



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

(11) PI 0109097-6 B1



* B R P I 0 1 0 9 0 9 7 B 1 *

(22) Data de Depósito: 09/03/2001

(45) Data da Concessão: 28/07/2015
(RPI 2325)

(54) Título: MÉTODO E SISTEMA DE COMPRESSÃO E DESCOMPRESSÃO DE CAMPO DE CABEÇALHO BASEADO NO TEMPORIZADOR EM REDES MÓVEIS COM PROTOCOLO IP

(51) Int.CI.: H04L29/06

(52) CPC: H04L29/06

(30) Prioridade Unionista: 09/03/2000 US 09/522,363

(73) Titular(es): Nokia Corporation

(72) Inventor(es): Khiem Le

"MÉTODO E SISTEMA DE COMPRESSÃO E DESCOMPRESSÃO DE CAMPO DE CABEÇALHO BASEADO NO TEMPORIZADOR EM REDES MÓVEIS COM PROTOCOLO IP".

Campo da Invenção

[0001] A presente invenção relaciona a um método e aparelho para comprimir o campo do cabeçalho em um pacote de dados. Mais particularmente, a presente invenção relaciona a um método e a um aparelho para comprimir o campo do cabeçalho de um pacote de dados usando um Esquema Baseado na Referência e na Temporização.

Descrição da Técnica Anterior

[0002] Para a multimídia de tempo real baseada no Protocolo Internet (IP), o Protocolo de Transferência de Tempo Real (PTR) é predominantemente usado no topo do Protocolo de Datagrama do Usuário (PTR PDU/IP, é descrito em detalhes na RFC1889). O tamanho dos cabeçalhos IP/PDU/PTR combinados é de pelo menos 40 bytes para o IPv4 e de pelo menos 60 bytes para o IPv6. A sobrecarga de 40-60 bytes por pacote pode ser considerada pesada nos sistemas (ex, como redes celulares), onde a eficiência espectral é uma preocupação primária. Por conseguinte, existe uma necessidade para os mecanismos de compressão do cabeçalho IP/PDU/PTR adequados. O esquema de compressão do cabeçalho atual é descrito na RFC2508, o qual é capaz de comprimir o cabeçalho IP/PDU/PTR de 40/60 bytes abaixo de 2 ou 4 bytes sobre os enlaces ponto-a-ponto. Os algoritmos existentes de compressão do cabeçalho são baseados na observação de que a maioria dos campos dos cabeçalhos de pacote IP permanece constante em um fluxo de pacote durante o comprimento de uma sessão. Assim, é possível comprimir a informação do cabeçalho estabelecendo o estado de compressão (toda a informação do cabeçalho) no descompressor e transportando

simplesmente uma quantidade mínima de informação do cabeçalho do compressor para o descompressor.

[0003] A RFC2508 é baseada na idéia de que na maior parte do tempo, os campos PTR que trocam de um pacote para o próximo, tal como o protocolo de tempo PTR, pode ser preditos através da extrapolação linear. Essencialmente, a única informação que tem que ser enviada é um número sequencial, usado para a detecção de erro e de perda do pacote (como também o contexto ID). Quando o transmissor determina que a extrapolação linear não pode ser aplicada ao pacote atual, uma primeira ordem de informação de diferença com relação ao pacote imediatamente precedente é enviada. Para iniciar a sessão, todo o cabeçalho é enviado. Em adição, quando o receptor determina que há perda de pacote (como detectado pelo número sequencial incrementado por mais de 1), o receptor explicitamente pedirá ao transmissor para transmitir todo o cabeçalho para permitir a re-sincronização.

[0004] Contudo, o cabeçalho de compressão definido na RFC2508 não é adequado para certos ambientes (tal como, o ambiente celular ou sem fio), onde a largura de banda é na premium e os erros são comuns. No esquema de compressão do cabeçalho RFC2508, o protocolo de tempo PTR é assumido para ter na maior parte do tempo um padrão crescente linearmente. Quando o cabeçalho conforma ao padrão, essencialmente apenas um número sequencial curto é necessário no cabeçalho comprimido. Quando o cabeçalho não conforma ao padrão, a diferença entre os protocolos de tempo PTR do cabeçalho atual e do prévio é enviada no cabeçalho comprimido. Além disso, a otimização é possível usando uma tabela de codificação. Esta aproximação possui três desvantagens. A primeira é que esta não é robusta a erros, quando a perda do cabeçalho prévio invalidará a

descompressão do cabeçalho atual. A segunda é que as diferenças ou saltos do protocolo de tempo PTR podem ser muito grandes, deste modo, sobrecarregando a tabela de consulta de codificação. Por exemplo, se o meio for voz, tais diferenças grandes podem ser causadas por um intervalo de silêncio. A terceira é que o tamanho da diferença codificada resultante é variável, o que torna mais difícil prever e gerenciar a largura de banda a ser alocada.

[0005] Então, existe uma necessidade por um esquema de compressão do cabeçalho que pode acomodar um salto arbitrário no valor do campo (ex, no valor do protocolo de tempo PTR), produzindo um tamanho mais consistente ou constante, e é mais robusto a erros.

Resumo da Invenção

[0006] De acordo com uma incorporação da presente invenção, a técnica de descompressão do cabeçalho baseada no temporizador é provida. Uma fonte PTR gera o campo do cabeçalho, tal como o protocolo de tempo PTR. O protocolo de tempo é enviado sobre a rede para o compressor. No compressor, a função de redução da flutuação de fase (FRF) é usada para determinar se a flutuação de fase do protocolo de tempo recebido (cabeçalho) é excessiva. Se a flutuação de fase é excessiva, o pacote é descartado. Caso contrário, o compressor calcula o campo do cabeçalho comprimido (protocolo de tempo comprimido) baseado no protocolo de tempo PTR e no valor inicial do protocolo de tempo. O protocolo de tempo comprimido representa a flutuação de fase que é calculada como efeito à rede entre a fonte e o descompressor na transmissão de pacotes. A flutuação de fase calculada é uma acumulação da flutuação de fase da rede, representando o efeito da rede entre a fonte e o compressor na transmissão de pacotes e a flutuação de fase de rádio representando o efeito da rede

entre o compressor e o descompressor na transmissão de pacotes. Deveria ser observado que o termo "rede" como usado aqui é pretendido para ser um termo amplo de forma a não impedir, por exemplo, os enlaces de rádio em uma rede de telecomunicações sem fio. O pacote PTR, incluindo o protocolo de tempo comprimido, é então transmitido sobre o enlace ou da rede para o descompressor.

[0007] O descompressor descomprime o protocolo de tempo comprimido calculando primeiro uma estimativa ou aproximação do protocolo de tempo baseado no valor atual do temporizador localizado no terminal (isto é, baseado no tempo decorrido). A aproximação do protocolo de tempo é então refinada ou corrigida baseada no protocolo de tempo comprimido fornecido no cabeçalho de pacote. Desta maneira, o protocolo de tempo para o pacote atual (cabeçalho) é regenerado baseado em um temporizador local e o protocolo de tempo comprimido fornecido no cabeçalho atual. O pacote e o protocolo de tempo regenerado são então fornecidos para o ponto final PTR para processamento.

[0008] O esquema baseado no temporizador da presente invenção inclui várias vantagens. O termo "esquema baseado no temporizador" como usado aqui é inclusivo, o esquema baseado no temporizador usando um protocolo de tempo comprimido e o esquema baseado no temporizador e na referência como descrito aqui. O tamanho do protocolo de tempo comprimido (ou outro campo de cabeçalho) é constante e pequeno. Além disso, o tamanho não muda como uma função do comprimento do intervalo de silêncio. Nenhuma sincronização é requerida entre o processo do temporizador na fonte de PTR (gerando o protocolo de tempo) e o temporizador no processo do descompressor. Em adição, esta técnica é robusta a erros, como a informação de protocolo de tempo parcial no cabeçalho comprimido é mesma contida e

apenas necessita ser combinada com o valor do temporizador do descompressor para produzir o valor do protocolo de tempo PTR completo. A perda ou a corrupção de um cabeçalho não invalidará os cabeçalhos comprimidos subsequentes.

[0009] Uma segunda incorporação da presente invenção provê um esquema de retirada do cabeçalho, no qual o cabeçalho (ex., incluindo o protocolo de tempo PTR) é tirado ou removido do pacote de PTR antes da transmissão. O extrator do cabeçalho e o gerador do cabeçalho são conectados por um circuito, tal como uma conexão (ex., circuito ou circuito virtual) ou essencialmente de um canal de taxa de bits constante. Depois de iniciar, o extrator do cabeçalho retira ou remove o cabeçalho (incluindo remover o protocolo de tempo e o número sequencial) de cada pacote e então transmite os pacotes sem o cabeçalho para o regenerador do cabeçalho. Para eliminar a flutuação de fase do pacote no extrator do cabeçalho, os pacotes podem ser transmitidos no espaço de tempo de acordo com o protocolo de tempo PTR (PT) no cabeçalho. Então, nesta incorporação, o protocolo de tempo não é explicitamente fornecido no pacote PTR (nem mesmo um protocolo de tempo comprimido). Preferivelmente, a informação de temporização é fornecida implicitamente para o regenerador do cabeçalho baseado no canal de taxa de bits essencialmente constante entre o extrator e o regenerador do cabeçalho. O canal de taxa de bits essencialmente constante pode ser fornecido de vários modos diferentes.

[00010] Nesta segunda incorporação, depois que a inicialização ocorre (ex., fornecendo o número sequencial inicial e o protocolo de tempo ao regenerador do cabeçalho), o regenerador do cabeçalho pode regenerar o protocolo de tempo para os pacotes sequenciais incrementando o contador do protocolo de tempo local por

PT_progresso para todo T ms, e regenera os números sequenciais dos pacotes incrementando o contador NS local por 1 para toda a duração do pacote. Estes campos podem ser regenerados baseado apenas no temporizador local ou no contador devido ao canal de taxa de bit essencialmente constante, fornecido entre o extrator do cabeçalho e o regenerador do cabeçalho, nos quais nenhuma flutuação de fase de pacote é introduzida. Então, depois de iniciar, estes campos do cabeçalho podem ser regenerados no regenerador do cabeçalho com referência apenas ao relógio local.

[00011] Contudo, um ou mais eventos básicos de descontinuidade (ex., a troca no tamanho do pacote ou PT_progresso, uma troca não-linear no protocolo de tempo, etc.) pode ocorrer, se não endereçados, poderiam comumente invalidar a aproximação da extração do cabeçalho, que apenas confia no temporizador local ou no relógio para a regeneração do campo. A cadeia de cabeçalho é uma sequência de cabeçalhos de pacote possuindo campos conhecidos ou linearmente previsíveis. A transição de uma cadeia para outra pode ser causada por quaisquer dos vários eventos de descontinuidade. Quando isto ocorre, o extrator do cabeçalho identifica o evento de descontinuidade e envia a informação do cabeçalho atualizada relacionada ao evento para o regenerador do cabeçalho para permitir o protocolo de tempo e a regeneração do número sequencial para continuar. Uma técnica similar para fornecer a informação de cabeçalho atualizada pode ser usada quando também houver uma transferência.

Breve descrição dos Desenhos

[00012] A presente invenção se tornará mais aparente a partir da descrição detalhada a seguir, quando lida em conjunto com os desenhos apensos, nos quais:

Figura 1 - é um diagrama em blocos ilustrando o sistema de acordo com um exemplo de incorporação da presente invenção;

Figura 2 - é um diagrama ilustrando o formato não-comprimido do pacote PTR de acordo com uma incorporação da presente invenção;

Figura 3 - é um diagrama ilustrando o formato do cabeçalho PTR não-comprimido de acordo com um exemplo de incorporação da presente invenção;

Figura 4 - é um diagrama ilustrando o formato do cabeçalho PTR comprimido de acordo com um exemplo de incorporação da presente invenção;

Figura 5 - é um diagrama ilustrando um exemplo de operação de compressão e descompressão do cabeçalho de acordo com uma incorporação da invenção;

Figura 6 - é um diagrama ilustrando um exemplo de operação de compressão descompressão do cabeçalho de acordo com outra incorporação da invenção;

Figura 7 - é um diagrama ilustrando um exemplo de operação de transferência de acordo com uma incorporação da presente invenção;

Figura 8 - é um diagrama em blocos ilustrando um exemplo de uma pilha de acordo com uma incorporação da presente invenção;

Figura 9 - é uma tabela ilustrando a informação que pode ser fornecida nas mensagens de acordo com uma incorporação da invenção;

Figura 10 - é um diagrama ilustrando o processo de transferência de acordo com uma incorporação da presente invenção;

Figura 11 - é um diagrama ilustrando uma inicialização para a banda de acordo com uma incorporação da invenção;

Figura 12 - é um diagrama ilustrando uma inicialização para fora-de-banda de acordo com uma incorporação da invenção;

Figura 13 - é um diagrama ilustrando os passos de cálculo da flutuação da rede de acordo com o primeiro método da presente invenção;

Figura 14 - é um diagrama ilustrando os passos de cálculo da flutuação da rede de acordo com o segundo método agrupado como Opção 1 da presente invenção;

Figura 15 - é um diagrama ilustrando os passos de cálculo da flutuação da rede de acordo com o terceiro método agrupado como Opção 2 da presente invenção.

Melhor Modo de Executar a Invenção

I - Esquema Baseado no temporizador usando o Protocolo de Tempo Comprimido

Arquitetura

[00013] A Figura 1 é um diagrama em blocos ilustrando o sistema de acordo com uma incorporação da presente invenção. O terminal 102 é conectado a uma rede IP 108. O terminal 102 pode ser um computador pessoal ou similar rodando o PTR/PDU/IP, e fornecendo as amostras de voz empacotadas nos pacotes PTR para a transmissão sobre a rede 110. O terminal 102 inclui um ponto final PTR 104 que identifica este terminal (ex., incluindo o endereço IP, o número da porta, etc.) como uma fonte ou destino para os pacotes PTR. A rede IP é fornecida como um exemplo, porém, outros tipos de redes comutadas por pacote ou similares podem ser usados. Deveria ser observado que o termo "rede" como usado aqui é pretendido para ser um termo amplo para não impedir, por exemplo, os enlaces de rádio na rede de telecomunicações sem fio. O terminal 102 também inclui um temporizador local 103 para gerar um protocolo de tempo.

[00014] A infraestrutura da rede de acesso (IRA) 110 é conectada a rede IP 108. O terminal sem fio 130 é acoplado pelo enlace de frequência de rádio (RF) 140 à IRA 110. O terminal sem fio 130 como descrito aqui poderia, por exemplo, ser um compressor sem fio ou um descompressor sem fio dependendo do seu ambiente. Isto ocorre particularmente quando a fonte dos pacotes ou o destino dos pacotes são separados do terminal sem fio 130. O enlace RF 140 inclui um enlace ascendente 142 (do terminal 130 para IRA 110) e um enlace descendente 144 (da IRA 110 para o terminal 130). A IRA 110 conecta um ou mais terminais (incluindo o terminal 130) sem fio (ou frequência de rádio) em uma região para a rede IP 108, incluindo a conversão entre sinais a cabo (fornecido da rede IP 108) e sem fio ou sinal de RF (fornecido para ou do terminal 130). Assim, a IRA 110 permite que os pacotes PTR recebidos da rede IP 108 sejam enviados sobre o enlace RF 140 para o terminal sem fio 130, e permite que os pacotes PTR do terminal 130 sejam enviados sobre a rede IP 108 para outro terminal, tal como o terminal 102.

[00015] De acordo com uma incorporação da presente invenção, a IRA 110 inclui um ou mais adaptadores de IRA (IRA_AD), tal como IRA_AD 112 e IRA_AD 114 cada um dos quais preferivelmente inclui um temporizador. Cada IRA_AD executa a compressão do cabeçalho (antes da transmissão descendente) e a descompressão (depois da transmissão ascendente). Os cabeçalhos (ou um ou mais campos do cabeçalho, tal como o protocolo de tempo) para pacotes PTR recebidos da rede IP 108 é comprimido pela IRA_AD 112 antes da transmissão para o terminal 130 sobre o enlace descendente 142, e os cabeçalhos de pacote recebidos do terminal 130 são descomprimidos pelo IRA_AD 112 antes da transmissão para a rede IP 108. Então, cada IRA_AD pode ser

considerada para ser um compressor/descompressor 115. Cada IRA_AD pode conectar os terminais localizados em uma área específica ou diferente dentro da região da rede IP 108. A IRA DC 112 inclui um temporizador 113 para implementar a técnica de descompressão baseada no temporizador. A IRA_AD 112 também inclui a função de redução de flutuação de fase (FRF) 115, que opera para medir a flutuação nos pacotes (ou cabeçalhos) recebidos sobre a rede 108 descartando quaisquer pacotes/cabeçalhos que têm flutuação excessiva.

[00016] IRAs adicionais, tal como a IRA 120, pode ser provida para conectar outros terminais localizados nas regiões adicionais à rede IP 108. A IRA 120 inclui uma ou mais IRA_ADS, tal como IRA_AD 122 (Figura 1). Cada IRA_AD inclui um temporizador e um FRF.

[00017] O terminal 130 inclui um ponto final PTR 132, o qual é uma fonte e/ou destino (receptor) para os pacotes PTR. O terminal 130 inclui um adaptador terminal (termo_AD) 136, o qual executa a compressão do cabeçalho (para os pacotes a serem transmitidos no enlace ascendente 142) e a descompressão (nos pacotes recebidos sobre o enlace descendente 144). Assim, o adaptador terminal (termo_AD) pode ser considerado um compressor/descompressor de cabeçalho 137, similar ao IRA_AD.

[00018] O adaptador terminal (termo_AD) 136 também inclui um temporizador 134 (um temporizador de recepção) para calcular a aproximação (ou estimativa) de um protocolo de tempo PTR do cabeçalho atual. O adaptador terminal (termo_AD) 136 então usa a informação adicional no cabeçalho PTR para refinar ou corrigir a aproximação do protocolo de tempo. De acordo com uma incorporação da invenção, a aproximação do protocolo de tempo é corrigida ou ajustada baseada no protocolo de tempo comprimido fornecido no cabeçalho PTR. Desta maneira, o temporizador

local e o protocolo de tempo comprimido podem ser usados para regenerar o protocolo de tempo correto para cada cabeçalho PTR. Outros terminais (tal como o terminal 150) podem ser fornecidos, cada um inclui o seu próprio ponto final PTR, o adaptador terminal e o temporizador.

[00019] A configuração apresentada na Figura 1 é fornecida apenas como um exemplo e, a invenção não está limitada a esta. Preferivelmente, a Figura 1 simplesmente provê um exemplo onde os dados PTR são transmitidos sobre o enlace de dados ou sistema (tal como o enlace sem fio 140), onde a largura de banda é no prêmio e os erros não são incomuns. A presente invenção não está limitada ao enlace sem fio, mas é aplicável a uma ampla variedade de enlaces (incluindo os enlaces cabeados, etc.).

[00020] Um exemplo de aplicação ou de sistema onde a compressão do cabeçalho baseada no temporizador e no esquema de descompressão pode ser útil é onde os pacotes de voz sobre IP (ou telefonia-IP) são transmitidos sobre os sistemas celulares. Quando o VoIP é aplicado nos sistemas celulares, é importante minimizar o suplemento do cabeçalho IP/PDU/PTR devido à largura de banda limitada da interface aérea (RF) ou sem fio. Em tal sistema, por exemplo, a IRA_AD conectaria a rede IP a um terminal de computador rodando PTR/PDU/IP (ex., terminal 130) e possuindo uma interface celular ou RF para receber os pacotes PTR sobre o enlace sem fio ou RF. Esta é meramente um exemplo da aplicação da técnica de compressão/descompressão da presente invenção.

[00021] A Figura 2 é um diagrama ilustrando o formato de um pacote PTR não comprimido de acordo com uma incorporação da presente invenção. Como apresentado na Figura 2, o pacote PTR não comprimido inclui um cabeçalho

IP, um cabeçalho PDU 212, um cabeçalho PTR 214 e a carga útil que poderia ser uma amostra de voz 216.

[00022] A Figura 3 é um diagrama ilustrando o formato do cabeçalho PTR não comprimido de acordo com uma incorporação da presente invenção. Como apresentado na Figura 3, o cabeçalho PTR não comprimido inclui o protocolo de tempo (PT) 310, o número sequencial (NS) 312 e alguns outros campos 314. Devido à natureza de pacote comutada da rede IP 108, os pacotes PTR podem chegar defeituosos. O número sequencial 312 é usado no receptor PTR ou no destino PTR (ex., terminal 130, Figura 1) para montar as amostras de voz PTR na ordem correta. Contudo, os números sequenciais nos pacotes PTR não refletirão nenhuma mudança não-linear no campo (ex., intervalos de silêncio do sinal de voz). Então, o protocolo de tempo (PT) 310 é fornecido para indicar a temporização relativa de cada pacote.

[00023] Como observado acima, existe um pouco de interesse de um cabeçalho auxiliar de 40-60 bytes fornecidos pelo cabeçalho IP/PDU/PTR em cada pacote PTR é muito grande. Em particular, um protocolo de tempo PTR de 4 bytes é particularmente penoso para os pacotes PTR operando sobre baixa velocidade ou aos enlaces de largura de banda limitada (tal como o enlace 140). Como resultado, existe uma necessidade para um mecanismo efetivamente de compressão dos cabeçalhos PTR e particularmente a compressão do campo do protocolo de tempo no cabeçalho PTR.

[00024] A técnica de compressão de cabeçalho descrita inicialmente na RFC2508 envia um pacote PTR completo (não comprimido), incluindo todos os campos de destino/receptor PTR. Vários dos campos dos cabeçalhos durante uma conexão são estáticos, e assim, não precisam ser transmitidos após o pacote inicial ser enviado e recebido. Para a maioria dos pacotes, apenas o número

sequencial e o protocolo de tempo mudarão de pacote para pacote. De acordo com a RFC2508, os campos não-estáticos (ex., protocolo de tempo e o número sequencial) serão atualizados no receptor adicionando (fixo) das diferenças da primeira ordem para os valores prévios destes campos armazenados no receptor. Por exemplo, o número sequencial de cada pacote PTR recebido será automaticamente incrementado por um para cada pacote. Saltos ou trocas adicionais (isto é, diferente da primeira diferença de ordem) nos campos não-estáticos devem ser transmitidos separadamente para o receptor. Infelizmente, na RFC2508, a perda do cabeçalho prévio invalidará a descompressão no receptor. Além disso, o tamanho das diferenças varia, o que torna mais difícil de gerenciar e predizer a largura de banda usando a técnica de compressão da RFC2508.

[00025] De acordo com uma incorporação da presente invenção, a técnica para compressão do cabeçalho é fornecida para que possa ser usada para comprimir o protocolo de tempo PTR (ou outro campo) mais efetivamente de um cabeçalho do pacote. De acordo com uma incorporação da presente invenção, o esquema de compressão pode acomodar um salto arbitrário no valor do protocolo de tempo PTR, enquanto rendendo um cabeçalho PTR comprimido de tamanho constante (ou protocolo de tempo PTR de tamanho constante).

[00026] A Figura 4 é um diagrama ilustrando o formato do cabeçalho PTR comprimido de acordo com uma incorporação da presente invenção. Como apresentado na Figura 4, o cabeçalho PTR comprimido pode consistir em um tipo de mensagem 410 indicando o tipo de mensagem, a máscara de bit 412 a qual identifica os campos que estão mudando, e o campo do protocolo de tempo comprimido 414. O tipo de mensagem 410 pode indicar um protocolo de tempo comprimido se um protocolo de tempo comprimido for

fornecido no cabeçalho do pacote. De acordo com uma incorporação da presente invenção, o campo do protocolo de tempo comprimido 414 inclui os bits menos significativos k (lsbs) de um valor que pode indicar o tempo decorrido entre os pacotes. De acordo com uma incorporação da invenção, o protocolo de tempo comprimido 414 pode prover uma parte (isto é, os bits k menos significativos) do valor do contador da fonte (ou diferença do contador). O contador da fonte pode ser usado para gerar o protocolo de tempo para cada cabeçalho do pacote PTR. Os campos opcionais 416 podem ser usados para prover os campos atualizados ou alterados para estes campos identificados na máscara de bit 412.

Operação Completa de Compressão e Descompressão do Protocolo de Tempo

[00027] A compressão e a descompressão do protocolo de tempo PTR serão descritas brevemente, de acordo com uma incorporação da invenção. De acordo com uma incorporação, o pacote PTR é gerado no ponto final PTR (tal como o ponto final PTR 104 do terminal 102) e é enviado a outro ponto final PTR. Neste exemplo, o ponto final PTR 104 é uma fonte de um ou mais pacotes PTR a serem enviados para o ponto final PTR 132 (destino) do terminal 130. O cabeçalho do pacote PTR inclui um protocolo de tempo que é gerado na fonte PTR (ex., o terminal 102) baseado no relógio de parede.

[00028] O pacote PTR é direcionado sobre a rede IP 108 para a IRA_AD 112 da IRA 110. A IRA_AD 112 comprime um ou mais campos no cabeçalho(s) do pacote PTR. Em particular, a IRA_AD comprime o protocolo de tempo PTR 310 (Figura 3) no protocolo de tempo comprimido 414 (Figura 4). Outros campos no cabeçalho podem ser comprimidos pela remoção deles ou usando alguma outra técnica. O pacote PTR,

incluindo o protocolo de tempo comprimido 414, é então transmitido sobre o enlace descendente 144 do enlace RF 140 para o terminal 130.

[00029] Após a recepção do pacote PTR com o cabeçalho comprimido (isto é, protocolo de tempo comprimido 414), o adaptador terminal (termo_AD) 136 do terminal 130 descomprime o valor do protocolo de tempo. O adaptador terminal 136 descomprime o protocolo de tempo comprimido 414 calculando uma estimativa ou aproximação do primeiro protocolo de tempo baseado no valor atual do temporizador 134. A aproximação do protocolo de tempo é então refinada ou corrigida com base no protocolo de tempo comprimido 414 fornecido no cabeçalho do pacote. Desta maneira, o protocolo de tempo para o pacote atual (cabeçalho) é regenerado baseado no temporizador local (temporizador 134) e no protocolo de tempo comprimido fornecido no cabeçalho atual. Os outros campos do cabeçalho do pacote (como o número sequencial) também podem ser regenerados. O pacote e o protocolo de tempo regenerado são então fornecidos ao ponto final PTR 132 para processamento. O ponto final PTR 132 então retransmite as amostras de voz na ordem apropriada (como especificado pelos números sequenciais) e tendo a temporização apropriada como especificado pelo protocolo de tempo regenerado (ex., calcular para qualquer intervalo de silêncio).

[00030] A IRA_AD 112 também pode receber os cabeçalhos comprimidos (incluindo o protocolo de tempo comprimido) sobre o enlace RF 140, e descomprime o protocolo de tempo usando a técnica de descompressão baseada no temporizador descrito acima. Então, a IRA_AD 112 pode incluir tipicamente um temporizador para permitir que IRA_AD descomprima o protocolo de tempo comprimido como descrito acima. Similarmente, o termo_AD 136 do terminal

130 também podem comprimir o protocolo de tempo do pacote PTR antes da transmissão do pacote PTR sobre o enlace RF 140 para a IRA 110. Para simplificar a explicação das incorporações exemplares da invenção, a maior parte da descrição será direcionada ao caminho descendente 144. De acordo com uma incorporação da invenção, os pacotes PTR podem ser transmitidos em ambas as direções (ascendente 142 e descendente 144). Assim, ambos o IRA_AD 112 da IRA 110 e o termo_AD do terminal 130 podem operar como um compressor do protocolo de tempo (para transmissão do cabeçalho/pacote sobre o enlace RF) e um descompressor (depois do recebimento do cabeçalho comprimido recebido sobre o enlace RF 140).

Exemplos de Incorporações de Compressão e Descompressão do Protocolo de Tempo.

[00031] Exemplos de incorporações de compressão e descompressão do protocolo de tempo serão descritos brevemente. É assumido que os dados nos pacotes PTR são dados de voz. As variáveis a seguir e as fórmulas são definidas apenas para ajudar na explicação de algumas das características da presente invenção, mas a invenção não está limitada a esta. Além disso, a presente invenção não está limitada aos sistemas que usam os mesmos ou tipos similares de variáveis, e não está limitada aos sistemas que executam os cálculos específicos descritos abaixo. As variáveis e os cálculos são meramente fornecidos como exemplo de uma incorporação da invenção.

[00032] T - é o espaço de tempo entre as amostras de voz PTR. (Se houver uma amostra de voz fornecida em cada pacote PTR, então o T também é o espaço entre os cabeçalhos do pacote PTR).

[00033] PT - protocolo de tempo.

[00034] PT_progresso - o protocolo de tempo PTR é incrementado por PT_progresso para todo T ms. Em outras palavras, o protocolo de tempo PTR aumenta PT_progresso para todo pacote PTR novo. O PT_progresso é uma constante (ex., 100) que depende do codec de voz. O PT_progresso é fornecido ao receptor (terminal 130) e para a IRA_AD 112.

[00035] PT0 - protocolo de tempo PTR do primeiro cabeçalho da sessão recebida no receptor PTR. O primeiro cabeçalho da sessão é considerado como um cabeçalho de sincronização porque este é usado para sincronização. PT0 é o valor inicial do protocolo de tempo PTR fornecido a ambos, o compressor (ex., IRA_AD 112) e o descompressor (ex., termo_AD 136) no começo da sessão (para sincronização). De acordo com uma incorporação, a IRA_AD e o termo_AD são iniciados ou sincronizados pela recepção de um pacote PTR com o cabeçalho não comprimido (incluindo o protocolo de tempo não comprimido fornecendo PT0). De acordo com uma incorporação da presente invenção, a técnica de descompressão baseada no temporizador requer o fornecimento de um protocolo de tempo inicial PT0 (ex., através de um cabeçalho inicial ou de sincronização que são não-comprimidos) para o compressor do protocolo de tempo (isto é, IRA_AD 112) e para o descompressor (isto é, termo_AD 136) antes dos cabeçalhos comprimidos poderem ser descomprimidos corretamente (isto é, assim o descompressor pode regenerar o protocolo de tempo corretamente).

[00036] O protocolo de tempo PTR do cabeçalho do pacote m (gerado no tempo $m \cdot T$ ms) = $PT0 + PT_progresso \cdot m$. Assume-se que existe um cabeçalho para cada amostra de voz. Como apresentado nos exemplos descritos abaixo, esta fórmula pode ser estendida para o caso de amostra de voz múltipla (ex., 3 amostras de voz) por cabeçalho do pacote.

[00037] m - é um inteiro que indica o número de amostras de voz que foram enviadas. m é reajustado ou limpo para 0 no começo da sessão. m é proporcional a (ou indica) a quantidade de tempo decorrido desde o começo da sessão, m é incrementado por 1 para todo T ms.

[00038] $PT_{atual} = PT0 + m_{atual} * PT_{progresso}$ - O protocolo de tempo atual para o cabeçalho do pacote atual.

[00039] Temporizador do receptor - o temporizador no receptor PTR (ou destino PTR), tal como o temporizador 134 do terminal 130. O temporizador do receptor local está rodando tipicamente livre e não será reajustado no começo da sessão. Preferivelmente, o tempo decorrido no receptor PTR entre o recebimento dos dois cabeçalhos do pacote pode ser obtido subtraindo o valor do temporizador do cabeçalho atual do valor do temporizador do receptor, quando o cabeçalho do pacote prévio foi recebido. Através da permissão, o temporizador do receptor está livre para rodar, um temporizador do receptor pode ser compartilhado por muitos fluxos ou sessões. Alternativamente, o temporizador do receptor pode ser reajustado no começo de cada sessão. Ao restaurar ou limpar o temporizador do receptor no começo da sessão (isto é, após a recepção do cabeçalho de inicialização) requereria um temporizador do receptor dedicado (processo de temporizador) para cada sessão ou fluxo. O primeiro protocolo de tempo não-comprimido ($PT0$) da sessão pode ser fornecido para a IRA_AD e o termo AD no cabeçalho de inicialização. O primeiro cabeçalho é fornecido para iniciar o compressor (IRA_AD 112) e o descompressor (termo AD 136). O temporizador do receptor é então incrementado por 1 a todo T ms. A IRA_AD 112 (compressor) usa o valor $PT0$ para comprimir o protocolo de tempo dos

cabeçalhos subsequentes do pacote PTR. O termo_AD 136 (descompressor) usa o valor PT0 para descomprimir o valor do protocolo de tempo comprimido (ex., para regenerar o protocolo de tempo nos cabeçalhos PTR subsequentemente recebidos).

[00040] Temporizador_atual - o valor do temporizador no receptor PTR (ex., terminal 130) quando o cabeçalho atual é recebido.

[00041] Último_temporizador - o valor no tempo no receptor quando o último cabeçalho foi recebido. (O temporizador_atual é armazenado como o último_temporizador para o próximo cálculo do cabeçalho do protocolo de tempo).

[00042] m_último - o valor de m para o último cabeçalho recebido; m indica o número de quadros de voz que tem decorrido desde o cabeçalho de inicialização.

[00043] Para comprimir o protocolo de tempo do pacote atual, a IRA_AD 112 calcular o valor atual de m como: $m_{atual} = (PT_{atual} - PT_0) / PT_{progresso}$. Então, a IRA_AD subtrai o valor inicial do protocolo de tempo (no começo da sessão) do protocolo de tempo atual. Esta diferença é dividida pelo progresso do protocolo de tempo (PT_progresso). Porém, em algumas incorporações, pode ser desnecessário executar atualmente a operação de divisão. Outras técnicas podem ser usadas para gerar adequadamente m_atual sem executar a operação de divisão, a qual pode requerer um alto processador auxiliar.

[00044] Os bits k menos significativos do m_atual são então fornecidos como o protocolo de tempo comprimido 414. O pacote PTR incluindo o protocolo de tempo comprimido 414 é então transmitido sobre o enlace RF 140 para o destino PTR ou receptor (ex., terminal 130).

[00045] No receptor PTR (ex., terminal 130), o adaptador terminal (termo_AD) 136 descomprime o protocolo

de tempo comprimido 414. O valor do temporizador_atual do cabeçalho prévio é primeiro armazenado como o último_temporizador. Então, quando o cabeçalho atual chega, o termo_AD 136 lê o valor do temporizador do receptor 134 e armazena este na memória como temporizador_atual. Logo, dif_temporizador é calculado como: $\text{dif_temporizador} = \text{temporizador_atual} - \text{temporizador_último}$. A IRA_AD calcula o valor exato do m_atual encontrando o inteiro d onde:

$$(-L/2 < d \leq L/2 \text{ onde } L=2k) \quad (\text{Eq. 1})$$

tal que bits k menos significativos de:

$$d + m_último + \text{dif_temporizador}) = \text{protocolo de tempo comprimido 414 (para cabeçalho atual)}. \quad (\text{Eq. 2})$$

[00046] Como observado, o protocolo de tempo comprimido recebido também é k bits. Uma vez de ter sido calculado usando as Equações 1 e 2, o PT_atual pode então ser calculado como:

$$\text{PT_atual} = \text{PT0} + (d + m_último + \text{dif_temporizador}) * \text{PT_progresso} \quad (\text{Eq. 3})$$

[00047] Na equação 3, o valor atual ou correto do m_atual é apresentado nos parênteses como $(d + m_último + \text{dif_temporizador})$. $m_último + \text{temporizador}$ é a aproximação do m_atual, enquanto d é a diferença entre a aproximação do m_atual e o valor correto do m_atual. Além disso, $\text{PT0} + (m_último + \text{dif_temporizador}) * \text{PT_progresso}$ é uma aproximação do valor do protocolo de tempo atual, e $d * \text{PT_progresso}$ é a diferença entre o protocolo de tempo atual aproximado e o valor atual (ou correto) do protocolo de tempo atual.

[00048] Então, pode ser visto que o receptor PTR primeiro calcula uma aproximação (ou estimativa) do protocolo de tempo atual baseado no tempo decorrido entre o recebimento do cabeçalho atual e do cabeçalho prévio (que foi descomprimido corretamente), como: protocolo de tempo atual aproximado = $PT0 + (m_{\text{último}} + \text{dif_temporizador}) * PT_{\text{progresso}}$. O protocolo de tempo atual aproximado é então ajustado ou corrigido pela quantidade $d * PT_{\text{progresso}}$ para calcular o valor do protocolo de tempo atual correto (PT_{atual}).

[00049] Após o PT_{atual} ser calculado, o pacote PTR atual (incluindo o seu protocolo de tempo regenerado ou descomprimido, PT_{atual}) é fornecido até o ponto final PTR 132. Este processo de compressão e de descompressão é transparente aos pontos finais PTR.

[00050] A Figura 5 é um diagrama ilustrando um exemplo da operação de compressão e descompressão do cabeçalho de acordo com uma incorporação da invenção. Este exemplo aplica algumas das fórmulas específicas descritas acima para ilustrar algumas características da presente invenção. Neste exemplo da incorporação, é assumido que os temporizadores na fonte PTR 502 e no receptor PTR 504, possuindo a mesma frequência, mas não é tipicamente sincronizado. O temporizador na fonte PTR (ex., incrementando por 1 todo T ms) é usado para gerar o protocolo de tempo, enquanto o temporizador (ex., temporizador 134) no receptor PTR é usado para regenerar ou descomprimir o protocolo de tempo PTR.

[00051] Referindo à Figura 5, no começo da sessão, um cabeçalho de inicialização 508 é gerado na fonte PTR, incluindo um valor do protocolo de tempo inicial ($PT0$). O cabeçalho de inicialização 508 é transmitido para a IRA e então remetido para o receptor PTR 504 (ex.,

terminal 130). O protocolo de tempo no cabeçalho de inicialização não está comprimido. Após a recepção do cabeçalho de inicialização, o valor do protocolo de tempo inicial (PT0) é armazenado na memória na IRA_AD, junto com o PT_progresso. De acordo com uma incorporação, os dois cabeçalhos de inicialização podem ser transmitidos para a IRA_AD. A IRA_AD pode calcular então PT_progresso como o segundo protocolo de tempo - primeiro protocolo de tempo. O termo_AD pode calcular similarmente o PT_progresso ou receber o valor no pacote.

[00052] Similarmente, quando o cabeçalho de inicialização 508 é recebido no receptor PTR (terminal 130), o protocolo de tempo inicial (PT0) é armazenado na memória junto com o PT_progresso. Além disso, no recebimento 510 do cabeçalho de inicialização 508 (Figura 5), o m_atual é limpo ou reajustado para zero (0), e o temporizador do receptor é então lido e armazenado como temporizador_receptor_inicial 516. Em vez de ler, o temporizador no começo da sessão, o temporizador do receptor pode ser reajustado ou limpo. Neste exemplo, o valor lido do temporizador do receptor no começo da sessão apenas ocorre para ser zero (0) para simplicidade. Assim, o exemplo apresentado na Figura 5 se aplica para ambas as incorporações (apenas ler o temporizador do receptor, ou restaurar este para zero), porque o temporizador rodando livre é lido como zero no começo da sessão. Igualmente, não é necessário limpar o m_atual, mas ao invés poderia registrar um valor para m_atual. O temporizador do receptor é então incrementado (ex., por 1) para todo T ms (que é a mesma frequência que a do temporizador na fonte PTR 502 usada para gerar o protocolo de tempo). O cabeçalho de inicialização 508 chega ao receptor PTR 502 após um retardo

fixo (tamanho do retardo 512) e um retardo variável (flutuação acumulativa 514).

[00053] Logo, a fonte PTR 502 gera o próximo pacote PTR (o primeiro pacote PTR da sessão depois do cabeçalho de inicialização). Este pacote PTR é gerado a $3 \cdot T$ ms depois que o cabeçalho de inicialização foi gerado, e assim incluiria tipicamente, por exemplo, três (3) amostras de voz. Outros números são possíveis. Então, o protocolo de tempo para o cabeçalho deste pacote é: $PT(1) = PT_0 + 3 \cdot PT_progresso$, como apresentado na Figura 5. $PT(1)$ refere ao protocolo de tempo gerado depois de $3T$ ms após a inicialização. Neste exemplo, será assumido, por exemplo, que o $PT_progresso$ é 100. É assumido que PT_0 é 0, por exemplo. Assim, $PT(1) = 300$.

[00054] O valor do protocolo de tempo para este pacote, $PT(1)$, é recebido no IRA_AD e é comprimido baseado no $PT(1)$ (o valor do protocolo de tempo), PT_0 (o valor do protocolo de tempo inicial) e $PT_progresso$ (a quantidade de protocolo de tempo incrementa para todo T ms). De acordo com um exemplo de incorporação, o protocolo de tempo comprimido pode ser calculado como os bits k menos significativos do m_atual . A IRA_AD 112 calcula o valor atual de m como: $m_atual = (PT_atual - PT_0) / PT_progresso$. Neste exemplo, será assumido que $PT_progresso$ é 100, por exemplo. Neste exemplo, m_atual é calculado como: $m_atual = (300 - 0) / 100 = 3$. k neste exemplo será dois (2). Assim, os dois bits menos significativos do m_atual (11 em binário) são fornecidos como protocolo de tempo comprimido 414 para este pacote (PTC_1), Figura 5.

[00055] O protocolo de tempo comprimido (PTC_1) chega ao receptor PTR 502 e o termo_AD 136 no receptor PTR regenera ou descomprime o protocolo de tempo, $PT(1)$, para o pacote atual. O valor do temporizador_atual (zero) é

armazenado como `último_temporizador` e `m_atual` é armazenada como `m_último`, `m_atual` foi previamente estabelecido em zero no começo da sessão (isto é, após a recepção do cabeçalho de sincronização). O valor do temporizador do receptor (3 neste caso) é lido e armazenado como `temporizador_atual`. `dif_temporizador` é então calculada como `temporizador_atual - temporizador_último`, que é $3 - 0 = 3$. `dif_temporizador + m_último` é uma aproximação de `m_atual`.

[00056] A seguir o termo `_AD 136` calcula o valor exato ou corrigido para o `m_atual` usando as equações (1) e (2). Usando a equação (2), os dois bits menos significativos de $(d + m_último + dif_temporizador) = PTC\ 1$ (o protocolo de tempo comprimido para o cabeçalho atual). Neste caso, `m_último` é zero (0), a `dif_temporizador` é três (3) e `PTC 1` é três (3). Assim, os dois bits menos significativos de $(d + 0 + 3) = 3$. Assim, `d` é igual a zero.

[00057] Usando a equação (3), o protocolo de tempo descomprimido para este pacote é então calculado como $PT(1) = PT0 + (d + m_último + dif_temporizador) * PT_progresso$. Como resultado, então $PT(1) = 0 + (0 + 0 + 3) * 100 = 300$. O protocolo de tempo descomprimido para este pacote, $PT(1) = 300$, é então fornecido ao ponto final `PTR 132` na fonte `PTR`, junto com os dados `PTR` e outros campos do cabeçalho descomprimidos. O valor correto ou atual para `m_atual` é $(d + m_último + dif_temporizador)$. Então, para este pacote, pode ser visto que a aproximação de `m_atual` é igual ao valor correto de `m_atual` (mas isto não é verdade no caso geral). O `m_atual` é então atualizado para ser 3.

[00058] O próximo pacote e o protocolo de tempo são gerados na fonte `PTR`, incluindo o protocolo de tempo $PT(2) = 0 + 6 * 100 = 600$. A `IRA_AD`, o $PT(2) = 600$ é comprimido em um protocolo de tempo comprimido como os 2

bits menos significativos de $(600-0)/100 = 6$. Neste caso, 6 em binário é 110. Assim, os dois bits menos significativos de 110 é 10. Assim, $PTC2 = 10$ em binário.

[00059] O protocolo de tempo comprimido para este pacote ($PTC2$) é então recebido no termo_AD 136 depois que o temporizador do receptor alcança o valor de 7, devido ao retardo do tamanho e a flutuação acumulativa. O valor do temporizador_atual (3) é armazenado como último_temporizador e, m_{atual} (3) é armazenado como $m_{último}$. O valor do temporizador do receptor atual (7 neste caso) é lido e armazenado como temporizador_atual. $dif_temporizador$ é então calculada como $temporizador_atual - temporizador_último$, que é $7 - 3 = 4$. $dif_temporizador + m_{último}$ é uma aproximação do m_{atual} , que é 7.

[00060] A seguir, o termo_AD 136 calcula o valor exato ou corrigido para m_{atual} usando as equações (1) e (2). Usando a equação (2), os dois bits menos significativos de $(d + m_{último} + dif_temporizador) = PTC\ 2$ (o protocolo de tempo comprimido para o cabeçalho atual). Neste caso, $m_{último}$ é 3, a $dif_temporizador$ é 4 e $PTC2$ é 10 (em binário, que é 3 em decimal). A equação 2 é resolvida para d como a seguir: $2\ Isbs\ (d + 3 + 4) = 2$. Sete em binário é 111. Assim, $d = -1$. d é a diferença entre a aproximação do m_{atual} e o valor atual de m_{atual} . Colocando d na equação (3), o protocolo de tempo para este pacote é calculado como $PT(2) = 0 + (-1 + 3 + 4) * 100 = 600$. Assim, o termo_AD do receptor PTR regenera corretamente (ex., descomprimido) o protocolo de tempo PTR baseado em um temporizador local e o protocolo de tempo comprimido.

[00061] Deveria ser observado que, as técnicas prévias distintas, são desnecessárias para reenviar o cabeçalho de inicialização no evento no qual um ou mais

pacotes não chegam ao receptor PTR. Em outras palavras, a sincronização entre a fonte PTR e o receptor, é apenas necessária uma vez no começo da sessão ou conexão. Isto é porque, o protocolo de tempo atual é calculado no receptor PTR baseado no `m_último` e na `dif_temporizador`. A `dif_temporizador` é calculada como `temporizador_atual - último_temporizador`. então, os valores de `m_último` e do `último_temporizador` correspondem ao último pacote, indiferente do pacote que foi recebido por último (ex., indiferentemente se os pacotes enviados após o "último" pacote foram erroneamente retirados ou perdidos). Como resultado, o esquema de compressão baseado no temporizador de acordo com uma incorporação da invenção é robusto a erros e diminui as exigências de largura de banda, porque é desnecessário enviar um pacote de sincronização novo (ex., incluindo os valores completos não-comprimidos para todos os cabeçalhos) no evento onde um erro é detectado (ex., um ou mais pacotes retirados ou perdidos).

[00062] Em uma operação normal, a discrepância entre a aproximação e o valor exato do `m_atual` é causada por:

- a) Flutuação acumulativa entre a mesma fonte do protocolo de tempo PTR e o receptor; retardo atual = retardo de tamanho + flutuação acumulativa, onde o tamanho do retardo é constante e a flutuação acumulativa varia de um cabeçalho para o próximo, e $0 \leq \text{flutuação acumulativa} \leq \text{flutuação máxima acumulativa}$; e
- b) Possível assincronismo entre o processo do temporizador e o processo do descompressor, dependendo da implementação do temporizador. devido ao assincronismo, pode haver um erro mais

ou menos ou um (+ ou -1) no valor do temporizador (temporizador_atual).

[00063] A Figura 6 é um diagrama ilustrando um exemplo de operação de compressão e descompressão do cabeçalho de acordo com uma outra incorporação da invenção. Como da Figura 5, a Figura 6 é um diagrama ilustrando o efeito de flutuação e do assincronismo do temporizador. Na Figura 5, o temporizador do receptor é reajustado ou limpo apenas no começo da sessão. (isto não é necessário uma vez que o temporizador do receptor pode ser permitido apenas para continuar a rodar.) Porém, no exemplo da incorporação ilustrada na Figura 6, o temporizador do receptor é reajustado ou limpo para zero (0) para cada pacote. Assim, quando o cabeçalho do pacote comprimido é recebido, o valor do temporizador é lido indicando o valor da dif_temporizador descrito acima (uma vez que o temporizador indica o tempo decorrido desde o último cabeçalho do pacote). Pode haver muitos modos diferentes para implementar a invenção. O que é importante é que a diferença do temporizador deveria ser medida indicando o tempo decorrido (como medido pelo temporizador do receptor local) entre o último protocolo de tempo descomprimido com sucesso e o protocolo de tempo atual (dif_temporizador descrito na Figura 5).

[00064] Referindo à Figura 6, o cabeçalho n é gerado como o protocolo de tempo = $PT0 + 3*PT_progresso$. Este protocolo de tempo do cabeçalho n é comprimido e transmitido para o receptor PTR, e descomprimido. O temporizador é então reajustado para o receptor. Os próximos cabeçalhos, (n+1), (n+2) e (n+3) são gerados e enviados, mas apenas o cabeçalho (n+3) é recebido (isto é, n+1 e n+2 do cabeçalho são perdidos). Para simplificar, o

cabeçalho de (n+2) e (n+3) não são apresentados na Figura 6. O cabeçalho (n+1) é apresentado na Figura 6 como cabeçalho m+n. O cabeçalho (m+n) é gerado e enviado, com o protocolo de tempo $PT = PT_0 + 6 * PT_{\text{progresso}}$. Este protocolo de tempo do cabeçalho (m+n) é comprimido e então enviado ao receptor PTR. O valor do temporizador é 4 (indicando dif_temporizador). Este valor é usado para descomprimir o protocolo de tempo para o cabeçalho (m+n). Então, o exemplo da Figura 6 é bem parecido com o exemplo apresentado na Figura 5, exceto que o temporizador é reajustado depois de receber cada cabeçalho na Figura 6.

[00065] Indiferentemente de qual técnica é usada (Figura 5 ou Figura 6), o esquema de compressão baseado no temporizador efetivo pode ser usado. Porém, se a flutuação acumulativa é excessiva, pode não ser possível regenerar o protocolo de tempo correto baseado no protocolo de tempo comprimido. Em muitos exemplos, a condição a seguir deveria ser satisfeita para k, para permitir que o temporizador baseado no esquema de compressão ilustrado nas Figuras 5 e/ou 6 trabalhe corretamente:

$$[\text{Condição 1}] (\text{Flutuação máxima integral} + 2) < 2^k$$

onde a flutuação máxima integral (FMI) é a flutuação máxima acumulativa, expressa em unidades de T ms, arredondadas para o próximo inteiro mais alto. Por exemplo, se $T = 20$ ms, a flutuação máxima acumulativa de 15 ms resultará em $FMI = 1$. O 2 é adicionado ao FMI para responder pelo possível erro causado pelo assincronismo do temporizador.

[00066] Devido às exigências de tempo-real conversacionais, a flutuação acumulativa em uma operação

normal pode ser apenas de alguns tempos de T ms. Então, neste caso, o valor de k igual a 4 é mais que suficiente, como até 16 amostras de voz (isto é, $16 \cdot T$ ms) a discrepância pode ser corrigida no receptor PTR. Situações anormais ou de erro podem resultar na flutuação, excedendo os valores normais. Uma entidade de redução da flutuação pode ser adicionada ascendentemente do compressor para assegurar que a flutuação, como visto pelo compressor, permaneça dentro de saltos aceitáveis.

[00067] As vantagens do esquema de compressão do protocolo de tempo ilustrado nas Figuras 5 e/ou 6 incluem:

a) O tamanho do protocolo de tempo é constante e pequeno. O cabeçalho comprimido consiste tipicamente de um tipo de mensagem indicando o tipo de mensagem (k_1 bits), a máscara de bit indica qual campo está alterando, e o campo que contém os bits k menos significativos de m_{atual} (bits k). Assumindo que a mesma máscara de bit MTSI de 4 bits é usada como na RFC2508, e $k_1 = 4$, o tamanho do cabeçalho comprimido quando apenas o PT PTR troca (este caso é sem dúvida o mais frequente) é de 1,5 bytes. Além disso, o tamanho não altera como uma função do comprimento do intervalo de silêncio.

b) Como apresentado, por exemplo, na Figura 6, o temporizador do receptor opera na mesma frequência do temporizador da fonte PTR (usado para gerar o protocolo de tempo original); a sincronização de fase entre o temporizador da fonte e o temporizador do receptor não é requerida (porque este é o tempo decorrido quando medido, o temporizador do receptor é o que é usado para regenerar o protocolo de tempo).

c) No receptor, nenhuma sincronização é requerida entre o processo do temporizador e o processo do

descompressor. Por exemplo, o processo do temporizador pode incrementar o temporizador por 1 para todo T ms, enquanto o processo do descompressor está pronto para executar a descompressão quando o cabeçalho novo é recebido. Contudo, não é requerido que o ponto no qual o temporizador incrementa seja alinhado ou sincronizado com o ponto onde o cabeçalho é recebido (veja Figura 6).

d) Robustez a erros, como a informação parcial PT PTR no cabeçalho comprimido está mesmo contida e apenas precisa ser combinada com o temporizador do receptor, para render o valor completo PT PTR. A perda ou a corrupção do cabeçalho não invalidará os cabeçalhos comprimidos subsequentes.

e) Nenhuma memória ou valores precisa ser mantido ou armazenado pelo compressor para o propósito de compressão/descompressão PT PTR.

Handover

[00068] De acordo com uma incorporação, cada IRA_AD é designada para uma área específica (ex., as conexões dos terminais localizados em uma área específica). Os terminais (tal como o terminal 130) podem mover de uma área a outra. Quando os terminais movem de uma área para outra, o terminal deve ser desligado, ou comutado de um IRA_AD para outro IRA_AD.

[00069] Um caso de handover (HO) a considerar é o handover inter_IRA_AD, onde pode haver um rompimento causado pela comutação do antigo IRA_AD para o novo IRA_AD. O assunto em questão é como manter a continuidade da informação pelo handover, de forma que depois do handover, a compressão/descompressão no termo_AD 136 e no IRA_AD novo continuam sem rompimento.

Enlace Descendente

[00070] Não existe nenhuma descontinuidade no lado receptor, o qual é o terminal (ex., terminal 130, Figura 1). A regra do compressor é transferida de um IRA_AD a outro. Depois do handover, os cabeçalhos são direcionados em um novo caminho passando pelo IRA_AD novo em vez do IRA_AD antigo. Em adição, dependendo do projeto do sistema, pode ou não ser re-encaminhado os pacotes de em-trânsito durante o handover. Os pacotes em-trânsito são estes gerados pela fonte, mas que ainda não chegaram no receptor pelo tempo do handover. O re-encaminhamento tenta entregar os pacotes em-trânsito para o terminal.

[00071] Para executar o handover, o IRA_AD antigo tem de transferir o valor inicial do protocolo de tempo pela sessão (PTO) e PT_stride para o IRA_AD novo. Estes dois valores permitem que o novo IRA_AD continue a comprimir os protocolos de tempo novos (nos cabeçalhos de pacote novos) recebidos da fonte PTR (ex., terminal 102). Deixando o cabeçalho_atual ser o cabeçalho real a ser descomprimido pelo termo_AD depois do handover, e o seu PT_atual seu protocolo de tempo PTR. O termo_ADC pode descomprimir o PT_atual, contanto que a condição seguinte seja conhecida:

[Condição 2] (Flutuação integral transiente descendente + 2) < 2^k

onde a flutuação integral transiente descendente (FITD) é a flutuação transiente descendente do cabeçalho_atual, expresso em unidades de T ms, arredondadas para o próximo inteiro mais alto. A flutuação transiente descendente é definida como = retardo total do cabeçalho_atual - tamanho do retardo no caminho antigo. Se

o cabeçalho_atual não for o cabeçalho do pacote em-trânsito direcionado, o retardo total do cabeçalho_atual também é o tamanho do retardo no caminho novo + a flutuação acumulativa para o cabeçalho_atual no caminho novo. Então, a flutuação transiente descendente = tamanho do retardo no caminho novo - o tamanho do retardo no caminho antigo + a flutuação acumulativa para o cabeçalho_atual.

[00072] Se o cabeçalho_atual for o cabeçalho de um pacote em-trânsito direcionado, o retardo total do Cabeçalho_atual = retardo total causado pelo direcionamento e re-direcionamento. Na prática, os sistemas deveriam preferivelmente ser projetados para manter a flutuação transiente descendente na mesma gama de valor como a flutuação acumulativa no estado estável (isto é, sem handover). Então, baseado nestas suposições (que pode nem sempre ser aplicada, nenhum problema relacionado a um handover específico para o descendente é esperado se a condição 1 (acima observada) for conhecida.

Enlace Ascendente

[00073] Nesta descrição do enlace ascendente, o termo_AD 136 do terminal (ex., terminal 130) comprime o protocolo de tempo e envia este sobre o enlace RF 140 para o IRA_AD local ou correspondente. A fonte PTR neste caso é o terminal 130. Mesmo quando a fonte PTR (o terminal 130) altera os locais físicos (requerendo um handover no IRA_AD), a regra do receptor (descompressor) é transferida de um IRA_AD para outro. A fonte PTR permanece ancorada ao terminal (ex., terminal 130, Figura 1).

[00074] A Figura 7 é um diagrama ilustrando um exemplo de operação do handover de acordo com uma incorporação da presente invenção. Para minimizar a sobrecarga da interface de ar, alguma informação necessita ser transferida de um IRA_AD ANTIGO 710 para um IRA_AD novo

para handover. A informação é o valor do temporizador no IRA_AD antigo. O IRA_AD antigo 710 lê (ou toma conhecimento) o valor atual do temporizador (T_u) no IRA_AD antigo DC e envia este para o IRA_AD novo, junto com PT_0 , $PT_{progresso}$ e o m_{atual} , bloco 714 (Figura 7). O IRA_AD novo retoma incrementando o seu temporizador a partir de (T_u). Vamos tomar $T_{transferir}$ 715 (Figura 7) para ser o tempo para transferir o Temporizador. Em adição, o temporizador processa no IRA_AD antigo e o novo IRA_AD pode ter uma diferença de fase que é no máximo de T ms. Deixe o $cabeçalho_{atual}$ ser o primeiro cabeçalho real a ser descomprimido pelo IRA_AD novo depois do handover, e PT_{atual} ser seu protocolo de tempo PTR. O IRA_AD novo pode descomprimir PT_{atual} quando a condição a seguir for conhecida:

$$[\text{Condição 3}] \text{ (Flutuação integral transiente ascendente + 2 + 1) } < 2^k$$

onde a flutuação integral transiente ascendente (FITA) é a flutuação transiente ascendente expressa em unidades de T ms, arredondadas para o próximo inteiro mais alto. A flutuação transiente ascendente é definida como = retardo total do $cabeçalho_{atual}$ - o tamanho do retardo no caminho antigo + o $handover_T$. desde retardo total do $cabeçalho_{atual}$ = o tamanho do retardo no caminho novo + a flutuação acumulativa para o $cabeçalho_{atual}$, a flutuação transiente ascendente = tamanho do retardo no caminho novo - tamanho do retardo no caminho antigo + a flutuação acumulativa para o $cabeçalho_{atual}$ + o $handover_T$. Comparado ao caso do enlace descendente, 1 é adicionado para contar para a diferença de fase entre o temporizador de IRA_AD antigo e o temporizador de IRA_AD novo.

[00075] Especificamente, a Figura 7 também ilustra a flutuação transiente ascendente, a qual inclui a diferença de retardo do tamanho e do handover T . Neste exemplo, o IRA_AD antigo decide preparar o handover antes que o temporizador incremente o temporizador. Então este envia $T_u = 0$ para o IRA_AD novo DC 712. O handover_T é aproximadamente de T ms. No IRA_AD novo 712, devido ao assincronismo do processo do temporizador processe, quase T ms decorrido antes do temporizador ser incrementado. Também existe uma flutuação acumulativa no caminho novo para o cabeçalho $(n+m)$. Como resultado, o valor do temporizador ao ler quando o cabeçalho $(n+m)$ for recebido é 2, enquanto que o valor real deveria ser 4. Assim, há uma distorção de -2 . Contudo que a condição 3 seja conhecida, a distorção pode ser eliminada e o protocolo de tempo PTR pode ser descomprimido corretamente.

[00076] De acordo com uma incorporação, T_u é transmitido em uma rede de sinalização de alta velocidade conectando o IRA_AD antigo e novo. Por conseguinte, o tempo handover_T deveria ser no máximo de apenas alguns poucos T ms. Porém, os casos onde o handover de T_u não tem êxito, ou não oportuno deve ser considerado. Nestes casos, o IRA_AD novo notificará o termo_AD, o qual envia o protocolo de tempo PTR completo até que um reconhecimento seja recebido.

Redução da Flutuação

[00077] De acordo com uma incorporação da presente invenção, o esquema de compressão baseado no temporizador usando o protocolo de tempo comprimido e o temporizador do receptor local pode ser predito nas condições a seguir estabelecidas.

[Condição 1] (Flutuação integral máxima + 2) < 2^k ,

[Condição 2] (Flutuação integral transiente descendente + 2) < 2^k ,

[Condição 3] (Flutuação integral transiente ascendente + 2 + 1) < 2

[00078] Devido às exigências de conversação em tempo-real, pode esperar razoavelmente que as várias flutuações acima sejam na ordem de T ms em uma operação normal. Então, um valor pequeno de k, ex. 4, é normalmente mais do que suficiente para permitir corrigir qualquer distorção ou erro. Porém, em condições anormais, no caminho da fonte PTR para o receptor (falhas, etc.) ou outras situações podem existir, onde as flutuações se tornam excessivas (onde o protocolo de tempo correto não pode ser gerado baseado no protocolo de tempo comprimido e no temporizador do receptor local). Para uma negociação com estes casos, a função de redução de flutuação (FRF) 115 (Figura 1) pode ser fornecido como uma extremidade frontal para o compressor, para filtrar (ou descartar) os pacotes que têm flutuação excessiva (ex., onde quaisquer das condições 1, 2 ou 3 acima não são conhecidas).

[00079] Para visualizar ou identificar os pacotes possuindo MIJ excessivo, a função de redução de flutuação (FRF) calcula a flutuação de cada pacote recebido sobre a rede 108. Se a flutuação do pacote medido for maior que $2^k - 2$, este é considerado uma flutuação excessiva e o pacote é descartado. Caso contrário, o cabeçalho (ou campo do cabeçalho) é comprimido (como descrito acima) e então transmitido para o terminal receptor (ex., terminal 130).

[00080] O FRF calcula a flutuação do pacote atual como a seguir: a flutuação = valor absoluto de (PT2 - PT1 - dif_temporizador FRF), onde PT2 é o protocolo de tempo do pacote atual, PT1 é o protocolo de tempo do pacote

prévio, e dif_temporizador de FRF é a diferença no temporizador de FRF entre o pacote atual e o pacote prévio (tempo decorrido). Este valor de flutuação é comparado ao valor de $2^k - 2$. Se a flutuação for maior que $2^k - 2$, o pacote é descartado. Caso contrário, o cabeçalho do pacote é comprimido no IRA_AD e o pacote com o cabeçalho comprimido é enviado ao receptor PTR.

[00081] Esta função de redução de flutuação (FRF) 115 é uma técnica efetiva para limitar a flutuação nos pacotes recebidos pelo terminal receptor (porque a flutuação introduzida sobre o enlace RF pode ser considerada desprezível). Além disso, o FRF opera para usar a largura de banda disponível mais eficazmente sobre o enlace de RF 140. Na ausência do FRF 115, um ou mais pacotes que têm uma flutuação maior do que $2^k - 2$ poderiam ser transmitidos ao receptor PTR sob o enlace 140. Porém, no receptor, se a flutuação for excessiva (isto é, condição 1 não é encontrada), o valor de protocolo de tempo correto não pode ser gerado, fazendo com que o receptor descarte o pacote. Assim, a FRF opera meramente para filtrar estes pacotes que têm flutuação excessiva que seriam descartados de qualquer maneira no receptor (evitando desperdício de uma largura de banda valiosa sobre o enlace 140).

II - Esquema de Retirar o Cabeçalho

[00082] A segunda incorporação da presente invenção provê um esquema de retirada do cabeçalho baseado no temporizador, no qual o cabeçalho ou um ou mais campos do cabeçalho (ex., incluindo o protocolo de tempo) é retirado do pacote PTR antes de transmitir através do enlace de largura de banda baixa (ex., através do enlace de RF 140, Figura 1). Neste caso, o protocolo de tempo não é explicitamente fornecido no pacote PTR. Preferivelmente, a informação de temporização pode ser implicitamente

fornecida para o regenerador do cabeçalho para incrementar o temporizador local baseado essencialmente em um canal de taxa de bit constante ou em uma conexão a circuito entre o extrator do cabeçalho (ex., o qual pode existir no ARI_AD) e o regenerador do cabeçalho (ex., o qual pode existir no terminal 130).

Avaliação de Extração do Cabeçalho

[00083] A extração do cabeçalho é baseada na idéia de que para algumas aplicações ou serviços, não é necessário transportar toda a informação contida nos cabeçalhos IP/PDU/PTR, ou porque eles não mudam, ou porque eles não são essenciais à aplicação/serviço. A voz é um exemplo típico básico. Para prover um serviço equivalente ao serviço de voz celular existente (ex., sobre o enlace RP 140, Figura 1), a única informação do cabeçalho variável que é essencial é o protocolo de tempo PTR (PT). Também é desejável manter a transparência para o número sequencial PTR (NS). A transparência aqui (para o NS) significa que o NS depois da extração ou regeneração é igual ao NS original. A extração do cabeçalho confia na informação de temporização implícita provida por uma conexão a circuito ou essencialmente a um canal de taxa de bit constante (onde nenhuma flutuação de pacote é introduzida) para permitir que o protocolo de tempo PTR seja regenerado baseado apenas no temporizador local ou contador. Isto elimina a necessidade para enviar o protocolo de tempo explicitamente (ou até mesmo enviar o protocolo de tempo comprimido). Para alcançar a transparência do NS, os NSs comprimidos podem ser usados em combinação com a informação de temporização do canal ou conexão a circuito. Uma conexão a circuito preferivelmente fornece um canal possuindo uma taxa de bit essencialmente constante. Quando não existe nenhuma amostra de voz (ex., intervalo de silêncio), o canal pode ou não

ser alocado a outros tráfegos e/ou usuários. As vantagens deste esquema de extração de cabeçalho incluem:

a) sobrecarga do cabeçalho inferior, incomparável por qualquer outro esquema (menos do que a técnica do cabeçalho comprimido descrito acima nas Figuras 1 - 6);

b) robustez a erros, uma vez que a informação de temporização da transmissão a circuito ou do canal de taxa de bits essencialmente constante é inerentemente não afetada por erros;

c) possibilidade para comutar durante uma chamada para a compressão do cabeçalho (ex., a técnica das Figuras 1 - 6), se assim desejado. Isto pode ser útil se a chamada se torna de multimídia, quando o meio de não-voz é adicionado para voz. Além disso, note que a extração do cabeçalho não designa ou impede uma multiplexação estatística, se implementado, poderia ocorrer na camada inferior.

[00084] A Figura 8 é um diagrama em blocos ilustrando o exemplo de uma pilha de acordo com uma incorporação da presente invenção. A pilha de extração do cabeçalho 802 e a pilha do regenerador de cabeçalho 830 são apresentadas. Como um exemplo, a pilha de extração do cabeçalho 802 ilustra alguns dos componentes que podem ser usados para retirar um ou mais campos do cabeçalho do pacote, enquanto a pilha do regenerador de cabeçalho 830 ilustra alguns dos componentes que podem ser usados para regenerar o cabeçalho do pacote. A pilha de extração do cabeçalho 802 poderia ser provida, por exemplo, dentro de um tipo de adaptador IRA_AD (ex., IRA_AD 112, Figura 1), enquanto a pilha do regenerador do cabeçalho 830 pode residir, por exemplo, em um tipo de adaptador terminal (ex., termo_AD 136, Figura 1).

[00085] Recorrendo à Figura 8, a pilha de extração do cabeçalho 802 inclui as camadas PTR e PDU 804, e uma camada IP 806. As camadas PTR/PDU/IP geram o pacote PTR 808 (que inclui um protocolo de tempo no cabeçalho PTR). A seguir, na pilha de extração do cabeçalho 802, o pacote PTR 808 é fornecido ao extrator de cabeçalho (EC) 810 para tirar ou remover um ou mais cabeçalhos ou campos do cabeçalho. As camadas C1 e C2 812 são fornecidas, onde C2 pode ser uma camada de enlace de dados e a camada C1 pode ser uma camada física, por exemplo. Outras camadas poderiam ser providas se necessário. Similarmente, a pilha do regenerador do cabeçalho 830 inclui as camadas C1 e C2 correspondentes 820, o regenerador do cabeçalho (RC) 822 que regenera o cabeçalho (incluindo o protocolo de tempo PTR) para prover o pacote PTR completo 824 (incluindo o cabeçalho RTI/PDU/IP). O pacote 824 é fornecido para a camada IP 826 e então para as camadas PDU e PTR 828. As camadas C1 e C2 da pilha de extração do cabeçalho 802 e da pilha do regenerador do cabeçalho 830 estão em comunicação do outro lado do enlace 815 ou da interface aérea (como o enlace RF 140) ou pela rede. Por exemplo, a voz sobre os pacotes IP é passada pelo extrator do cabeçalho 810 antes da transmissão sobre o enlace 815 (ex., enlace sem fio ou rede). Na extremidade de recepção (a pilha do regenerador de cabeçalho 830), o regenerador de cabeçalho 822 regenera os cabeçalhos antes de entregar para o recipiente. As camadas C2/C1 podem prover uma conexão a circuito, isto é, provendo um canal de taxa de bits essencialmente constante entre o extrator do cabeçalho 810 e o regenerador de cabeçalho 822. Além disso, para uma máxima eficiência, a camada C1 também pode executar a otimização de carga útil de voz como uma proteção de bit desigual, além da codificação e da intercalação de canal aperfeiçoada. Note

que o conceito de extração de cabeçalho é aplicável indiferente se a otimização de carga útil é feita ou não.

[00086] Na operação, o extrator de cabeçalho (EC) 810 elimina a flutuação nos pacotes PTR entrantes, e os joga atrás de acordo com o protocolo de tempo (PT) PTR no cabeçalho. Aqui, ao eliminar a flutuação significa programar a transmissão da amostra de voz na conexão a circuito ou em um canal de taxa de bits essencialmente constante conforme o protocolo de tempo. Em outras palavras, os pacotes, depois da remoção ou retirada dos cabeçalhos, são transmitidos no canal a circuito ou canal de taxa de bit essencialmente constante às vezes baseado nos seus protocolos de tempo no pacote. Os pacotes com flutuação excessiva são descartados, usando a função de redução da flutuação (FRF 115, Figura 1), por exemplo. O regenerador de cabeçalho (RC) 822 reconstrói os campos IP/PDU/PTR que podem ser classificados nas categorias a seguir:

a) estático: O valor não muda para a duração da sessão, ex. endereços IP;

b) não estático: O valor poderia mudar a princípio de um pacote para o próximo, mas na prática, para voz, apenas o campo não-estático é essencial para preservar através da extração do cabeçalho é o protocolo de tempo PTR (PT). O número sequencial (NS) PTR também é preservado. Os campos estáticos podem ser transferidos de uma vez por todas como parte de um cabeçalho cheio na fase de inicialização, ao começo da sessão. Um mecanismo de entrega seguro pode ser usado (ex., usando Reconhecimentos ou Recs do receptor PTR para acusar o recebimento da informação de inicialização). O protocolo de tempo e o número sequencial serão discutidos brevemente.

Protocolo de tempo (PT) PTR

[00087] No caso de voz, o protocolo de tempo PTR (PT) aumenta linearmente como uma função de relógio de parede (isto é, temporizador de fonte) à fonte PTR. Se o intervalo de tempo entre as amostras de voz sucessivas é T ms, então o protocolo de tempo PTR do cabeçalho n (gerado no tempo n*T ms) = protocolo de tempo PTR do cabeçalho 0 (gerado no tempo 0) + PT_progresso*n, onde PT_progresso e T são constante dependentes do codec de voz. Isto é verdade se houver um pacote por amostra de fala (voz). Mais geralmente, o protocolo de tempo PTR (PT) é da forma $PT_0 + m * PT_progresso$, onde PT_0 é $< PT_progresso$ e m é um inteiro. O mesmo comportamento é visto no extrator de cabeçalho (EC) depois da flutuação ter sido eliminada.

[00088] No começo de uma sessão ou conexão, a fase de inicialização é executada para inicializar o receptor PTR (isto é, inicializar o regenerador de cabeçalho). Na fase de inicialização, o extrator de cabeçalho continua enviando a informação de inicialização (Ini_info) até que um Rec seja recebido do receptor. Ini_info(n) consiste essencialmente no cabeçalho n IP/PDU/PTR completo (incluindo o protocolo de tempo inicial e o número sequencial). O número sequencial PTR é usado para identificar este cabeçalho de inicialização particular, desde que os cabeçalhos de inicialização subsequentes incluirão o número sequencial maior (assumindo que o primeiro cabeçalho de inicialização não é reconhecido).

[00089] No regenerador de cabeçalho (RC) 822, quando a Ini_info(n) é recebida corretamente, o RC 822 envia um rec(n). Uma vez que o regenerador de cabeçalho (RC) 822 tenha recebido todo o cabeçalho, o EC 810 pára de enviar os cabeçalhos completos. O RC 822 também inicia o contador do protocolo de tempo local, que é inicializado no

protocolo de tempo PTR recebido na Ini_info(n). O contador PT é similar ao temporizador do receptor na Figura 1, mas o contador PT é incrementado por PT_progresso a todo T ms (preferível por 1, mas é o mesmo princípio do temporizador do receptor). Para os quadros de voz extraídos subsequentes (isto é, os pacotes PTR onde os cabeçalhos foram retirados ou removidos), o PTR é regenerado do contador do protocolo de tempo (PT). O temporizador do receptor (temporizador PT) tem a mesma frequência do relógio ou do temporizador usado na fonte PTR (isto é, o temporizador da fonte) para gerar o protocolo de tempo. Além disso, a conexão a circuito provê uma taxa de bits essencialmente constante, e assim, os retardos do pacote não são variáveis ou não mudam de pacote para pacote. Como resultado, não há nenhuma flutuação de pacote devido ao canal de taxa de bits essencialmente constante. Então, depois que o receptor PTR recebe a informação de inicialização, incluindo o valor do protocolo de tempo inicial (PT0), o receptor PTR pode regenerar um protocolo de tempo correto para cada pacote subsequente (depois da inicialização) apenas baseado no contador do protocolo de tempo (ou temporizador do receptor).

[00090] O canal de taxa de bits essencialmente constante fornecido entre o extrator de cabeçalho 810 e o regenerador de cabeçalho 822 necessita apenas prover um número predeterminado de bits sobre um período predeterminado de tempo entre o extrator de cabeçalho 810 e o regenerador de cabeçalho 822, mas esta função pode ser executada de uma variedade de modos diferentes. Por exemplo, o canal pode ser um canal de taxa de bits constante que é dedicado ao extrator 810 e ao regenerador 822 ou compartilhado entre vários usuários. O canal pode prover, por exemplo, um bit a todo milissegundo, ou provê

100 bits a todo 100 milissegundos, mas onde a taxa de dados pode não ser constante (isto é, pode variar) dentro de um período de 100 ms. Como um exemplo adicional, o canal pode prover um número predeterminado de bits para uma ou mais rajadas de dados entre o extrator de cabeçalho e o regenerador de cabeçalho. Por exemplo, o canal pode prover um estouro ou uma rajada de 1000 bits a todo 10 milissegundos. Assim, o canal de taxa de bits essencialmente constante necessita apenas prover um número predeterminado de bits sobre um período predeterminado de tempo, mas pode realizar isto usando técnicas diferentes.

Número Sequencial (NS) PTR

[00091] O NS PTR (como visto pelo EC 810) normalmente aumenta por 1 de um pacote para o próximo. As únicas exceções são quando os pacotes são perdidos ou desordenados. No enlace ascendente, a perda ou a desordem do pacote não é esperada que aconteça, desde que o extrator de cabeçalho (EC) 810 e a fonte PTR estejam muito perto um do outro. Então, o seguinte se aplica ao descendente. O EC 810 armazena de forma limitada para tentar reordenar pacotes antes de tirar os seus cabeçalhos. O pacote com NS PTR n é considerado perdido se ainda não tiver sido recebido até que o pacote com NS PTR $(n+1)$ tiver seu cabeçalho retirado. O pacote com NS PTR m é desordenado se, pelo tempo este for recebido, o pacote com NS PTR k teve seu cabeçalho tirado, e $k > m$. O comprimento da memória de reordenação é um parâmetro de projeto. Quanto maior a memória resultará em um retardo excessivamente grande, enquanto que uma memória menor resultará em muitos pacotes descartados. O parâmetro também depende da qualidade provida pela rede IP ascendente 108 do EC 810. O RC 822 mantém o contador NS que é sua melhor estimativa de NS. Observando a `Ini_info`, o RC 822 pode obter o NS inicial e o

número de bits também contidos em um pacote conhecido como o tamanho do pacote (tamanho). O RC 822 inicializa o contador NS com o NS da Ini_info. O RC 822 então "conta" que os bits de voz recebidos sobre o canal de taxa de bits essencialmente constante e incrementa o contador NS 1 para todo p_tamanho bits de voz (não é incrementado quando nenhum pacote é recebido, ex. durante um intervalo de silêncio). De acordo com uma incorporação, o RC 822 não conta os bits recebidos de fato. Preferivelmente, o contador NS no RC 822 é incrementado por 1 durante todo o pacote, onde a duração de pacote é o tempo exigido para receber um pacote de bits (p_bits). Assim, a duração do pacote será uma função do tamanho do pacote (p_tamanho) e a taxa de bits (que é constante sobre a conexão a circuito).

[00092] Assim, pode ser visto que depois que a inicialização ocorre (provendo o NS inicial e PT ao RC 822), o RC 822 pode gerar o protocolo de tempo para os pacotes subsequentes incrementando o contador PT por PT_progresso para todo T ms, e incrementando o contador NS por 1 por toda a duração do pacote. Então, depois da inicialização, estes campos podem ser regenerados no RC 822 com referência apenas ao relógio local (assumindo que PT_progresso e a duração do pacote são conhecidos por RC 822). Ao incrementar o contador NS baseado no tempo (duração do pacote) em lugar da contagem atual dos bits recebidos é mais robusto a erros. No evento, um ou mais bits são derrubados antes de alcançar o RC 822, o contador NS refletirá o verdadeiro valor e não será afetado pelos bits perdidos.

Descontinuidade e Cadeia

[00093] A descrição anterior indica que PT e NS podem ser completamente retirados por EC 810 antes da transmissão do outro lado do enlace (ex., enlace RF 140), e

então regenerado pelo RC 822 que mantém um relógio local ou temporizador (ex., incrementando o contador PT por $PT_progresso$ a todo T ms e incrementando o contador NS por 1 por toda a duração do pacote). Porém, um ou mais eventos de descontinuidade básicos podem ocorrer, os quais se não endereçados, poderia provavelmente invalidar a aproximação de regeneração baseada no temporizador como descrita acima. Alguns dos eventos de descontinuidade podem incluir:

a) Evento "Novo Jato": Mudança passageira na diferença PT entre o pacote n e $(n+1)$ (começo de um novo jato de conversa); isto também pode ser descrito como uma mudança não-linear ou uma troca no protocolo de tempo (PT);

b) Evento "Mudança de tamanho": Mudança no tamanho do pacote PTR ($p_tamanho$), causado por uma mudança no número de quadros de voz empacotados em um e/ou o tamanho do quadro de voz;

c) Evento "Mudança progresso": Mudança no $PT_progresso$ (causado, ex., por uma mudança no tipo de carga útil PT).

[00094] Definimos uma cadeia do cabeçalho como uma sucessão de cabeçalhos de pacote, tal que todos os pacotes possuem o mesmo tamanho ($p_tamanho$), os números sequenciais são sucessivos, isto é n , $(n+1)$, $(n+2)$, etc., e os protocolos de tempo (PTs) dos pacotes sucessivos são espaçados pelo mesmo incremento $PT_progresso$. Em outras palavras, pode ser considerado que uma cadeia do cabeçalho é uma cadeia de cabeçalho que têm alguns campos de pacote em comum (ex., o tamanho do pacote), e outros campos que aumentam linearmente através dos pacotes sucessivos, tal como NS e PT. A cadeia é normalmente um jato de conversa (ex., uma série de amostras de voz fornecidas entre os intervalos de silêncio).

[00095] A transição de uma cadeia para outra pode ser causada por quaisquer dos eventos de descontinuidade, isoladamente ou até mesmo em combinação. Neste esquema, quando uma cadeia começa (e a cadeia prévia termina), o EC 810 determina qual evento de descontinuidade aconteceu, e adequadamente envia a inicialização da informação de cadeia necessária (cadeia_ini) para o RC 822.

[00096] A Figura 9 é uma tabela que ilustra a informação que pode ser provida nas mensagens de acordo com uma incorporação da invenção. Ini_info inclui tipicamente um cabeçalho completo (incluindo NS e PT completo), e é enviado do EC 810 para o RC 822 (inicializar RC 822) no começo da sessão. EC 810 continuará a reenviar a ini_info até receber um rec do RC 822, antes de proceder ao envio dos pacotes de dados sem o cabeçalho. Depois disso, pode haver uma ou mais cadeias as quais podem acontecer e que podem requerer atualizações adicionais dos campos ou valores que mudam de uma cadeia para outra. Estes valores alterados são fornecidos ao RC 822 usando cadeia_ini.

[00097] Cadeia_ini inclui o valor p_tamanho (se este tem trocado a cadeia prévia), e o valor PT_progresso (se este tem trocado a cadeia prévia). Se nenhuma troca não-linear ocorrer ao PT de uma cadeia para a próxima, o RC 822 pode continuar a regenerar o PT baseado no contador PT usado na cadeia antiga. Porém, se uma troca não-linear no protocolo de tempo (PT) acontece entre as cadeias (isto é, perda de temporização), o protocolo de tempo atualizado deve ser enviado explicitamente na cadeia_ini do EC 810 para o RC 822. O PT atualizado pode ser enviado como um protocolo de tempo comprimido 414 (veja Figura 4) descrito acima tao longo como a condição 1 é conhecida, como descrito acima. Caso contrário, se a condição 1 não é

conhecida, o protocolo de tempo atualizado completo deve ser transmitido ao RC 822.

[00098] No modo rec., depois que o EC 810 envia a cadeia_ini ao RC 822, o EC 810 pode requerer o RC 822 para reconhecer (ou rec) o recebimento da informação da cadeia atualizada (cadeia_ini) antes do EC 810 enviar os pacotes de dados adicionais (pacotes sem cabeçalho) para o RC 822. No modo reconhecido ou modo rec., o EC 810 envia repetidamente uma mensagem de cadeia_ini ao RC 822 até que o EC 810 receba um rec do RC 822 para uma mensagem de cadeia_ini. Depois de receber um rec do RC 822, o EC 810 enviará então os pacotes restantes da cadeia como pacotes de cabeçalho retirado (desde que o PT e o NS para os pacotes da cadeia nova possam ser regenerados usando apenas o relógio local ou temporizador). A exigência de rec (no modo reconhecido) para a mensagem da cadeia_ini, o EC 810 previne de enviar uma nova cadeia sem notificar o RC 822. Por exemplo, se o EC 810 envia uma mensagem de cadeia_ini nova (ex., provendo os campos atualizados ou as informações relacionadas a um evento de descontinuidade) o enlace entre o EC 810 e RC 822 está temporariamente rompido, o EC 810 não pode proceder ao envio dos pacotes de cabeçalhos retirados até receber o primeiro rec do RC 822.

[00099] Uma vez que o EC 810 está confiante que o RC 822 recebeu a informação de cadeia_ini, os quadros de voz (ex., os pacotes de dados) são enviados então sem o cabeçalho para o restante da cadeia. Para estes quadros sem cabeçalhos, o PT e o NS são regenerados usando o relógio local no RC 822.

[000100] O EC 810 pode determinar os eventos como a seguir:

a) Evento "Novo jato": a diferença PT entre o pacote que tem NS = n e o pacote que têm = (n+1) é

diferente de PT_progresso. Isto significa o começo de uma cadeia nova ou jato de conversa. Para assegurar a sincronização de NS, a cadeia_ini consiste em um NS ou NS comprimido neste caso, (C_NS). Se não houver informações de NS enviada, o RC 822 não pode estar seguro que incrementando o contador NS por 1 por toda a duração do pacote dará para um NS preciso. Isto é porque poderia ter havido uma desconexão de enlace durante a qual os bits de voz estariam perdidos entre o EC 810 e RC;

b) Evento "Mudança de tamanho": o tamanho do pacote PTR que tem NS = n é diferente do pacote previamente recebido; isto afetará o valor de duração do pacote (a taxa à qual o contador NS é incrementado). Cadeia_ini inclui valor de p_tamanho novo;

c) Evento "Mudança progressiva": Determinado da análise do campo (PT) do tipo de carga útil no pacote PTR, a cadeia_ini inclui novo valor PT_progresso.

[000101] Estes eventos de descontinuidade são fornecidos apenas como exemplos. Outros tipos de eventos de descontinuidade são possíveis.

[000102] Eventos podem acontecer em combinação (evento combinação). Neste caso, a cadeia_ini inclui toda a informação dos eventos básicos correspondentes. Por exemplo, se "Novo jato" acontece na combinação com "Mudança de tamanho", cadeia_ini = {C_NS, valor p_tamanho novo}.

Procedimento para enviar Ini info, Cadeia ini

[000103] Ini_info é normalmente enviado no modo de rec, desse modo, o EC 810 enviará a Ini_info até o reconhecimento pelo RC 822. Cadeia_ini pode ser enviada no rec ou modo de não rec. O EC 810 enviará a cadeia_ini a todo pacote até ser reconhecido pelo RC 822 no modo de rec. Uma vez que o rec é recebido, o EC 810 envia apenas os bits de voz para o restante da cadeia, sem qualquer cabeçalho.

No modo de não rec, o EC 810 enviará a cadeia_ini a um certo (predeterminado) número de tempos antes de enviar os bits de voz apenas para o restante da cadeia. Opcionalmente, a cadeia_ini pode ser repetida a algum intervalo durante a cadeia para assegurar que o RC 822 seja sincronizado (ex., tenha os próprios valores).

[000104] Um evento de combinação que inclui o evento básico de "Mudança do tamanho" ou "Mudança progressiva" tipicamente exigirá que a cadeia_ini seja enviada no modo de rec. Neste, a cadeia_ini init levará um número de geração. O número de geração é um contador incrementado sempre que p_tamanho ou PT_progresso altera. É usado no caso onde p_tamanho ou PT_progresso altera em uma sucessão rápida, para manter o rasto de qual mudança foi reconhecida pelo RF 822. Por exemplo, se p_tamanho mudar do valor p_tamanho_0 para p_tamanho_1, então novamente o valor de p_tamanho_2, o EC 810 enviará para cadeia_ini que contém p_tamanho_1, com número de geração 3, então subsequentemente outra cadeia_ini que contém p_tamanho_2, com número de geração 4. O recebimento de um rec subsequente à segunda cadeia_ini será ambíguo, se não transportar o número de geração da cadeia_ini que é reconhecida. Se o evento de combinação é apenas "Novo jato", a cadeia_ini (C_NS) pode ser enviada em um modo rec ou não rec. O modo não rec é baseado na idéia de que C_NS será repetido pelo menos ao começo de todo jato de conversa. Então a probabilidade de que o RC 822 nunca irá re-sincronizar o seu NS é assintomaticamente pequeno. Além, se o NS for dessincronizado, este é apenas causado através da perda de pacote entre o EC 810 e RC 822. Então, o efeito da dessincronização de NS é regenerada NS < NS correto. Esta é apenas uma inconsistência passageira que é corrigida tendo o NS aumentando a diferença assim que o próximo C_NS

seja recebido. Um aumento de NS por mais de 1 será interpretado pelo ponto final PTR de recepção como uma perda de pacote(es), e normalmente não deveria afetar o retorno do próprio pacote recebido. O modo não rec também permite dispensar com um canal para levar recs em um estado estável, isto é depois de estabelecer a chamada e entre os handovers.

Handover

[000105] Quando a extração/regeneração do cabeçalho é aplicada nos sistemas celulares ou outros sistemas onde os terminais podem mover de um adaptador de rede (IRA_AD) a outro, os handovers deveriam ser considerados.

[000106] O handover pode ser modelado como passando por três fases: preparação do handover, execução do handover e conclusão do handover. Há uma função chamada gerente de handover (TR) (que pode ser fornecido no IRA 110) que decide começar a preparação do handover. Tradicionalmente, a preparação do handover consiste em sinalizar as trocas de mensagens com o sistema alvo para reservar os recursos no sistema alvo e obter a informação necessária sobre a célula alvo. A execução de transferência é iniciada pelo gerente de fonte TR que envia um comando TR ao terminal receptor (ou estação móvel), junto com a informação sobre a célula alvo. Em resposta ao comando TR, o terminal (ou estação móvel) executa o handover. O handover completo envolve a troca de sinalização entre a estação terminal ou móvel e o sistema alvo, a notificação para a fonte, e a liberação dos recursos não necessários (ex., à fonte).

Enlace Ascendente

[000107] A IRA_AD atua como um RC 822 para a transmissão ascendente de dados (veja ascendente 142,

Figura 1). A IRA_AD alvo tem que ser fornecida com a informação necessária para regenerar todo o cabeçalho. As restrições principais incluem a continuidade de PT PTR e NS PTR através do handover (TR).

[000108] A Figura 10 é um diagrama que ilustra um processo de handover de acordo com uma incorporação da presente invenção. O terminal 130 (ou estação móvel EM), como um exemplo, pode notificar a fonte IRA_AD 112 que o tamanho do pacote mudou, usando uma mensagem de cadeia_ini, passo 902. A fonte IRA_AD 112 reconhece esta atualização para o p_tamanho, passo 904. Subsequentemente, o terminal 130 move para uma área nova coberta pelo IRA_AD 114 alvo, e o gerente TR 901 notifica a fonte IRA_AD da preparação para handover, passo 906. A fonte IRA_AD envia então uma informação de inicialização_HO (HO_ini_u) para o IRA_AD 114 alvo, passo 908. HO_ini_u é uma vista calculada do cabeçalho IP/PDU/PTR cheio. A vista calculada consiste no último cabeçalho regenerado, mas com PT PTR substituído por PT0_u, m_último_u, PT_progresso_u, e o valor do Temporizador_u PT. Estes valores são relacionados ao PT_último, o PT PTR do ultimo cabeçalho regenerado como a seguir: $PT_{\text{último}} = PT0_u + m_{\text{último}_u} * PT_{\text{progresso}_u}$. Temporizador_u PT é um contador na fonte IRA_AD que foi incrementado por 1 para todo T ms. Além disso, HO_ini_u inclui p_tamanho_u (tamanho atual do pacote na direção ascendente). De HO_ini_u, IRA_AD alvo deriva os campos estáticos, como também os valores iniciais aproximados para os campos variáveis (PT PTR e NS PTR). O comando de handover é enviado do gerente TR 901 para o terminal 130 (estação móvel), passo 910, fazendo com que o terminal 130 comute e agora use a IRA_AD alvo para comunicação. Porém, o gerente TR pode não ser necessário assim como outras técnicas podem ser usadas para iniciar o handover.

[000109] O handover é considerado como rompendo qualquer cadeia entrante. Então, depois da conclusão do handover, a primeira amostra de voz a ser enviada sempre é controlada como uma nova cadeia, que requer o envio da informação de inicialização (HO_sinc_u), passo 912. Existem três pontos significativos no tempo: PT1 é o começo da preparação TR, PT2 é o recebimento da EM do comando TR e PT3 é o tempo que a fonte IRA_AD levou no instante de sua informação interna ser enviada na HO_ini_u. Deixe H0T para ser o tempo decorrido de PT1 a PT2. Do projeto do sistema, existe um limite superior em H0T: $H0T < H0T_{max}$. Um quarto ponto significativo no tempo é PT4: a primeira vez quando o terminal 130 quer retomar o envio de voz no sistema alvo depois da TR. No PT 4, o terminal 130 (EM) determina se a mudança mais recente em p_tamanho_u tem sido reconhecida por PT2_H0T_max. Nesse caso, o terminal 130 está certo de que HO_ini_u contém o valor atualizado de p_tamanho_u. então, não existe necessidade para incluir isto em HO_sinc_u. isto é porque os pontos no tempo são ordenados como $PT1 < PT3 < PT2$. Caso contrário, o terminal 130 (EM) incluirá o valor novo de p_tamanho_u em HO_sinc_u. O mesmo algoritmo se aplica a PT_progresso_u.

[000110] Em todos os casos, HO_sinc_u inclui C_NS. C_NS é necessário porque havia uma ruptura causada pela TR. C_PT é necessário se a taxa de bit, as durações do pacote, etc., na fonte e nos sistemas alvo não forem sincronizados. É provável que este seja o caso. HO_sinc_u é enviado preferivelmente no modo rec.

[000111] HO_ini_u e TR_sin_u são usados pelo IRA_AD 114 alvo para regenerar o cabeçalho cheio como a seguir. Todos os campos exceto PT e NS são copiados de HO_ini_u. NS é obtido descomprimindo C_NS em HO_sinc_u. PT é determinado descomprimindo C_PT em HO_sinc_u.

Enlace Descendente

[000112] A regra de EC é transferida de um IRA_AD para outro. Depois de transferir, os cabeçalhos são direcionados em um caminho novo que passa pelo IRA_AD novo em vez do IRA_AD antigo. Como resultado, poderia haver uma descontinuidade na temporização para regeneração PTR PT no terminal 130 (EM).

[000113] Para controlar o handover para o enlace ascendente, quando o gerente TR decide começar a preparação do handover, este notificará a fonte IRA_AD. A fonte IRA_AD envia então uma informação de inicialização_TR (HO_ini_d) para o IRA_AD alvo. HO_ini_d consiste de p_tamanho_d e de PT_progresso_d, os quais são os valores reconhecidos por último pela EM, junto com o número de geração deles. A primeira vez quando IRA_AD alvo quiser enviar voz depois de TR, IRA_AD alvo tem que enviar HO_sinc_d. HO_sinc_d consiste em C_PT e C_NS. Se o p_tamanho novo difere de p_tamanho_d, HO_sinc_d também contém o valor novo de p_tamanho. Se não, HO_sinc_d apenas contém o número de geração p_tamanho_d. A EM usa o número de geração para recuperar o p_tamanho correto. Este assume que a EM permanece na memória, dos últimos valores de p_tamanho, junto com o número de geração deles. O mesmo algoritmo se aplica para PT_progresso. HO_ini_d é enviada até que seja reconhecida pela EM_AD. HO_sinc_d é enviada no modo rec. O processo de handover é descrito na Figura 2. O caso apresentado é: a mudança mais recente em p_tamanho_u foi reconhecida pelo tempo PT2_HOT_max.

Enviando Mensagens

[000114] Cada informação anterior pode ser enviada na-banda ou fora-da-banda. Na aproximação de na-faixa, a informação é enviada no canal de voz roubando os bits menos significativos de voz. Na aproximação de fora-

da-banda, um canal passageiro dedicado é estabelecido e desfeito quando um rec é recebido. Uma combinação de na-banda e fora-da-banda é possível, desse modo, uma aproximação de fora-da-banda é tentada, mas a aproximação de na-banda é uma solução de retorno de falha se não houver nenhum recurso para o canal passageiro. Os reconhecimentos na banda, ou fora-da-banda podem ser enviados no próprio canal de rec dedicado deles, ou fora-da-banda de carona nos outros canais transientes dedicados (CIP, etc).

Na-banda

[000115] Indiferente de como o canal de voz a circuito é realizado, este pode ser modelado como um canal que pode transmitir bits B para todo T ms. Se S é o tamanho do quadro de voz em bits, $S < B$. É esperado com os codecs de voz que, a ini_info seja maior que S. Então uma ini_info não pode ser enviada no espaço de um único quadro de voz. Porém, existe um fator $R \geq 1$ tal como $(R-1)*S < H < R*S$. A ini_info(n) pode ser transportada em um canal a circuito, dividindo-os em uma parte maior de bits B e enviando a parte maior para todo T ms. Um cabeçalho cheio consumirá o espaço de R amostras de voz sucessivas. A Figura 11 é um diagrama ilustrando uma inicialização na-banda de acordo com uma incorporação da invenção. Se houver atividade de voz contínua, as ini_info enviadas são ini_info(0), ini_info(R), ini_info(2R), etc. até que um rec(n) seja recebido. Na Figura 11, estas mensagens ini_info são apresentadas como ini-info 500 e ini_info 502. O extrator de cabeçalho rec ini_info 500, mas não antes do EC 810 enviar um segundo pacote ini_info 502. O próximo pacote 504 é enviado do EC 810 para o RC 822 como uma carga útil do pacote 504 (sem cabeçalho). O RC 822 então regenera o NS e PT e os outros campos do cabeçalho.

[000116] ini_info(0) ocorre das amostras de voz 0, 1, ..., (R-1), ini_info(R) ocorre das amostras de voz R, (R+1), ..., (2R-1), e assim por diante. Se houver atividade de voz descontínua, o cabeçalho 0 é seguido por um intervalo de silêncio $L \cdot T$ ms, então ini_info(0) é repetido. A outra informação (cadeia_ini, HO_sinc_d, HO_sinc_u, Rec), todos têm um tamanho bem abaixo de S, assim eles ajustam no espaço do quadro de voz. Eles roubam os bits de voz menos significativos. Para simplicidade, a análise não leva em conta a expansão causada pela codificação de canal, mas o conceito é válido com ou sem a codificação de canal. O processo de inicialização para o caso na-banda é apresentado na Figura 3.

Fora-da-banda

[000117] A Figura 12 é um diagrama que ilustra uma inicialização fora-da-banda de acordo com uma incorporação da invenção. Um canal separado é estabelecido com a largura de banda apropriada para transportar apenas a ini_info concorrentemente com a voz, que é levada em um canal de voz na aproximação fora-da-banda. O canal separado é chamado de canal de inicialização passageiro (CIP). O sistema pode tentar alocar bastante largura de banda para o CIP para permitir o envio de todo o cabeçalho uma vez para todo T ms. O CIP é projetado para ter uma relação de temporização fixa com o canal de voz.

[000118] Os reconhecimentos fora-da-banda podem ser enviados alocando um canal de reconhecimento passageiro (CRP), ou enviados fora-da-banda, mas de carona em um canal passageiro dianteiro. HO_sinc_u pode ser enviado fora-da-banda sobre o canal de sincronização de handover ascendente passageiro (CSHAP). CSHAP é rompido quando HO_sinc_u é reconhecido. O mesmo se aplica para HO_sinc_d usando um

canal de sincronização de handover descendente passageiro (CSHDP).

Casos de falha

[000119] Pode haver casos onde IRA_AD alvo não terá HO_ini até que a execução de handover tenha acabado. As razões incluem o retardo excessivo na sinalização da rede entre os dois IRA_AD, que necessitam executar o handover depressa, etc. Nestes casos, a rede enviará uma notificação para a EM que então reinicia o processo de inicialização como no começo da chamada.

Caso comum onde p_tamanho e PT_progresso são constantes.

[000120] O caso onde p_tamanho e PT_progresso são constantes é sem dúvida o mais comum para voz. Neste caso, nenhuma das considerações causadas por uma possível mudança de p_tamanho e PT_progresso se aplica. O esquema genérico é simplificado. HO_ini_d não é necessário. HO_sinc_d e HO_sinc_u apenas transportam C_NS e C_PT e C_PT. A cadeia_ini leva C_NS. Esta transporta C_PT apenas se existir uma mudança de temporização de uma cadeia para a próxima. O terminal (EM) não tem que permanecer na memória dos últimos valores de p_tamanho e PT_progresso. No caso de TR, o terminal (EM) não tem que determinar se inclui p_tamanho_u em HO_sinc_u.

[000121] Como descrito acima, as únicas informações nos cabeçalhos IP/PDU/PTR que são essenciais para voz básica são os campos estáticos e o protocolo de tempo PTR(PT) e o número sequencial PTR (NS) também é muito desejável. O esquema descrito aqui alcança a transparência para estes campos de informação, e provê um cabeçalho vantajoso na eficiência de compressão. A continuidade de todos os campos estáticos e não-estáticos é mantida pelo handover. A administração da largura de banda também é

feita mais fácil, porque nas aproximações na-banda como também fora-da-banda são possíveis. Desde que a transparência é mantida para PT PTR e NS PTR, é até mesmo possível trocar de um lado para outro entre o esquema de extração de cabeçalho e o esquema de compressão de cabeçalho descritos aqui que mantêm a transparência para todos os campos. Ao comutar a compressão do cabeçalho pode ser necessário quando, por exemplo, outro meio é somado a voz.

III - Esquema Baseado no Temporizador e na Referência.

Avaliação do Esquema Baseado no Temporizador e na Referência.

[000122] O esquema baseado no temporizador e na referência é baseado nas observações (1) do protocolo de tempo PTR quando gerado na fonte PTR são correlatadas com uma função linear do tempo decorrido entre os pacotes, e (2) PT PTR são da forma $PT_0 + \text{índice} * PT_progresso$, onde PT) e PT_progresso são constantes, e o índice é um inteiro (em seguida o índice será referenciado como PT PTR acumulado).

[000123] Então, na operação normal, o protocolo de tempo PTR recebidos no descompressor são também correlacionados ao incrementar o temporizador continuamente, com uma distorção criada apenas pela flutuação acumulativa entre a fonte e o descompressor. Desde que a flutuação acumulativa inclua a flutuação de "rede" (a flutuação entre a fonte e o compressor) e a flutuação de "rádio" (a flutuação entre o compressor e o descompressor), o compressor pode calcular um limite superior da flutuação acumulativa pela adição para uma flutuação da rede observada um limite superior da flutuação de rádio. O compressor então apenas envia como PT PTR comprimido o "k" bits menos significativos do PT PTR

acumulado. O descompressor descomprime o PT PTR calculando primeiro uma aproximação, e então refinando a aproximação com a informação no PT PTR comprimido para determinar o valor exato. A aproximação é obtida somando o PT PTR do cabeçalho previamente descomprimido a um valor proporcional no tempo decorrido desde que o cabeçalho previamente descomprimido foi recebido. O valor exato do PT PTR é determinado como o mais próximo para a aproximação, cujo k bits menos significativos do PT PTR acumulado correspondente associam ao PT PTR comprimido. O compressor escolhe um valor k como o menor valor permitido, isso permitiria que o descompressor descomprimisse corretamente, baseado no limite superior da flutuação acumulativa.

Caso de Voz

[000124] Primeiro, o esquema baseado no temporizador e na referência será descrito com respeito a voz. Como um exemplo, se o intervalo de tempo entre as amostras de voz sucessivas for de 20 ms, então o protocolo de tempo PTR do cabeçalho n (gerado no tempo $n \cdot 20$ ms) = protocolo de tempo PTR do cabeçalho 0 (gerado no tempo 0) + $PT_progresso \cdot n$, onde $PT_progresso$ é uma constante dependente do codec de voz. Por conseguinte, o PT PTR nos cabeçalhos entrantes no descompressor também segue um padrão linear como uma função do tempo, mas menos próximo, devido à flutuação de retardo entre a fonte e o descompressor. Na operação normal (ausência de estrondos ou falhas), a flutuação de retardo é limitada, para satisfazer as exigências do tráfego de tempo real de conversação.

[000125] Neste esquema, o receptor usa um temporizador para obter uma aproximação do PT PTR do cabeçalho atual (a ser descomprimido), e então refina a aproximação com a informação adicional recebida no cabeçalho comprimido.

[000126] Por exemplo, assuma o seguinte:

último_cabeçalho é o último cabeçalho descomprimido com êxito onde PT_último é o último PT PTR, e p_PT_último é o último PT PTR acumulado (no receptor);

T é o espaçamento de tempo normal entre duas amostras de voz sucessivas;

PT_progresso é o incremento PT PTR para todo T ms;

cabeçalho_atual é o cabeçalho de um pacote atual a ser descomprimido, onde o PT_atual é o PT PTR atual, e p_PT_atual é o PT PTR acumulado atual;

RFH é o número de sucessão de um cabeçalho cujo rec foi recebido pelo compressor PT_RFH é o PT PTR, e p_PT_RFH é o PT PTR acumulado;

Temporizador é um temporizador incrementado para todo T ms onde o compressor e descompressor cada um mantém o seu temporizador, referenciado respectivamente como temporizador_S e temporizador_R;

T_RFH é o valor do temporizador quando RFH foi recebido, e T_atual é o valor do mesmo temporizador quando o cabeçalho_atual foi recebido; e

N_flutuação (n, m) é a flutuação da rede observada do cabeçalho n relativo ao cabeçalho m (cabeçalho n é recebido subsequentemente ao cabeçalho m), onde N_flutuação (n, m) é calculado pelo compressor como a seguir:

$$N_{\text{flutuação}}(n, m) = \text{temporizador}(n, m) - (\text{PT PTR acumulado do cabeçalho } n - \text{PT PTR acumulado do cabeçalho } m);$$

onde temporizador (n, m) é o tempo decorrido do cabeçalho m para o cabeçalho n, expresso em unidades de T ms. N_flutuação (n, m) pode ser positivo ou negativo.

N_flutuação no compressor é a flutuação da rede, quantizado em unidades de T ms.

R_flutuação (n, m) é a flutuação de rádio do cabeçalho n relativo ao cabeçalho m, predito pelo compressor. R_flutuação depende apenas das características do canal compressor-descompressor (CC-CD). R_flutuação não tem que ser calculado precisamente, um limite bem superior rumo a R_flutuação é suficiente. Por exemplo, um limite superior pode ser max_rádio_flutuação, a flutuação máxima no CC-CD, se esta for conhecida.

[000127] Assim, de acordo com o anterior, a flutuação acumulativa para um pacote é calculada como a soma da flutuação da rede de rádio:

Logo, PT PTR é calculado como a seguir:

$PT_{PTR} = PT_0 + \text{índice} * PT_{\text{progresso}}$;

onde $PT_0 < PT_{\text{progresso}}$ e índice é um inteiro.

[000128] Assim, $PT_{\text{último}} = PT_0 + \text{índice}_{\text{último}} * PT_{\text{progresso}}$ e $PT_{\text{atual}} = PT_0 + \text{índice}_{\text{atual}} * PT_{\text{progresso}}$.

Compressor

[000129] O compressor envia no cabeçalho comprimido, k bits menos significativos de $p_{PT_{\text{atual}}}$.

[000130] O compressor roda o algoritmo a seguir para determinar k:

calcular max_rede_flutuação;

calcular $F1 = \text{max_rede_flutuação} + \text{max_rádio_flutuação} + J$;

onde $J = 2$ é o fator para responder pelo erro de quantização causado pelos temporizadores ao compressor e ao descompressor, que podem ser +1 ou -1; e

encontrar o menor k inteiro que satisfaça a condição:

$$(2 * F1 + 1) < 2^k.$$

[000131] A flutuação de rede no compressor pode ser calculada de três métodos diferentes, isto é o primeiro método ilustrado na Figura 13, o segundo método ilustrado na Figura 14 e o terceiro método ilustrado na Figura 15. O segundo e o terceiro métodos são descritos respectivamente abaixo em relação à Opção 1 e Opção 2. O primeiro método é adequado para calcular a flutuação de rede. Porém, os métodos preferidos para calcular a flutuação de rede no compressor é o segundo e terceiro métodos descritos respectivamente como Opção 1 e Opção 2 abaixo.

[000132] Como ilustrado na Figura 13, de acordo com o primeiro método de flutuação de rede para um pacote particular no compressor é calculado usando informação com respeito ao pacote imediatamente precedente. Por exemplo, assim a flutuação de rede para o pacote 2 (f_2) é calculado usando a informação com respeito ao pacote 1, a flutuação de rede para o pacote 3 (f_3) é calculado usando a informação com respeito ao pacote 2, a flutuação de rede para o pacote 4 (f_4) é calculado usando a informação com respeito ao pacote 3, e a flutuação de rede para o pacote 5 (f_5) é calculado usando a informação com respeito ao pacote 4.

[000133] Assim, de acordo com a Figura 13, a flutuação de rede para o pacote 2 iguala à flutuação calculada f_2 , a flutuação de rede para o pacote 3 iguala a flutuação calculada f_3 , a flutuação de rede para o pacote 4 iguala a flutuação de rede para o pacote 5 iguala a flutuação calculada f_5 .

Opção 1

[000134] Os passos usados para calcular a flutuação de rede para o segundo método da Opção 1 é ilustrado na Figura 14. Na Opção 1, a flutuação de rede para um pacote particular é calculada usando a informação

com respeito a um pacote de referência. Assim, assumindo que o pacote 2 é o pacote de referência como ilustrado na Figura 14, a flutuação f_3 do pacote 3 é calculada usando a informação com respeito ao pacote de referência 2, a flutuação f_4 do pacote 4 é calculada usando a informação com respeito ao pacote de referência 2, e a flutuação f_5 do pacote 5 é calculada usando a informação com respeito ao pacote de referência.

[000135] De acordo com o segundo método da Opção 1 como ilustrado na Figura 14, é assumido que: a flutuação $f_3 = 2$, a flutuação $f_4 = 3$ e a flutuação $f_5 = -1$, então antes do pacote 5 $N_flutuação_min = 2$ e $N_flutuação_max = 3$, considerando que no pacote 5 $N_flutuação_min = -1$ e $N_flutuação_max = 3$. Assim, a flutuação de rede máxima (Max) no pacote 5 $= N_flutuação_max - N_flutuação_min = 4$. Adequadamente, $max_rede_flutuação$ para o pacote 5 é 4. As equações para calcular a flutuação de rede de acordo com o método da Opção 1 e a sua descrição são agrupados abaixo.

[000136] A flutuação de rede do pacote atual é calculada de acordo com o método da Opção 1 como a seguir:

$$N_flutuação (cabeçalho_atual, RFH) = (T_atual - T_RFH) - (p_PT_atual - p_PT_RFH);$$

atualizar $N_flutuação_max$ e min de $N_flutuação$, onde $N_flutuação_max$ é definido como $Max \{Nflutuação (f, RFH)\}$, para todos os cabeçalhos f enviados desde RFH e incluindo RFH . $N_flutuação_min$ é definido como $Min \{Nflutuação (f, RFH)\}$, para todos os cabeçalhos f enviados desde RFH e incluindo RFH ; e

$$calcular \ max_rede_flutuação = (N_flutuação_max - N_flutuação_min).$$

[000137] Deveria ser observado que $N_flutuação_max$ e $N_flutuação_min$ podem ser positivo ou

negativo, mas $(N_{\text{flutuação_max}} - N_{\text{flutuação_min}})$ é positivo.

Opção 2

[000138] Os passos usados para calcular a flutuação de rede para o terceiro método da Opção 2 é ilustrado na Figura 15. Na Opção 2, a flutuação de um pacote em particular é calculada usando os cálculos de flutuação entre o pacote de interesse e cada um de um número predeterminado de pacotes precedentes. O número predeterminado de pacotes precedentes é definido como uma janela e tal janela pode ser de qualquer valor. No exemplo ilustrado na Figura 15, a janela tem um valor de 4 pacotes precedentes. A janela poderia ser fixada a qualquer outro valor como, por exemplo, 7 pacotes. Logo, a janela pode, por exemplo, ser fixada para ser um valor igual ao número de pacotes desde o último pacote de referência.

[000139] Como ilustrado na Figura 15, a flutuação de rede para o pacote 5 é calculada usando a informação com respeito ao pacote 1 $f(5, 1)$, ao pacote 2 $f(5, 2)$, ao pacote 3 $f(5, 3)$ e ao pacote 4 $f(5, 4)$. Como ilustrado na Figura 15, a flutuação de rede calculada para o pacote 5 com respeito a cada pacote: 1 é $f(5,1) = -2$, pacote 2 é $f(5, 2) = 3$, pacote 3 é $f(5, 3) = 4$, e pacote 4 é $f(5, 4) = 7$, então $\text{max_rede_flutuação} = 7$. As equações para calcular a flutuação de rede de acordo com o terceiro método da Opção 2 e sua descrição são agrupados abaixo.

[000140] A flutuação de rede do pacote atual é calculada de acordo com o método da Opção 2 como a seguir:

$N_{\text{flutuação calculado}}(\text{cabeçalho_atual}, j) = (T_{\text{atual}} - T_f) - (p_{\text{PT_atual}} - p_{\text{PT_f}})$ para todos os cabeçalhos f enviados antes do cabeçalho atual, e pertencendo a uma janela W onde T_f é o valor do

temporizador quando o cabeçalho f foi recebido, e p_{PT_f} é o PT PTR acumulado do cabeçalho f ; e

calcular $\max_rede_flutuação = |\text{Max } N_flutuação(\text{cabeçalho_atual}, f)|$ sobre todo f na janela W .

[000141] No caso onde uma avaliação do descompressor está disponível, a janela W inclui os cabeçalhos enviados desde que o último cabeçalho começa a ser recebido corretamente (ex., reconhecido). No caso de nenhuma avaliação, a janela W inclui os últimos cabeçalhos L enviados, onde L é um parâmetro.

Descompressor

[000142] Para descomprimir PT PTR do cabeçalho atual, o receptor calcula o tempo decorrido desde que o último_cabeçalho foi recebido, nas unidades T ms. Neste tempo, o temporizador (cabeçalho_atual, último_cabeçalho) é acrescentado a $p_{PT_último}$, para dar uma aproximação p_{PT_atual} . O receptor determina o valor exato de p_{PT_atual} escolhendo o valor mais próximo para a aproximação, cujo k bits menos significativos associam ao PT PTR comprimido. PT_atual é então calculado como $PT_0 + (p_{PT_atual}) * PT_progresso$.

[000143] Temporizador (cabeçalho_atual, último_cabeçalho) pode ser calculado como $(T_atual - T_último)$, onde T_atual e $T_último$ são os valores do Temporizador_R quando o cabeçalho_atual e o último_cabeçalho respectivamente foram recebidos.

Prova de correção

[000144] Para provar as correções do esquema baseado no temporizador e na referência a seguir é assumido que:

aprox_PT é a aproximação de p_{PT_atual} , calculado pelo descompressor como $p_{PT_último} + \text{temporizador}(\text{cabeçalho_atual}, \text{último_cabeçalho})$; e

exato_PT é o valor exato de p_PT_atual.

[000145] Baseado no anterior então:

$$|\text{aprox_PT} - \text{Exato_PT}| \leq |\text{flutuação}(\text{cabeçalho_atual}, \text{último_cabeçalho})|;$$

Devido à definição de max_rede_flutuação no compressor:

$$|\text{flutuação}(\text{cabeçalho_atual}, \text{último_cabeçalho})| \leq F1$$

onde $F1 = \text{max_rede_flutuação} + \text{max_radio_flutuação} + F$

F é o fator adicionado para responder pelo erro de quantização causado pelos temporizadores no compressor e no descompressor, os quais podem ser +1 ou -1. Então, $F = 2$ é suficiente.

[000146] Assim, a seguir:

$$|\text{aprox_PT} - \text{exato_PT}| \leq F1$$

Para calcular o exato_PT sem ambiguidade, é suficiente escolher k, tal que a condição $(2 \cdot F1 + 1) < 2^k$ seja satisfeita.

Caso do Pacote Desordenar Antes do Compressor

[000147] A desordem do pacote pode ser detectada pelo número sequencial PTR decrescente (NS PTR). Quando isto ocorre, o compressor pode codificar PT PTR acumulado usando um esquema diferente, por exemplo, VLE. O descompressor é notificado da codificação diferente através dos bits indicadores apropriados no cabeçalho comprimido.

[000148] Outra opção é aplicar o algoritmo do esquema baseado no temporizador e na referência normal - desordenação resultará provavelmente em um valor maior de k.

Enlace Ascendente

[000149] Nos enlaces sem fio, para a direção ascendente, a flutuação de rede é zero (desde que ambos a fonte PTR e o compressor fiquem situados no terminal sem

fio), e a flutuação de rádio normalmente é limitada e controlada para permanecer bem pequena. Então, o k esperado será muito pequeno e constante, o que minimiza a flutuação do tamanho do cabeçalho. Esta é uma vantagem muito significativa para a administração da largura de banda, desde que para o enlace ascende, o terminal tenha normalmente que pedir uma largura de banda aumentada da rede. Além disso, não há nenhuma desordem de pacote. Por conseguinte, o esquema baseado no temporizador é extremamente adequado para o enlace ascendente.

Enlace Descendente

[000150] Para a direção de enlace descendente, a flutuação de rede não é zero, mas a flutuação global normalmente é pequena para satisfazer as exigências de tempo real. O k esperado ainda será pequeno e normalmente constante. Pode haver mais flutuação em k, mas a administração da largura de banda é menos de uma emissão, desde que a rede controle a distribuição da largura de banda.

Handover

[000151] Nos sistemas celulares, existe um enlace de rádio da EM-para-rede e um enlace de rádio da rede-para-EM, referenciado, respectivamente, como ascendente e descendente. Quando a compressão/descompressão é aplicada nas ligações celulares, existe uma função baseada na EM, EM_AD (adaptador da EM) que faz a compressão e a descompressão, respectivamente, para o ascendente e o descendente. Existe uma entidade baseada na rede, denominada de IRA_AD (infraestrutura da rede de acesso - adaptador), este faz a descompressão e a compressão respectivamente para o ascendente e o descendente.

[000152] Um caso específico de handover a considerar é o handover inter-IRA_AD, onde pode haver um

rompimento causado pela comutação do IRA_AD antigo a um IRA_AD novo. O assunto é como manter a continuidade da informação pelo handover de forma que depois do handover, a compressão/descompressão na EM_AD e no IRA_AD novo continue sem rompimento.

[000153] Há dois métodos alternativos para handover, descritos abaixo.

Primeiro Método

[000154] O primeiro método usa o esquema de levar uma informação instantânea de contexto trocado entre o IRA_AD e EM_AD, com o método para estabelecer comunicação (handshake), como descrito no pedido relacionado Nº 09/522,497, depositado no dia 09 de março de 1999, na mesma data do presente pedido, "PROCEDIMENTO DE HANDOVER EFICIENTE PARA A COMPRESSÃO DO CABEÇALHO" por K. LE. Para o PT PTR, a informação de contexto contém o PT PTR completo do cabeçalho de referência. Logo depois do handover, os compressores (EM_AD para o enlace ascendente e IRA_AD para o enlace descendente) temporariamente descontinuem usando o esquema baseado no temporizador e enviam um PT PTR comprimido com respeito ao valor de referência. Por exemplo, a codificação VLE como descrita no pedido relacionado Nº 09/522, 497, depositado no dia 09 de março de 1999, na mesma data do presente pedido "PROCEDIMENTO DE HANDOVER EFICIENTE PARA A COMPRESSÃO DO CABEÇALHO" por K. LE., poderia ser usado. Uma vez que o reconhecimento tenha sido recebido, o compressor usa o valor reconhecido como o RFH, e comuta de volta para o esquema baseado no temporizador.

Segundo Método

[000155] O segundo método continua usando o esquema baseado no temporizador pelo handover.

Enlace Descendente

[000156] Não há nenhuma descontinuidade no lado receptor, que é a EM. A regra do compressor é transferida de um IRA_AD para outro. Depois do handover, os cabeçalhos são direcionados em um caminho novo que passa pelo IRA_AD novo em vez do IRA_AD antigo.

Compressor

[000157] O IRA_AD antigo transfere ao IRA_AD novo, um instantâneo da informação seguinte: T_RFH, p_PT_RFH, o valor atual do temporizador_S, PT0, e PT_progresso, enquanto usando o método para estabelecer comunicação (handshake). (os valores instantâneos serão referenciados com uma estrela, ex., T_RFH*). O IRA_AD novo inicializa seu temporizador_S com o valor atual do temporizador_S recebido do IRA_AD antigo e inicia incrementando o temporizador a todo T ms. A inicialização do temporizador_S com o valor atual do temporizador_S do IRA_AD antigo é uma descrição conceitual. Se existir um único temporizador_S compartilhado por fluxos múltiplos, o temporizador_S atual não é re-inicializado. Preferivelmente, a compensação entre o temporizador_S e o valor do IRA_AD antigo é registrada. A consideração é levada em conta nos cálculos futuros. Para comprimir o primeiro cabeçalho depois do handover, o IRA_AD novo envia k bits menos significativos de p_PT_atual. O IRA_AD novo determina k, e o número de bits a ser usado, como a seguir:

$$F2 = \text{limite superior de } N_{\text{flutuação}} \text{ (cabeçalho atual, RFH*)} + \text{max-radio}_{\text{flutuação}} + J,$$

onde k é selecionado para satisfazer uma condição de $(2 * F2 + 1) < 2^k$.

[000158] No anterior, $\text{max_radio_flutua\c{c}\~{a}o}$ é a flutuação máxima no segmento entre o novo IRA_AD e a EM_AD.

[000159] Um limite superior de $N_{\text{flutua\c{c}\~{a}o}}$ (cabeçalho atual, RFH*) é calculado como a seguir:

$|\text{Temporizador}(\text{cabe\c{c}alho_atual}, \text{RFH}^*) - (\text{p_PT_atual} - \text{p_PT_RFH}^*)| + T_{\text{handover}}$, onde $\text{temporizador}(\text{cabe\c{c}alho_atual}, \text{RFH}^*)$ é $(T_{\text{atual}} - T_{\text{RFH}^*})$;

T_{atual} é o valor do temporizador_S no IRA_AD novo quando o $\text{cabe\c{c}alho_atual}$ foi recebido;

T_{RFH^*} é o valor recebido do IRA_AD antigo;

T_{handover} é o limite superior do tempo para transferir a informação de contexto do IRA_AD antigo para o IRA_AD novo, expresso em unidades de T ms; e $J = 2$.

Descompressor

[000160] Para descomprimir PT PTR do $\text{cabe\c{c}alho_atual}$, o receptor calcula o tempo decorrido desde que RFH foi recebido, em unidades de T ms. Neste tempo, o $\text{temporizador}(\text{cabe\c{c}alho_atual}, \text{RFH})$, é acrescentado a p_PT_RFH , para dar uma aproximação de p_PT_atual . O receptor então determina o valor exato de p_PT_atual escolhendo o valor mais próximo da aproximação, cujo k bits menos significativos associam ao PT PTR comprimido. PT_atual é calculado então como $\text{PT0} + (\text{p_PT_atual}) * \text{PT_progresso}$.

[000161] O tempo decorrido desde que RFH foi recebido pode ser calculado como $(T_{\text{atual}} - T_{\text{RFH}})$.

Caso de falha

[000162] Quando as informações de contexto não podem ser transferidas para o IRA_AD novo de uma maneira oportuna, o IRA_AD novo enviará o PT PTR completo até que um reconhecimento seja recebido.

Enlace Ascendente

[000163] A regra do descompressor é transferida de um IRA_AD a outro. O compressor permanece ancorado na EM.

Descompressor

[000164] O IRA_AD antigo transfere ao IRA_AD novo, um instantâneo da informação seguinte: T_{RFH}^* , $p_{PT_{RFH}^*}$, valor atual de temporizador_R*, PT_0 , e $PT_{progresso}$, enquanto usando o método para estabelecer comunicação (handshake). O IRA_AD novo inicializa seu temporizador_R com o valor atual do temporizador_R recebido do IRA_AD² antigo e começa incrementando o temporizador a todo T ms. A inicialização do temporizador_R com o valor atual temporizador_R do IRA_AD antigo é apenas uma descrição conceitual. Se houver um único temporizador_R compartilhado para fluxos múltiplos, o temporizador_R atual não é re-inicializado. Preferivelmente, a compensação entre o temporizador_R e o valor do IRA_AD antigo é registrada. Essa compensação é levada em conta nos cálculos futuros. Ao descomprimir o primeiro cabeçalho depois do handover, o IRA_AD novo calcula o temporizador (cabeçalho_atual, RFH) acrescentando este a $p_{PT_{RFH}^*}$, para dar uma aproximação de $p_{PT_{atual}}$. O receptor então determina o valor exato para $p_{PT_{atual}}$ escolhendo o valor mais próximo para a aproximação, cujo k bits menos significativos associam ao PT_{PTR} comprimido. PT_{atual} é então calculado como $PT_0 + (p_{PT_{atual}}) * PT_{progresso}$.

[000165] Temporizador (cabeçalho_atual, RFH) pode ser calculado como $(T_{atual} - T_{RFH}^*)$. T_{atual} é o valor do temporizador_R quando o cabeçalho_atual foi recebido.

Compressor

[000166] A EM_AD envia os k bits menos significativos p_PT_atual. Determina k, o número de bits a ser usado, como a seguir:

Calcular $F2 = \text{limite superior de } N_{\text{flutuação}}(\text{cabeçalho_atual}, \text{RFH}^*) + \text{max_radio_flutuação} + F$;

Quando k é selecionado para satisfazer uma condição de $(2^{J2+1}) < 2^k$;

Aqui max_radio_flutuação é a flutuação máxima no segmento entre o IRA_AD novo e a EM_AD.

* Limite superior de $N_{\text{flutuação}}(\text{cabeçalho_atual}, \text{RFH}^*)$ é calculado como o $|\text{Temporizador}(\text{cabeçalho_atual}, \text{RFH}^*) - (\text{p_PT_atual_cabeçalho} - \text{p_PT_RFH}^*)| + T_{\text{handover}}$, onde temporizador(cabeçalho_atual, RFH*) é $(T_{\text{atual}} - T_{\text{RFH}^*})$;

T_{atual} é o valor do temporizador_S para o IRA_AD novo quando o cabeçalho_atual foi recebido; T_{RFH^*} é o valor recebido do IRA_AD antigo;

T_{handover} é o limite superior do tempo para transferir a informação de contexto do IRA_AD antigo para IRA_AD novo, expresso em unidades de T ms; e $F = 2$.

Caso de Falha

[000167] Quando as informações de contexto não podem ser transferidas para o IRA_AD novo de uma maneira oportuna, o IRA_AD novo notificará a EM_AD, a qual envia o PT PTR completo até um reconhecimento ser recebido.

Desempenho do Esquema

[000168] Devido às exigências de tempo real de conversação, a flutuação acumulativa na operação normal é esperada que seja no máximo apenas algum tempo T ms. Então, o valor de k em torno de 4 ou 5 são suficientes, como a flutuação de até 16 a 32 amostras de voz pode ser corrigida.

[000169] As vantagens deste esquema são como a seguir:

- o tamanho do cabeçalho comprimido é constante e pequeno. O cabeçalho comprimido inclui um tipo de mensagem que indica tipicamente o tipo de mensagem (k_l bits), uma máscara de bits indicando qual campo está mudando, e um campo que contém os k bits menos significativos do índice atual (k bits). Assumindo que uma máscara de 4-bits EMTI é usada, e $k_l = 4$, o tamanho do cabeçalho comprimido quando apenas o PT PTR muda (este caso é sem dúvida o mais frequente) é de 1.5 bytes. Além disso, o tamanho não muda como uma função do comprimento do intervalo de silêncio;

- nenhuma sincronização é requerida entre o processo do temporizador e o processo do descompressor;

- robustez a erros, como a informação parcial PT PTR no cabeçalho comprimido está mesmo contida e apenas precisa ser combinada com o temporizador do receptor para render o valor completo PT PTR. A perda ou a corrupção do cabeçalho não invalidarão os cabeçalhos comprimidos subsequentes.

[000170] O compressor precisa manter pouca informação de memória:

T_{RFH} , $p_{PT_{RFH}}$, $N_{flutuação_{max}}$, $N_{flutuação_{min}}$, PT_0 , e $PT_{progresso}$ na Opção 1 e $\{T_f, p_{PT_f}\}$, para todo o f na janela W , PT_0 , e $PT_{progresso}$ na Opção 2.

Redução da Flutuação

[000171] Devido às exigências de tempo real de conversação, pode esperar razoavelmente que as várias flutuações descritas acima está na ordem de alguns T ms na operação normal. Porém, a regra não pode sair dos casos onde a flutuação é maior e requereria então um k maior. Por exemplo, pode haver condições anormais no caminho da fonte

PTR para o receptor (falhas, etc.) durante as quais a flutuação se torna excessiva. Também, pode haver casos onde um valor constante de k é desejado ou desejável. Para lidar com estes casos, uma função de redução de flutuação pode ser implementada como uma interface no compressor para filtrar os pacotes com flutuação excessiva (isto é, a flutuação excedendo um pouco do valor limite).

[000172] No caso estacionário (nenhum handover), a flutuação é calculada como $F1$ e comparada a um limite estacionário como a seguir:

$$F1 = (N_{\text{flutuação_max}} - N_{\text{flutuação_min}}) + \text{max_radio_flutuação} + J.$$

[000173] No caso de handover, a flutuação é calculada como $F2$ e comparada a um limite de handover como a seguir:

$$F2 = \text{temporizador}(\text{cabeçalho_atual}, \text{RFH}^*) - (\text{p_PT_atual} - \text{p_PT_RFH}^*) + T_{\text{handover}} + \text{max_radio_flutuação} + J.$$

[000174] A diferença principal com respeito ao caso estacionário `nenhum_handover`, é a adição de T_{handover} . Para poder executar o handover em 100 ms, T_{handover} deve ser limitado através de aproximadamente 100 ms na prática, assim $T_{\text{handover}} =$ aproximadamente 5 ou 6 unidades de T ms ($T = 20$ ms). O valor de $k = 5$ é suficiente.

[000175] O estacionário e os limites do handover podem ou não serem os mesmos.

Caso de Vídeo

[000176] No caso de uma fonte de vídeo PTR, não é necessariamente verdadeiro que existe um espaçamento de tempo constante entre os pacotes, e além disso, o PTR necessariamente não incrementa por um progresso constante de um pacote para o próximo. Porém, o PTR e o

espaçamento de tempo entre os pacotes são discretos. Assim, como a seguir:

protocolo de tempo PTR do pacote m = protocolo de tempo PTR do pacote 0 (gerado no tempo 0) + $PT_progresso * [índice + ajuste (m)]$,

onde $PT_progresso$ é uma constante dependente no codec, e $ajuste(m)$ é um inteiro que depende de m e reflete as diferenças com respeito a um comportamento linear tal como em voz; e o espaçamento de tempo entre dois pacotes sucessivos é um múltiplo inteiro de T ms.

[000177] A seguir, o comportamento na fonte PTR é referenciado como comportamento linear ajustado. Usando a mesma anotação como para voz, $PT_último = PT0 + PT_progresso * [índice_último + ajuste (índice_último)]$, e $T_atual = PT0 + PT_progresso [índice_atual] + ajuste(índice_atual)$. O parâmetro Ajuste pode ser positivo ou negativo. Assim, a diferença principal comparada à voz é o termo adicional Ajuste.

[000178] O PT PTR é um cabeçalho entrando no descompressor também segue um padrão linear ajustado como uma função de tempo, mas menos próximo, devido à flutuação de retardo entre a fonte e o descompressor. Na operação normal (ausência de estrondos ou falhas), a flutuação de retardo é limitada, para satisfazer as exigências do tráfego de tempo real de conversação.

[000179] Como acima é assumido que PT_PTR acumulado do cabeçalho $atual = índice_atual + ajuste (índice_atual)$. A mesma anotação será usada com respeito a p_PT_atual , por exemplo.

Compressor

[000180] O compressor envia no cabeçalho comprimido os k bits menos significativos de p_PT_atual .

Descompressor

[000181] O algoritmo a ser usado é igual ao de voz.

Handover

[000182] Os dois métodos alternativos para handover descritos para voz, se aplicam também para vídeo.

Valor de k

[000183] Para voz, foi apresentado que $k = 4$ ou 5 são suficientes ($2^k = 16$ ou 32). No caso de vídeo, um valor maior de k é requerido devido ao Ajuste. Considerando que, o vídeo é estruturado em 30 quadros por segundo, $\text{Ajuste} < 30$. Então, $k = 7$ ou 8 bits deveriam ser suficientes em uma operação normal.

[000184] Especificamente, são ilustradas e/ou descritas várias incorporações da presente invenção. Porém será apreciado que modificações e variações da presente invenção estão cobertas pelos ensinamentos anteriores e dentro do quadro das reivindicações apenas sem sair do espírito e escopo planejado da invenção.

[000185] Enquanto a presente invenção foi descrita em detalhes e ilustradamente nos desenhos apenas, estes não estão limitados a tais detalhes, uma vez que várias mudanças e modificações reconhecíveis pelo técnico no assunto podem ser feitas à invenção sem sair do espírito e o escopo desta.

REIVINDICAÇÕES

1. Método para transmitir em uma rede entre uma fonte (102) e um receptor (130), um campo de cabeçalho atual de um pacote atual usando a técnica de compressão baseada no temporizador, o método é **CARACTERIZADO** pelo fato de que compreende:

prover, do compressor (115) para o descompressor (137), um valor inicial do campo de cabeçalho;

calcular, no compressor (115), um campo de cabeçalho comprimido do pacote atual baseado no campo de cabeçalho atual do pacote atual e da flutuação, onde calcular compreende:

calcular o efeito da flutuação da rede entre a fonte (102) e o descompressor (137) usado na transmissão dos pacotes; e

calcular o campo de cabeçalho comprimido como uma parte do valor do campo, a parte sendo uma função da flutuação;

receber o campo de cabeçalho comprimido do pacote atual no descompressor (137);

estimar o campo de cabeçalho do pacote atual baseado no tempo decorrido no descompressor (137) entre a recepção do campo de cabeçalho comprimido do pacote atual e a recepção do campo de cabeçalho do pacote prévio, o qual foi descomprimido e de um valor do campo descomprimido do pacote prévio; e

corrigir o campo de cabeçalho atual estimado baseado no campo de cabeçalho comprimido recebido no descompressor (137).

2. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que calcular o efeito da flutuação compreende:

calcular o efeito de flutuação da rede antes do compressor (115); e

calcular o efeito de flutuação da rede entre o compressor (115) e o descompressor (137).

3. Método, de acordo com a reivindicação 2, **CARACTERIZADO** pelo fato de que o efeito de flutuação da rede entre o compressor (115) e o descompressor (137) é estabelecido para um valor de limite superior a flutuação.

4. Método, de acordo com a reivindicação 2, **CARACTERIZADO** pelo fato de que calcular o efeito de flutuação da rede antes do compressor (115) compreende:

calcular o efeito de flutuação do pacote atual usando a informação com respeito ao pacote de referência.

5. Método, de acordo com a reivindicação 2, **CARACTERIZADO** pelo fato de que calcular o efeito de flutuação da rede antes do compressor (115) compreende:

calcular o efeito de flutuação do pacote atual usando a informação com relação ao pacote atual e a cada um de um número pré-determinado de pacotes precedentes.

6. Método, de acordo com a reivindicação 2, **CARACTERIZADO** pelo fato de que calcular o efeito de flutuação da rede antes do compressor (115) compreende:

calcular o efeito de flutuação do pacote atual usando a informação com relação ao pacote atual e a cada pacote precedente até um pacote de referência.

7. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que calcular, no compressor (115), o campo de cabeçalho comprimido do pacote atual compreende:

calcular o campo de cabeçalho comprimido do pacote atual como k bits menos significativos do valor do campo, onde k é um inteiro.

8. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que o campo de cabeçalho compreende o protocolo de tempo.

9. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que o campo de cabeçalho compreende o protocolo de tempo PTR.

10. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que calcular, no compressor (115), o campo de cabeçalho comprimido do pacote atual compreende:

converter o valor do campo em outro valor, referenciado como um valor empacotado, o qual tem poucos bits para representar; e

calcular o campo de cabeçalho comprimido do pacote atual como k bits menos significativos do valor empacotado, onde k é um inteiro.

11. Método para descomprimir o campo de cabeçalho atual do pacote atual transmitido em uma rede de um compressor (115) para um descompressor (137), o método é **CARACTERIZADO** pelo fato de que compreende:

receber um campo de cabeçalho comprimido de um pacote atual no descompressor (137), o campo de cabeçalho comprimido tendo sido calculado no compressor (115) como uma parte do valor do campo, o qual é calculado como um efeito de flutuação da rede entre uma fonte (102) e o descompressor (137) usado na transmissão dos pacotes;

calcular uma aproximação do campo de cabeçalho do pacote atual no descompressor (137) baseado no tempo decorrido desde a chegada do campo de cabeçalho comprimido prévio no descompressor (137) e de um valor do campo descomprimido do pacote prévio;

calcular a quantidade de correção do campo de cabeçalho para o pacote atual no descompressor (137)

baseado no campo de cabeçalho descomprimido do pacote atual; e

descomprimir o campo de cabeçalho comprimido do pacote atual no descompressor (137) ao ajustar a aproximação do campo de cabeçalho do pacote atual por uma quantidade baseada na quantidade de correção do campo de cabeçalho.

12. Método, de acordo com a reivindicação 11, **CARACTERIZADO** pelo fato de que o efeito de flutuação da rede entre a fonte (102) e o descompressor (137) usado na transmissão dos pacotes é calculado ao calcular o efeito de flutuação da rede e ao calcular o efeito de flutuação da rede entre o compressor (115) e o descompressor (137).

13. Método, de acordo com a reivindicação 12, **CARACTERIZADO** pelo fato de que o efeito de flutuação da rede entre o compressor (115) e o descompressor (137) é estabelecido para um valor limite superior para a flutuação.

14. Método, de acordo com a reivindicação 12, **CARACTERIZADO** pelo fato de que calcular o efeito de flutuação da rede antes do compressor (115) compreende:

calcular o efeito de flutuação do pacote atual usando a informação com respeito ao pacote de referência.

15. Método, de acordo com a reivindicação 12, **CARACTERIZADO** pelo fato de que calcular o efeito de flutuação da rede antes do compressor (115) compreende:

calcular o efeito de flutuação do pacote atual usando a informação com relação ao pacote atual e a cada um de um número pré-determinado de pacotes precedentes.

16. Método, de acordo com a reivindicação 12, **CARACTERIZADO** pelo fato de que calcular o efeito de flutuação da rede antes do compressor (115) compreende:

calcular o efeito de flutuação do pacote atual usando a informação com relação ao pacote atual e a cada pacote precedente até um pacote de referência.

17. Método, de acordo com a reivindicação 11, **CARACTERIZADO** pelo fato de que o campo de cabeçalho comprimido é calculado como k bits menos significativos do valor do campo, onde k é um inteiro.

18. Método, de acordo com a reivindicação 11, **CARACTERIZADO** pelo fato de que o campo de cabeçalho comprimido é calculado como k bits menos significativos do valor empacotado, onde k é um inteiro; e

onde o descompressor (137) calcula uma aproximação do valor empacotado baseado no tempo decorrido desde a chegada do pacote prévio e do valor empacotado descomprimido do pacote prévio.

19. Método para executar handover entre a primeira e a segunda entidades de rede (112, 114) no sistema, a primeira e a segunda entidades de rede (112, 114) interfaceando o descompressor móvel (115) ao terminal de fonte (102) quando o descompressor móvel (115) é localizado na primeira e segunda áreas, respectivamente, o método é **CARACTERIZADO** pelo fato de que compreende:

receber um valor inicial do campo de cabeçalho na primeira entidade de rede (112) e no descompressor móvel (115) do terminal de fonte (102), a primeira entidade de rede (112) interfaceando o descompressor móvel (115), localizado na primeira área, para o terminal de fonte (102);

receber um campo de cabeçalho do primeiro pacote que é endereçado para o descompressor móvel (115) na primeira entidade de rede (112) do terminal de fonte (102);

comprimir o campo de cabeçalho do primeiro pacote na primeira entidade de rede (112) e enviar o primeiro

campo de cabeçalho comprimido do primeiro pacote para o descompressor móvel (115), o primeiro campo de cabeçalho comprimido sendo calculado como uma parte do valor do campo, o qual é calculado como o primeiro efeito de flutuação da rede entre o terminal de fonte (102) e o descompressor móvel (115) usado na transmissão dos pacotes;

receber e descomprimir o primeiro campo de cabeçalho comprimido do primeiro pacote no descompressor móvel (115) baseado no tempo decorrido desde a chegada do pacote prévio e o valor do campo descomprimido do pacote prévio;

o descompressor móvel (115) movendo da primeira área para a segunda área;

transmitir a informação de inicialização para a segunda entidade de rede (114) para inicializar a segunda entidade de rede (114) para compressão;

receber e comprimir, na segunda entidade de rede (114), um campo de cabeçalho do segundo pacote do terminal de fonte (102) que é endereçado para o descompressor móvel (115), e enviar o segundo campo de cabeçalho comprimido do segundo pacote para o descompressor móvel (115), o segundo campo de cabeçalho comprimido sendo calculado como uma parte do valor do campo, o qual é calculado como o segundo efeito de flutuação da rede entre o terminal de fonte (102) e o descompressor móvel (115) usado na transmissão dos pacotes e o tempo para transmitir a informação de inicialização para a segunda entidade de rede (114); e

receber e descomprimir o segundo campo de cabeçalho comprimido do segundo pacote no descompressor móvel (115) baseado no tempo decorrido desde a chegada de um pacote prévio e do valor do campo descomprimido do pacote prévio.

20. Método, de acordo com a reivindicação 15, **CARACTERIZADO** pelo fato de que cada um do primeiro e segundo efeito de flutuação da rede entre o terminal de fonte (102) e o descompressor móvel (115) usado na transmissão dos pacotes é calculado ao calcular o efeito de flutuação da rede antes do compressor (137) e calcular o efeito de flutuação da rede entre o compressor (137) e o descompressor (115).

21. Método, de acordo com a reivindicação 20, **CARACTERIZADO** pelo fato de que o efeito de flutuação da rede entre o compressor (137) e o descompressor (115) é estabelecido para um valor de limite superior a flutuação.

22. Método, de acordo com a reivindicação 19, **CARACTERIZADO** pelo fato de que o primeiro pacote é o pacote imediatamente precedente ao handover e o segundo pacote é o pacote imediatamente subsequente ao handover.

23. Método, de acordo com a reivindicação 20, **CARACTERIZADO** pelo fato de que calcular o efeito de flutuação da rede antes do compressor (137) compreende:

calcular o efeito de flutuação do pacote atual usando a informação com respeito ao pacote de referência.

24. Método, de acordo com a reivindicação 20, **CARACTERIZADO** pelo fato de que calcular o efeito de flutuação da rede antes do compressor (137) compreende:

calcular o efeito de flutuação do pacote atual usando a informação com relação ao pacote atual e a cada um de um número pré-determinado de pacotes precedentes.

25. Método, de acordo com a reivindicação 20, **CARACTERIZADO** pelo fato de que calcular o efeito de flutuação da rede antes do compressor (137) compreende:

calcular o efeito de flutuação do pacote atual usando a informação com relação ao pacote atual e a cada pacote precedente até um pacote de referência.

26. Método, de acordo com a reivindicação 19, **CARACTERIZADO** pelo fato de que o campo de cabeçalho compreende o protocolo de tempo.

27. Método, de acordo com a reivindicação 19, **CARACTERIZADO** pelo fato de que o campo de cabeçalho comprimido é calculado como k bits menos significativos do valor do campo, onde k é um inteiro.

28. Método, de acordo com a reivindicação 19, **CARACTERIZADO** pelo fato de que o campo de cabeçalho comprimido é calculado como k bits menos significativos do valor empacotado, onde k é um inteiro; e

onde o descompressor (115) calcula uma aproximação do valor empacotado baseado no tempo decorrido desde a chegada do pacote prévio e do valor empacotado descomprimido do pacote prévio.

29. Método para executar handover entre a primeira e a segunda entidades de rede (112, 114), a primeira e a segunda entidades de rede (112, 114) interfaceando o compressor móvel ao terminal receptor (130) quando o compressor móvel (115) é localizado na primeira e segunda áreas, respectivamente, o método é **CARACTERIZADO** pelo fato de que compreende:

receber um valor inicial do campo de cabeçalho na primeira entidade de rede (112) do compressor móvel (115);

receber, na primeira entidade de rede (112), o primeiro campo de cabeçalho comprimido do primeiro pacote recebido do compressor móvel (115), o primeiro campo de cabeçalho comprimido tem sido calculado no compressor móvel (115) como uma parte do valor do campo, o qual é calculado como o primeiro efeito de flutuação da rede entre a origem (112, 114) e o descompressor (137) usado na transmissão dos pacotes;

descomprimir, na primeira entidade de rede (112), o primeiro campo de cabeçalho comprimido do primeiro pacote baseado no tempo decorrido desde a chegada do pacote prévio e do valor do campo descomprimido do pacote prévio;

o compressor móvel (115) movendo da primeira área para a segunda área;

transmitir a informação de inicialização para a segunda entidade de rede (114) para inicializar a segunda entidade de rede (114) para compressão;

receber, na segunda entidade de rede (114), o segundo campo de cabeçalho comprimido do segundo pacote recebido do compressor móvel (115), o segundo campo de cabeçalho comprimido tem sido calculado no compressor móvel (115) como uma parte do valor do campo, o qual é calculado como o segundo efeito de flutuação da rede entre a origem (112, 114) e o descompressor (137) usado na transmissão dos pacotes e do tempo para transmitir a informação de inicialização para a segunda entidade de rede (114); e

descomprimir, na segunda entidade de rede (114), o segundo campo de cabeçalho comprimido do segundo pacote baseado no tempo decorrido desde a chegada do pacote prévio e do valor do campo descomprimido do pacote prévio.

30. Método, de acordo com a reivindicação 29, **CARACTERIZADO** pelo fato de que o primeiro e o segundo efeito de flutuação da rede entre a origem (112, 114) e o descompressor (137) usado na transmissão dos pacotes é calculado ao calcular o efeito de flutuação da rede antes do compressor móvel (115) e ao calcular o efeito de flutuação da rede entre o compressor móvel (115) e o descompressor (137).

31. Método, de acordo com a reivindicação 30, **CARACTERIZADO** pelo fato de que o efeito de flutuação da rede entre o compressor móvel (115) e o descompressor (137)

é estabelecido para um valor de limite superior a flutuação.

32. Método, de acordo com a reivindicação 29, **CARACTERIZADO** pelo fato de que o primeiro pacote é o pacote imediatamente precedente ao handover e o segundo pacote é o pacote imediatamente subsequente ao handover.

33. Método, de acordo com a reivindicação 30, **CARACTERIZADO** pelo fato de que calcular o efeito de flutuação da rede antes do compressor móvel (115) compreende:

calcular o efeito de flutuação do pacote atual usando a informação com respeito ao pacote de referência.

34. Método, de acordo com a reivindicação 30, **CARACTERIZADO** pelo fato de que calcular o efeito de flutuação da rede antes do compressor móvel (115) compreende:

calcular o efeito de flutuação do pacote atual usando a informação com relação ao pacote atual e a cada um de um número pré-determinado de pacotes precedentes.

35. Método, de acordo com a reivindicação 30, **CARACTERIZADO** pelo fato de que calcular o efeito de flutuação da rede antes do compressor móvel (115) compreende:

calcular o efeito de flutuação do pacote atual usando a informação com relação ao pacote atual e a cada pacote precedente até um pacote de referência.

36. Método, de acordo com a reivindicação 29, **CARACTERIZADO** pelo fato de que o campo de cabeçalho compreende o protocolo de tempo.

37. Método, de acordo com a reivindicação 29, **CARACTERIZADO** pelo fato de que o campo de cabeçalho comprimido é calculado como k bits menos significativos do valor empacotado, onde k é um inteiro; e

onde o descompressor (137) calcula uma aproximação do valor empacotado baseado no tempo decorrido desde a chegada do pacote prévio e do valor empacotado descomprimido do pacote prévio.

38. Método, de acordo com a reivindicação 29, **CARACTERIZADO** pelo fato de que o campo de cabeçalho comprimido é calculado como k bits menos significativos do valor do campo, onde k é um inteiro.

39. Sistema de comunicação, **CARACTERIZADO** pelo fato de que compreende:

uma fonte (102) provendo uma pluralidade de pacotes, cada pacote incluindo um campo de cabeçalho, a fonte sendo acoplada a uma rede;

um receptor (130) incluindo um descompressor (137);

uma entidade de rede (112) acoplada a rede e ao receptor (130) pela rede entre a entidade de rede (112) e o receptor (130), a entidade de rede incluindo o compressor (115) para executar a compressão do campo de cabeçalho para ao menos alguns dos pacotes enviados da fonte (102) e direcionado para o receptor (130), a entidade de rede (112) incluindo uma função de redução de flutuação para calcular a flutuação nos pacotes direcionados para o receptor (130) e descartar os pacotes possuindo uma flutuação que é maior do que o valor pré-determinado;

onde a flutuação é calculada como o total da quantidade de flutuação causada pela rede entre a fonte (102) e o descompressor (137) incluído no receptor (130).

40. Sistema, de acordo com a reivindicação 39, **CARACTERIZADO** pelo fato de que a flutuação é calculada ao calcular o efeito de flutuação da rede antes do compressor (115) e ao calcular o efeito de flutuação da rede entre o compressor (115) e o descompressor (137).

41. Sistema, de acordo com a reivindicação 39, **CARACTERIZADO** pelo fato de que o efeito de flutuação da rede entre o compressor (115) e o descompressor (137) é estabelecido para um valor de limite superior a flutuação.

42. Sistema, de acordo com a reivindicação 40, **CARACTERIZADO** pelo fato de que calcular a flutuação da rede antes do compressor (115) é calculada ao calcular o efeito de flutuação do pacote atual usando a informação com respeito ao pacote de referência.

43. Sistema, de acordo com a reivindicação 40, **CARACTERIZADO** pelo fato de que calcular o efeito de flutuação da rede antes do compressor (115) compreende:

calcular o efeito de flutuação do pacote atual usando a informação com relação ao pacote atual e a cada um de um número pré-determinado de pacotes precedentes.

44. Sistema, de acordo com a reivindicação 40, **CARACTERIZADO** pelo fato de que calcular o efeito de flutuação da rede antes do compressor (115) compreende:

calcular o efeito de flutuação do pacote atual usando a informação com relação ao pacote atual e a cada pacote precedente até um pacote de referência.

45. Sistema, de acordo com a reivindicação 39, **CARACTERIZADO** pelo fato de que o campo de cabeçalho comprimido é calculado como k bits menos significativos do valor empacotado, onde k é um inteiro; e

onde o descompressor (137) calcula uma aproximação do valor empacotado baseado no tempo decorrido desde a chegada do pacote prévio e do valor empacotado descomprimido do pacote prévio.

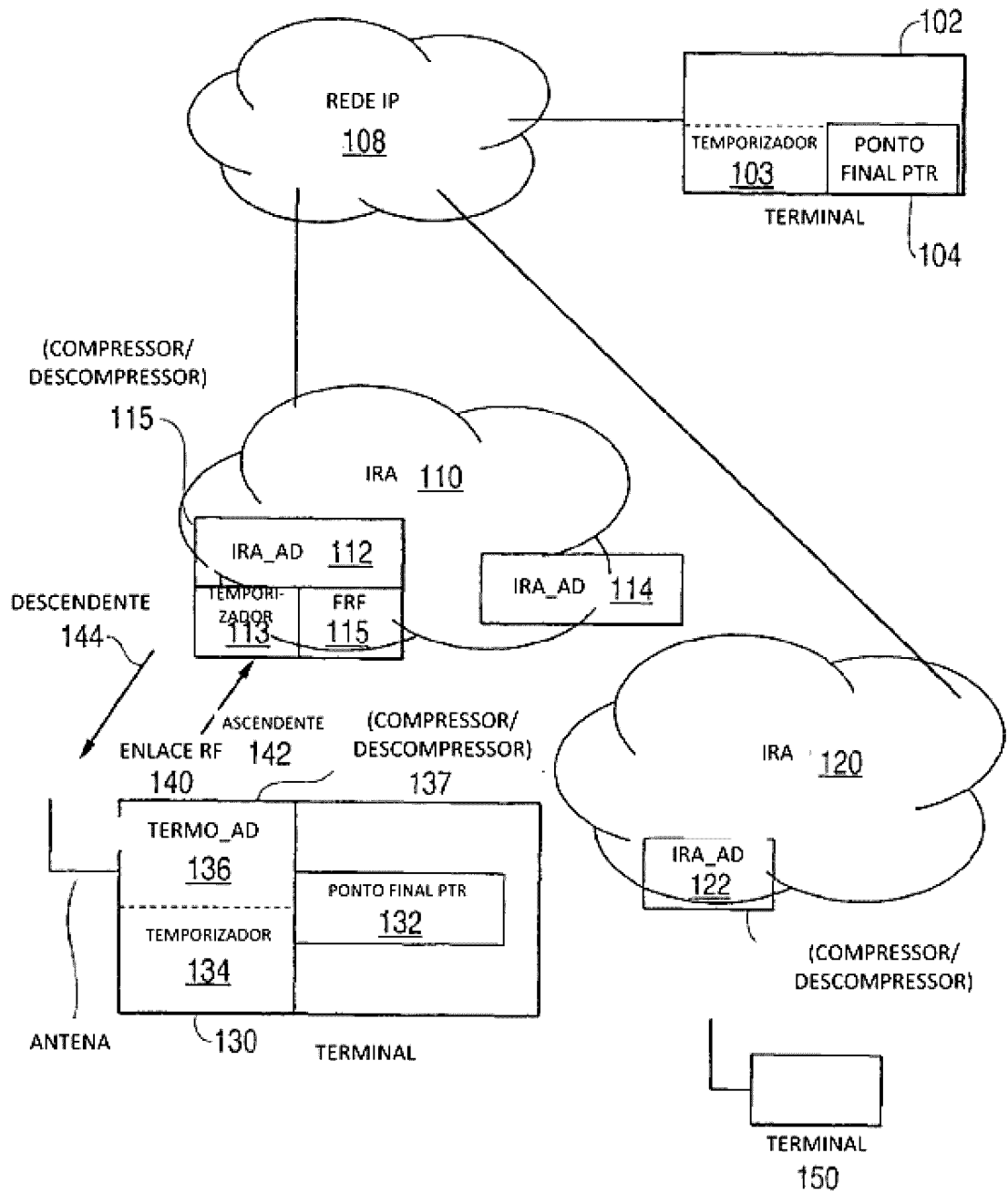


FIGURA 1

FORMATO DE PACOTE PTR

HDR IP <u>210</u>	HDR PDU <u>212</u>	HDR PTR <u>214</u>	CARGA ÚTIL <u>216</u>
----------------------	-----------------------	-----------------------	--------------------------

(EX. AMOSTRA DE VOZ)

FIGURA 2

FORMATO DE CABEÇALHO PTR DESCOMPRIMIDO

PROTOCOLO DE TEMPO (PT) <u>310</u>	NÚMERO SEQUENCIAL (NS) <u>312</u>	OUTROS CAMPOS <u>314</u>
--	--------------------------------------	-----------------------------

FIGURA 3

FORMATO DO CABEÇALHO PTR COMPRIMIDO

TIPO DE MENSAGEM <u>410</u>	MÁSCARA DE BIT <u>412</u>	PROTOCOLO DE TEMPO COMPRIMIDO <u>414</u>	CAMPOS OPCIONAIS <u>416</u>
--------------------------------	------------------------------	---	--------------------------------

FIGURA 4

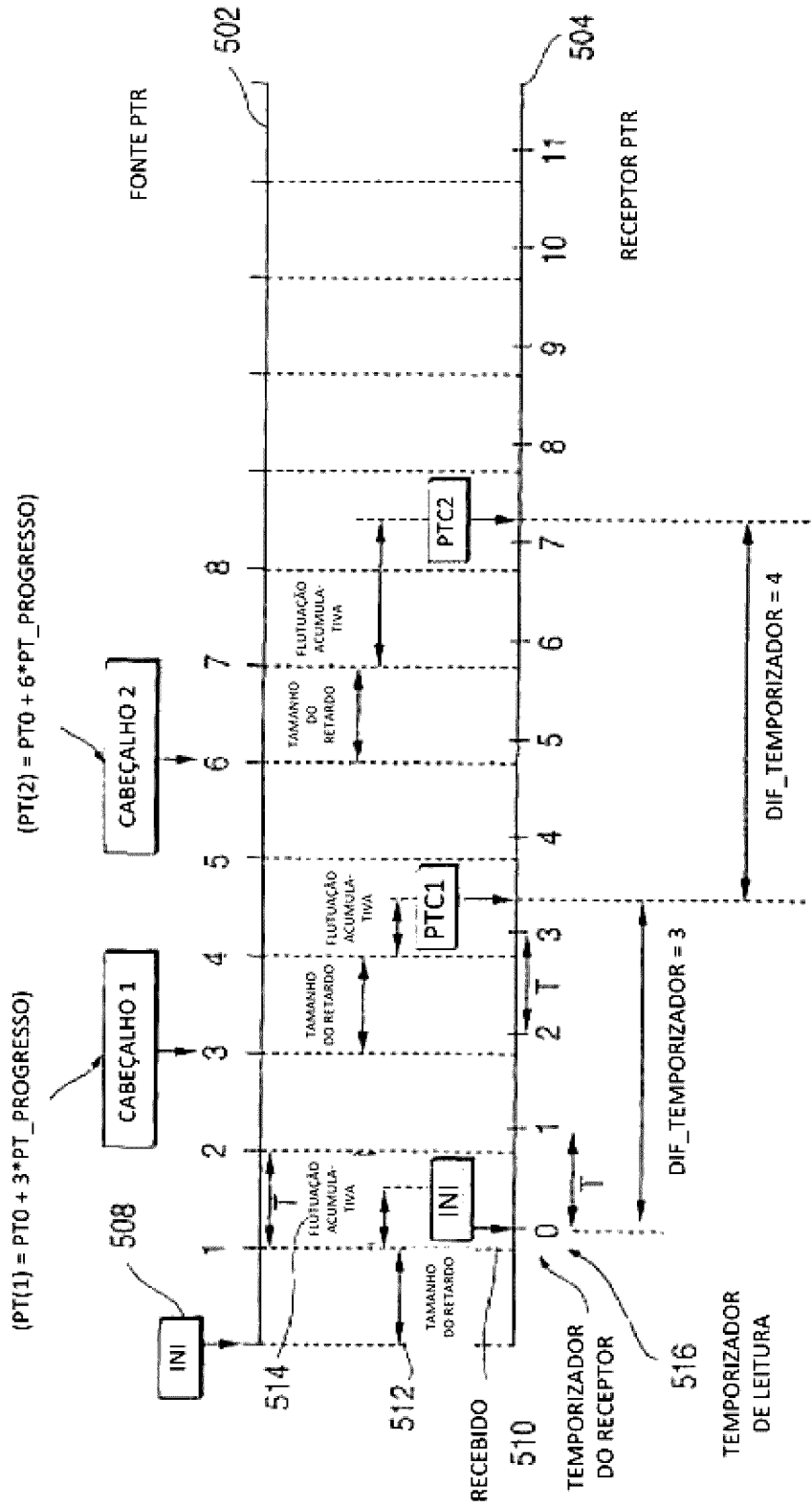


FIGURA 5

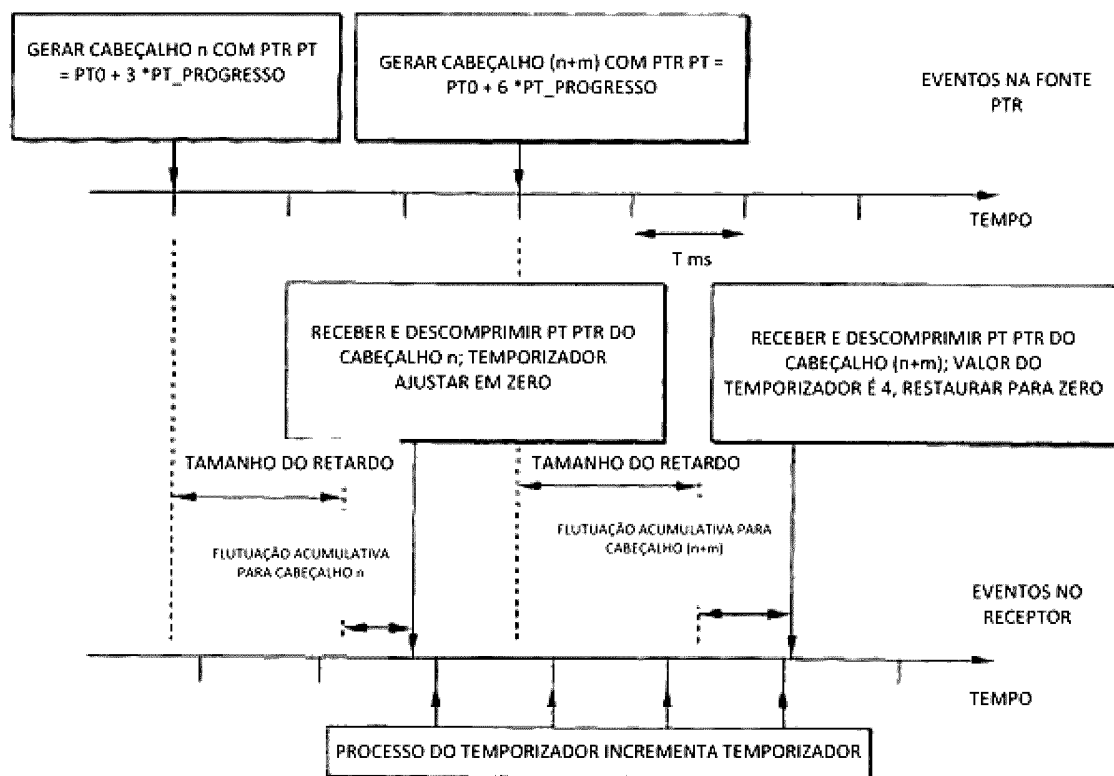


FIGURA 6

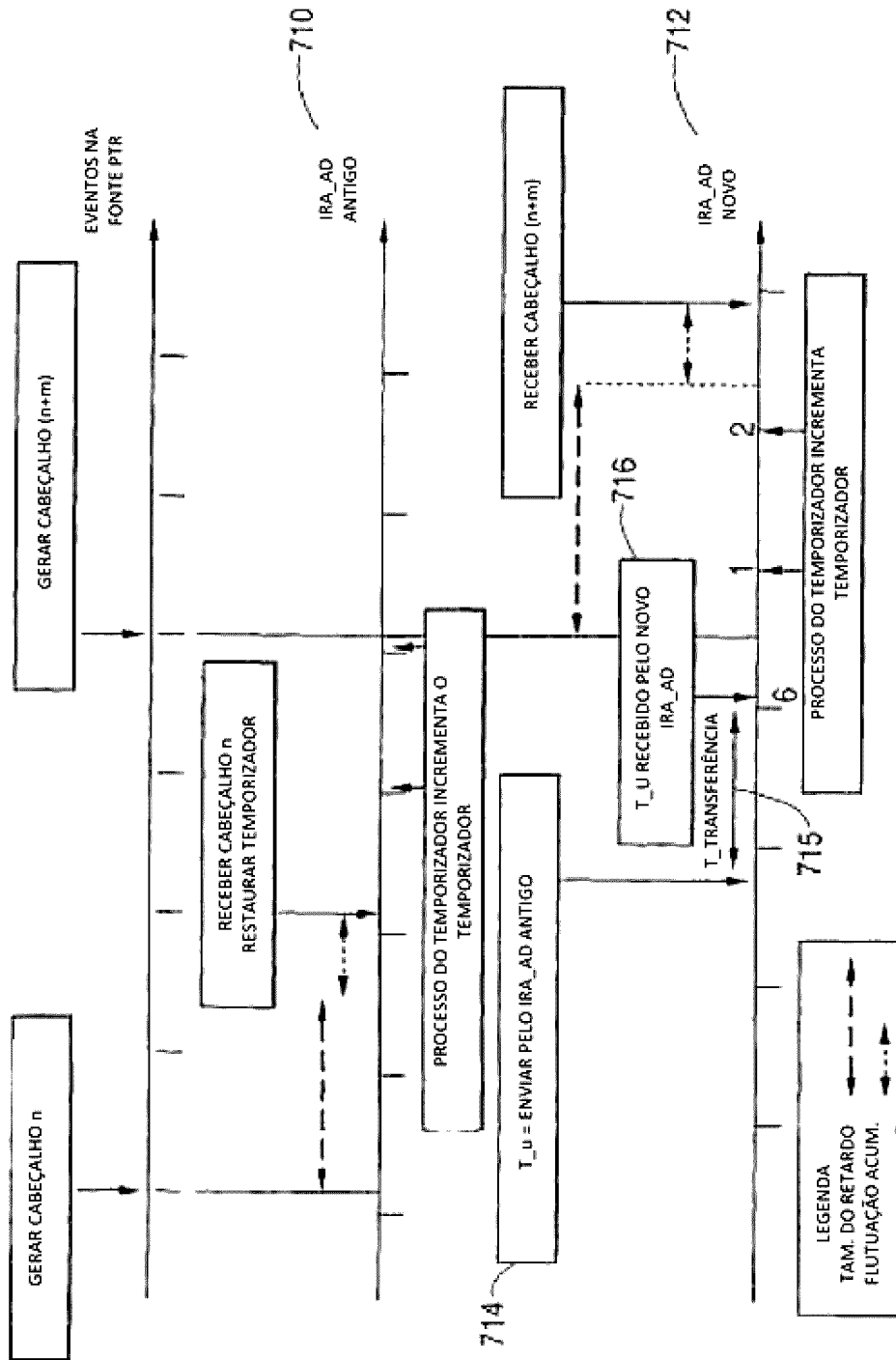


FIGURA 7

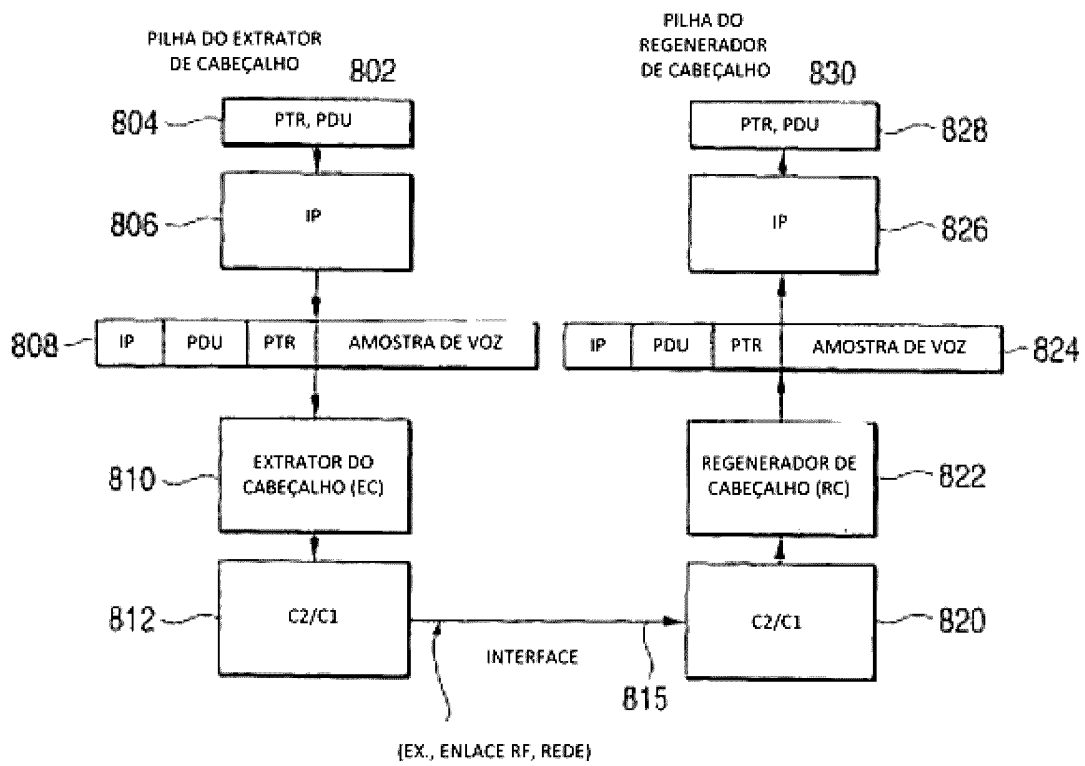


FIGURA 8

INFORMAÇÃO	CONTEÚDO	TAMANHO
ini_info(n)	CABEÇALHO IP/PDU/PTR COMPLETO; n É IMPLICITAMENTE ESPECIFICADO EM NS PTR	COM 40 BYTES = 320 BITS PELO MENOS; ENVIADO NA INTERFACE AÉREA
cadeia_ini(n)	C_NS, C_PT (SE A TEMPORIZAÇÃO NÃO FOR MANTIDA DE UMA CADEIA PARA A PRÓXIMA), p_tamanho (NÃO PROVÁVEL), PT_progresso (NÃO PROVÁVEL); n É IMPLICITAMENTE ESPECIFICADO EM C_NS	COM 8 BITS SE APENAS C_NS, 12 BITS SE C_NS E C_PT
HO_ini_u(n)	CABEÇALHO IP/PDU/PTR COMPLETO, MAS PT PTR SUBSTITUÍDO POR PT0_u, m_último, PT_progresso_u; n É IMPLICITAMENTE ESPECIFICADO EM NS PTR	LIGEIRAMENTE MAIOR QUE O CABEÇALHO COMPLETO; CARREGADO NA REDE CABEADA, ENTRE IRA_AD
HO_ini_d(n)	p_tamnhho_d, E PT_progresso_d, JUNTAMENTE COM O SEU NÚMERO DE GERAÇÃO	O TAMANHO DEPENDE DA CODIFICAÇÃO DE p_tamanho E PT_progresso; CARREGADO NA REDE CABEADA ENTRE IRA_AD
HO_sinc_u(n)	C_NS, C_PT (PROVÁVEL), p_tamanho_u, PT_progresso_u; n É IMPLICITAMENTE ESPECIFICADO EM C_NS	COM 8 BITS SE APENAS C_NS, 12 BITS SE C_NS E C_PT
HO_sinc_d(n)	C_SN, C_TS (PROVÁVEL), p_tamanho_d, PT_progresso_d; n É IMPLICITAMENTE ESPECIFICADO EM C_NS	COM 8 BITS SE APENAS C_NS, 12 BITS SE C_NS E C_PT
rec(n)		POUCOS BITS

FIGURA 9

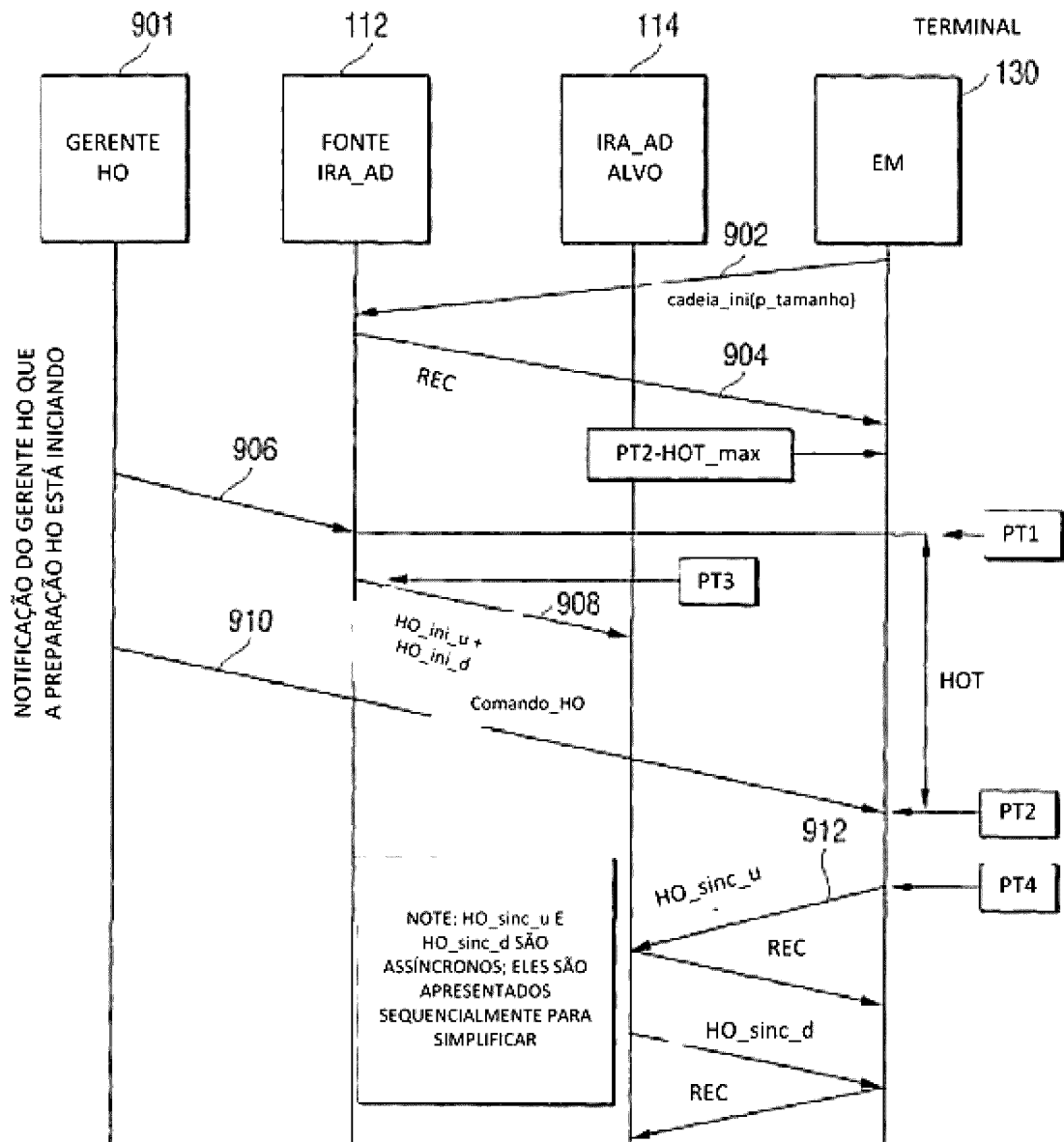


FIGURA 10

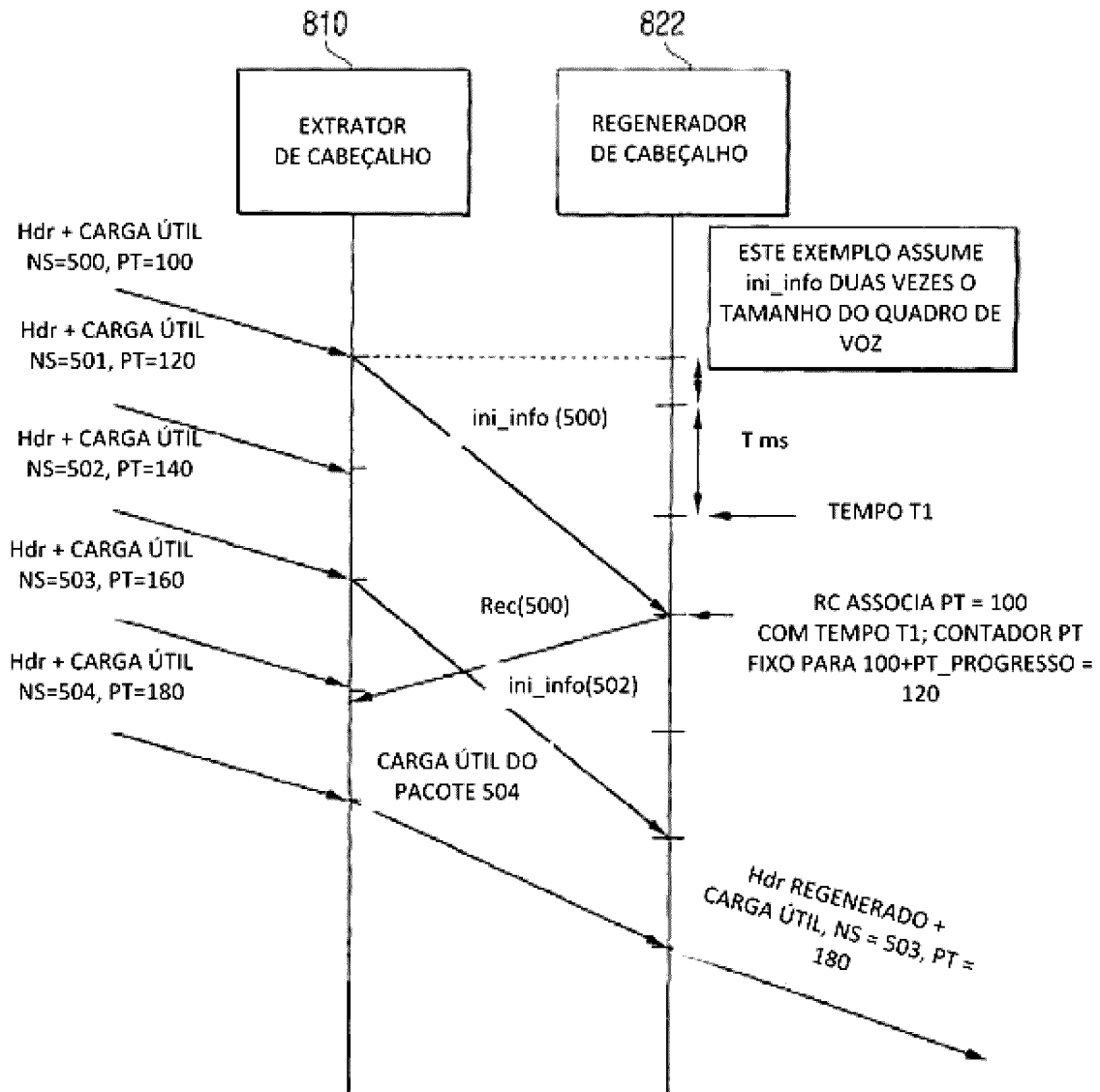


FIGURA 11

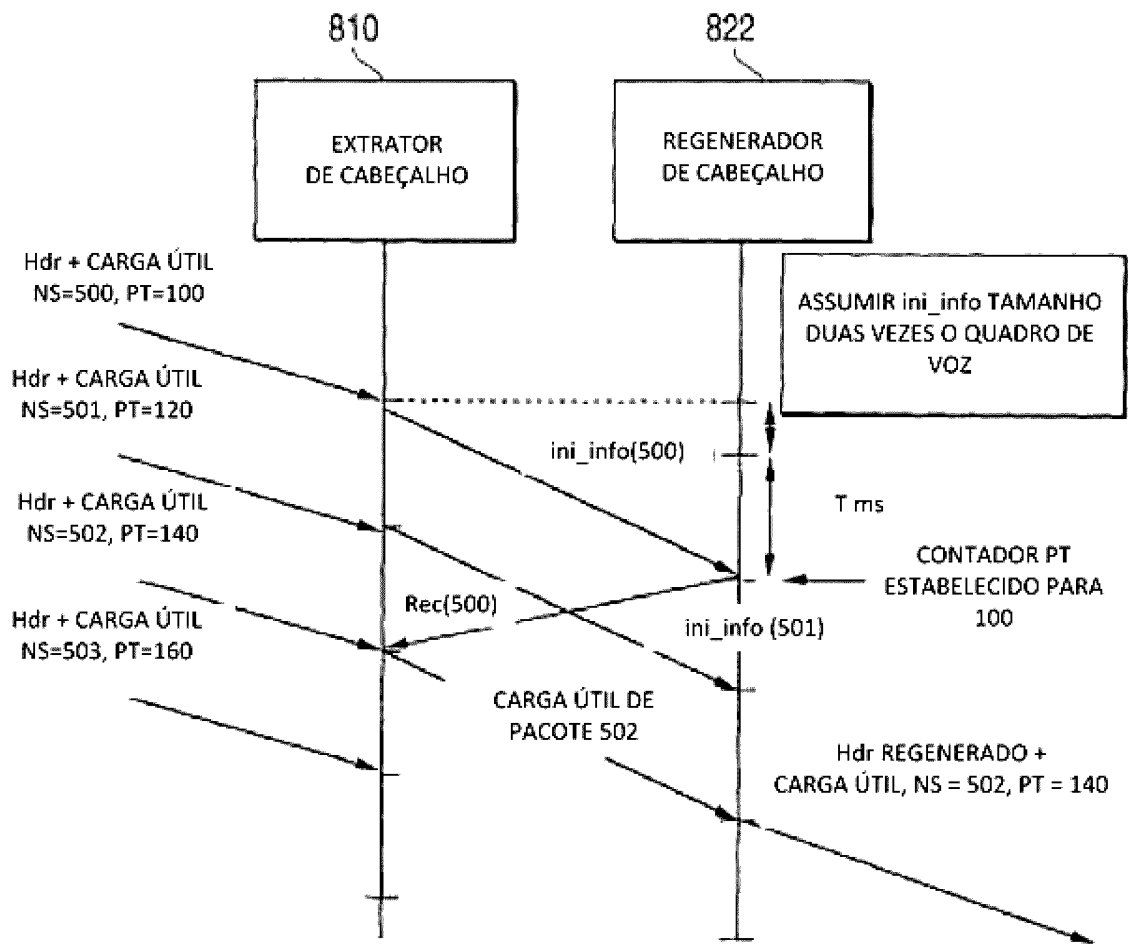
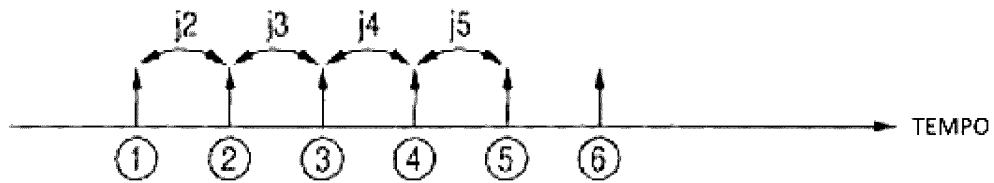


FIGURA 12

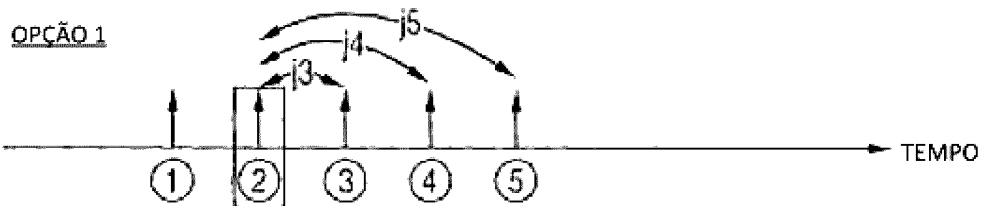


j_2 = FLUTUAÇÃO DO PACOTE 2 COM RELAÇÃO PARA 1
 j_3 = FLUTUAÇÃO DO PACOTE 3 COM RELAÇÃO PARA 2
 j_4 = FLUTUAÇÃO DO PACOTE 4 COM RELAÇÃO PARA 3

j_5 = FLUTUAÇÃO DO PACOTE 5 COM RELAÇÃO PARA 4 ← FEITO NO PACOTE 5

FLUTUAÇÃO DE REDE PARA PACOTE 2 = j_2
 FLUTUAÇÃO DE REDE PARA PACOTE 3 = j_3
 FLUTUAÇÃO DE REDE PARA PACOTE 4 = j_4 , etc

FIGURA 13



PACOTE 2 E A REFERÊNCIA

$N_{FLUTUAÇÃO(3, 2)} = j_3$ = FLUTUAÇÃO DO PACOTE 3 COM RESPEITO A REFERÊNCIA
 ASSUMIR $j_3 = 2$
 $N_{FLUTUAÇÃO(4, 2)} = j_4$ = FLUTUAÇÃO DO PACOTE 4 COM RESPEITO A REFERÊNCIA
 ASSUMIR $j_4 = 3$

$N_{FLUTUAÇÃO(5, 2)} = j_5$ = FLUTUAÇÃO DO PACOTE 5 COM RESPEITO A REFERÊNCIA
 ASSUMIR $j_5 = -1$

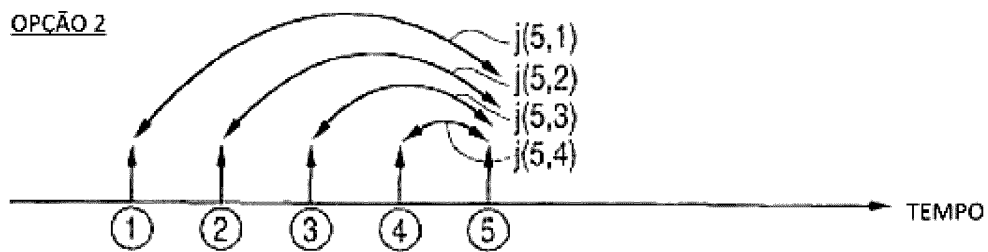
ANTES PAC 5: $N_{FLUTUAÇÃO_MIN} = 2$
 NO PAC 5: $N_{FLUTUAÇÃO_MIN} = -1$

$N_{FLUTUAÇÃO_MÁXIMA} = 3$
 $N_{FLUTUAÇÃO_MÁXIMA} = 3$

$N_{FLUTUAÇÃO_MAX_N_FLUTUAÇÃO_MIN} = 3 - (-1) = 4$

FLUTUAÇÃO DE REDE MÁXIMA = 1 PARA PACOTE 5

FIGURA 14



CALCULAR FLUTUAÇÃO DO PACOTE 5 COM RELAÇÃO PARA 1: $j(5, 1) = 2 = N_FLUTUAÇÃO (5, 1)$
 CALCULAR FLUTUAÇÃO DO PACOTE 5 COM RELAÇÃO PARA 2: $j(5, 2) = 3 = N_FLUTUAÇÃO (5, 2)$
 CALCULAR FLUTUAÇÃO DO PACOTE 5 COM RELAÇÃO PARA 3: $j(5, 3) = 4 = N_FLUTUAÇÃO (5, 3)$
 CALCULAR FLUTUAÇÃO DO PACOTE 5 COM RELAÇÃO PARA 4: $j(5, 4) = 7 = N_FLUTUAÇÃO (5, 4)$
 FLUTUAÇÃO DE REDE MÁXIMA = 7 PARA PACOTE 5

FIGURA 15

RESUMO**"MÉTODO E SISTEMA DE COMPRESSÃO E DESCOMPRESSÃO DE CAMPO DE CABEÇALHO BASEADO NO TEMPORIZADOR EM REDES MÓVEIS COM PROTOCOLO IP".**

A técnica de compressão/descompressão do cabeçalho baseada no temporizador e a técnica baseada no temporizador e na referência são providas. A fonte (102) gera um campo de cabeçalho, tal como o protocolo de tempo PTR. O pacote incluindo o campo de cabeçalho é enviado para o compressor (115), o qual calcula o campo de cabeçalho comprimido baseado no campo de cabeçalho do pacote da fonte (102) e da quantidade de flutuação. O campo de cabeçalho comprimido é calculado ao calcular o efeito de flutuação da rede antes do compressor (115) usado na transmissão dos pacotes e ao calcular o efeito de flutuação da rede entre o compressor (115) e o descompressor (137) usados na transmissão dos pacotes. O pacote incluindo o campo de cabeçalho comprimido é transmitido para o descompressor (137), o qual inclui um temporizador local (134). O descompressor (137) descomprime o campo de cabeçalho comprimido ao calcular a aproximação do campo de cabeçalho baseado no tempo decorrido desde a chegada do pacote prévio e do valor do campo no pacote prévio. A aproximação do campo de cabeçalho é então corrigida baseada no campo de cabeçalho comprimido provido no pacote.