uma primeira porção de uma segunda conexão; e

fundir uma segunda porção de cada uma das ditas primeira e segunda conexões.

7. Método, de acordo com a reivindicação 6, **CARACTERIZADO** pelo fato de que o término da dita primeira porção da dita primeira conexão adicionalmente compreende:

aguardar por um reconhecimento de recebimento de quaisquer dados previamente transmitidos mas não reconhecidos;

congelar a informação com relação a um estado de protocolo entre os dispositivos que se comunicam por meio da dita primeira conexão, em que a dita primeira conexão realizando comunicações entre o dito primeiro dispositivo e o dito segundo dispositivo, a dita segunda conexão realizando comunicações entre o dito primeiro dispositivo e um terceiro dispositivo, e em que cada uma das primeira e segunda conexões é realizada através de um quarto dispositivo; e

receber uma mensagem indicando um término de comunicação entre o dito primeiro dispositivo e o dito segundo dispositivo.

8. Método, de acordo com a reivindicação 7, **CARACTERIZADO** pelo fato de que o término da dita primeira porção da dita segunda conexão adicionalmente compreende:

atualizar informação com relação a um estado de protocolo entre os dispositivos que se comunicam por meio da dita segunda conexão;

receber uma mensagem indicando um término de comunicação entre o dito primeiro dispositivo e o dito terceiro dispositivo; e

congelar a informação com relação a um estado de protocolo entre os ditos dispositivos que se comunicam por meio da dita segunda conexão.

9. Método, de acordo com a reivindicação 8, **CARACTERIZADO** pelo fato de que adicionalmente compreende criar um mapeamento novo entre o dito segundo dispositivo e o dito terceiro dispositivo.

10. Método, de acordo com a reivindicação 9, **CARACTERIZADO** pelo fato de que o dito mapeamento novo modifica números de sequência e números de reconhecimento nos pacotes de dados de modo que os ditos pacotes pareçam com o dito terceiro dispositivo como se os ditos pacotes estivessem sendo transmitidos pelo dito primeiro dispositivo.

11. Método, de acordo com a reivindicação 6, **CARACTERIZADO** pelo fato de que os números de sequência máximos são registrados para cada conexão.

12. Método, de acordo com a reivindicação 6, **CARACTERIZADO** pelo fato de que os números de reconhecimento máximos são registrados para cada conexão.

13. Sistema para eliminar conexões redundantes, **CARACTERIZADO** pelo fato de que compreende:

meios para transmitir uma solicitação para fundir duas conexões;

meios para terminar uma primeira porção de uma primeira conexão;

meios para terminar uma primeira porção de uma segunda conexão; e

meios para fundir uma segunda porção de cada uma das ditas primeira e segunda conexões.

14. Sistema, de acordo com a reivindicação 13, **CARACTERIZADO** pelo fato de que os ditos meios para terminar a dita primeira porção da dita primeira conexão adicionalmente compreendem:

meios para terminar uma conexão entre o dito primeiro dispositivo e o dito segundo dispositivo, a dita primeira conexão realizando comunicações entre um primeiro dispositivo e um segundo dispositivo, a dita segunda conexão realizando comunicações entre o dito primeiro dispositivo e um terceiro dispositivo, e em que cada uma das ditas primeira e segunda conexões é realizada através de um quarto dispositivo;

meios para processar dados armazenados temporariamente pelo dito primeiro dispositivo recebidos do dito segundo dispositivo; e

meios para enviar uma mensagem indicando um término de comunicação com o dito segundo dispositivo.

15. Sistema, de acordo com a reivindicação 14, **CARACTERIZADO** pelo fato de que os ditos meios para terminar a dita primeira porção da dita segunda conexão adicionalmente compreendem:

meios para enviar dados armazenados temporariamente ao dito terceiro dispositivo; e

meios para enviar uma mensagem indicando um término de comunicação com o dito terceiro dispositivo.

16. Sistema, de acordo com a reivindicação 13, **CARACTERIZADO** pelo fato de que adicionalmente compreende meios para estabelecer a dita primeira conexão entre o dito primeiro dispositivo e o dito segundo dispositivo e meios para estabelecer a dita segunda conexão entre o dito primeiro dispositivo e um terceiro dispositivo.

17. Sistema, de acordo com a reivindicação 14, **CARACTERIZADO** pelo fato de que os ditos meios para terminar a dita primeira porção da dita segunda conexão adicionalmente compreendem:

meios para enviar um término de mensagem de conexão ao dito terceiro dispositivo;

meios para receber um reconhecimento para o dito término de mensagem de conexão; e

meios para enviar um reconhecimento final.

18. Sistema para fundir conexões, **CARACTERIZADO** pelo fato de que compreende:

meios para receber uma solicitação para fundir duas conexões;

meios para terminar uma primeira porção de uma primeira conexão;

meios para terminar uma primeira porção de uma segunda conexão; e

meios para fundir uma segunda porção de cada uma das ditas primeira e segunda conexões.

19. Sistema, de acordo com a reivindicação 18, **CARACTERIZADO** pelo fato de que os meios para terminar a dita primeira porção da dita primeira conexão adicionalmente compreende:

meios para aguardar por um reconhecimento de recebimento de quaisquer dados previamente transmitidos mas não reconhecidos;

meios para congelar informação com relação a um estado de protocolo entre os dispositivos que se comunicam por meio da dita primeira conexão, a dita primeira conexão realizando comunicações entre um primeiro dispositivo e um segundo dispositivo, a dita segunda conexão realizando comunicações entre o dito primeiro dispositivo e um terceiro dispositivo, e em que cada uma das ditas primeira e segunda conexões é realizada através de um quarto dispositivo: e

meios para receber uma mensagem indicando um término de comunicação entre o dito primeiro dispositivo e o dito segundo dispositivo.

20. Sistema, de acordo com a reivindicação 19, **CARACTERIZADO** pelo fato de que os meios para terminar a dita primeira porção da dita segunda conexão adicionalmente compreendem:

meios para atualizar informação com relação a um estado de protocolo entre os dispositivos que se comunicam por meio da dita segunda conexão;

meios para receber uma mensagem indicando um término de comunicação entre o dito primeiro dispositivo e o dito terceiro dispositivo; e

meios para congelar informação com relação a um estado de protocolo entre os ditos dispositivos que se comunicam por meio da dita segunda conexão.

21. Sistema, de acordo com a reivindicação 20, **CARACTERIZADO** pelo fato de que adicionalmente compreende criar um mapeamento novo entre o dito segundo dispositivo e o dito terceiro dispositivo.

22. Sistema, de acordo com a reivindicação 21, **CARACTERIZADO** pelo fato de que o dito mapeamento novo modifica números de sequência e números de reconhecimento nos pacotes de dados de modo que os ditos pacotes pareçam com o dito terceiro dispositivo como se os ditos pacotes estivessem sendo transmitidos pelo dito primeiro dispositivo.

23. Sistema, de acordo com a reivindicação 18, **CARACTERIZADO** pelo fato de que números de sequência máximos são registrados para cada conexão.

24. Sistema, de acordo com a reivindicação 18, **CARACTERIZADO** pelo fato de

que números de reconhecimento máximos são registrados para cada conexão.

25. Sistema, de acordo com a reivindicação 18, **CARACTERIZADO** pelo fato de que adicionalmente compreende meios para calcular uma soma de verificação.

26. Método para fundir conexões, **CARACTERIZADO** pelo fato de que o dito método compreende:

terminar um primeiro mapeamento entre um primeiro dispositivo e um segundo dispositivo;

terminar um segundo mapeamento entre o dito primeiro dispositivo e um terceiro dispositivo; e

criar um mapeamento entre o dito segundo dispositivo e o dito terceiro dispositivo.

27. Sistema para fundir conexões, **CARACTERIZADO** pelo fato de que compreende:

meios para terminar um primeiro mapeamento entre um primeiro dispositivo e um segundo dispositivo;

meios para terminar um segundo mapeamento entre o dito primeiro dispositivo e um terceiro dispositivo; e

meios para criar um mapeamento entre o dito segundo dispositivo e o dito terceiro dispositivo.

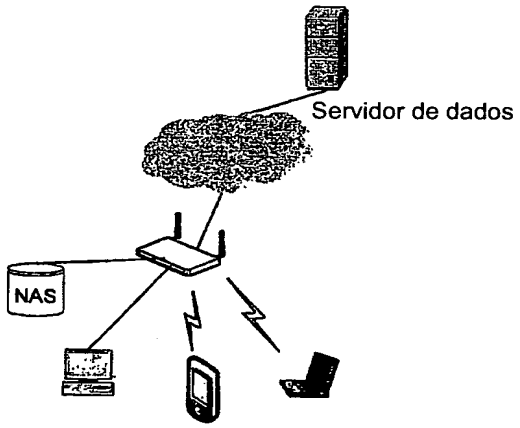


Fig. 1

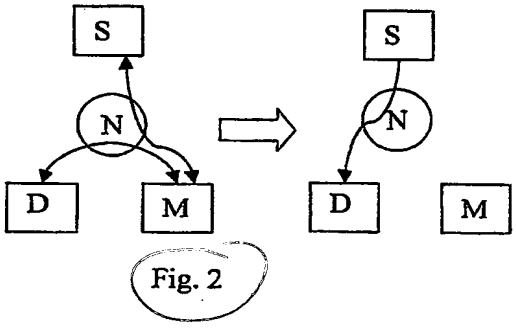


Fig. 2

RESUMO

"MÉTODO PARA ELIMINAR CONEXÕES REDUNDANTES"

Um método e sistema para eliminar conexões redundantes, incluindo transmitir uma solicitação para fundir duas conexões, terminar uma primeira porção de uma primeira conexão, terminar uma primeira porção de uma segunda conexão, e fundir uma segunda porção de cada uma das primeira e segunda conexões, são descritos. Um método e sistema para fundir conexões, incluindo receber uma solicitação para fundir duas conexões, terminar uma primeira porção de uma primeira conexão, terminar uma primeira porção de uma segunda conexão, e fundir uma segunda porção de cada uma das primeira e segunda conexões.