UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

RAFAEL MENDES PEREIRA

Multiplexação Adaptativa Baseada no E-model Para Redução do *Overhead* na Rede em Ligações VoIP Sobre IP *Security* Mantendo Qualidade nas Conversações

Dissertação apresentada como requisito parcial
para a obtenção do grau de
Mestre em Ciência da Computação

Profa. Dra. Liane Margarida Rockenbach Tarouco
Orientadora

Porto Alegre, maio de 2008

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Pereira, Rafael Mendes

Multiplexação Adaptativa Baseada no E-model Para Redução do *Overhead* na Rede em Ligações VoIP Sobre IP *Security* Mantendo Qualidade nas Conversações / Rafael Mendes Pereira. – Porto Alegre: PPGC da UFRGS, 2008.

85 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2008. Orientadora: Liane Margarida Rockenbach Tarouco.

1. VoIP. 2. Segurança. 3. Multiplexação. 4. Modelo E. 5. Qualidade. I. Tarouco, Liane Margarida Rockenbach. II. Título.

AGRADECIMENTOS

Agradeço à Universidade Federal do Rio Grande do Sul, em especial ao CINTED (Centro Interdisciplinar de Novas Tecnologias na Educação), pela infra-estrutura e recursos humanos disponibilizados e à CAPES pelo suporte financeiro que possibilitou a minha dedicação ao mestrado.

Agradeço aos orientadores da minha iniciação científica no LABI (Laboratório de Bioinformática) pelos fundamentais ensinamentos metodológicos e a todos os professores da minha graduação na Universidade Estadual do Oeste do Paraná.

Agradeço também a todas as pessoas importantes que me entornam, em especial a minha família, à Taci, aos amigos e a minha orientadora Liane R. M. Tarouco.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS

3DES *Triple Data Encryption Algorithm*

AAA *Authentication, Authorization and Accounting*

AES *Advanced Encryption Standard*

AH *Authentication Header*

CBC *Cipher Block Chaining*

CRTP *Compressed RTP*

DHCP *Dynamic Host Configuration Protocol*

DES *Data Encryption Algorithm*

DoS *Denied of Service*

DNS *Domain name system*

eCRTP *Enhanced CRTP*

ESP *Encapsulation Security Payload*

HMAC *Keyed-Hashing for Message Authentication*

HTTP *Hypertext Transfer Protocol*

IETF *Internet Engineering Task Force*

ITU-T *International Telecommunication Union*

IPsec *Internet Protocol Security*

QoS *Quality of Service*

MD5 *Message-Digest algorithm 5*

MIME *Multipurpose Internet Mail Extensions*

NTP *Network Time Protocol*

ROHC *Robust Header Compression*

RTP *Real-time Transport Protocol*

RTCP *Real-time Transport Control Protocol*

RTCP XP*RTCP Extended Reports*

S/MIME *Secure/Multipurpose Internet Mail Extensions*

SAP *Session Announcements Protocol*

SDP *Session Description Protocol*

SHA-1 *Secure Hash Algorithm 1*

SIP *Session Initiation Protocol*

SRTP *Secure Real-time Transport Protocol*

TCP *Transmission Control Protocol*

TFTP *Trivial File Transfer Protocol*

TLS *Transport Layer Security*

UA *User Agent*

UAC *User Agent Client*

UAC *User Agent Server*

URI *Uniform Resource Identifier*

VoIP *Voice over Internet Protocol*

VPN *Virtual Private Network*

LISTA DE FIGURAS

LISTA DE TABELAS

RESUMO

A tecnologia de Voz sobre IP (VoIP) traz novas oportunidades como também novos riscos. Desse modo, diversas soluções vêm sendo propostas com o objetivo de assegurar premissas de segurança nas conversações. Contudo, a aplicação de segurança em VoIP apresenta-se como uma tarefa complexa devido às características peculiares dessa tecnologia. Um exemplo consiste na proteção das conversações por meio do IPSec, o qual diminui, substancialmente, o uso efetivo da banda.

A multiplexação das ligações apresenta-se como uma alternativa para melhoria do desempenho da aplicação do IPSec em VoIP. Nessa técnica quanto mais pacotes agrupados, melhor é o resultado obtido. Entretanto, essa técnica mostra-se menos eficaz em cenários com poucas ligações simultâneas.

Assim, nesse trabalho é apresentada uma evolução da técnica de multiplexação, onde propõem-se um modelo adaptativo baseado no E-Model, cujos objetivos consistem em melhorar a redução do *overhead* na rede, respeitando os limites desejáveis de qualidade nas conversações.

Na proposta, em boa conjuntura dos fatores que influenciam o desempenho das ligações, expande-se o tempo de retenção, aumentando o número de pacotes agrupados e conseqüentemente obtendo maior taxa de compressão. O acréscimo é realizado respeitando os limiares de qualidade de cada ligação envolvida na multiplexação. Em situações críticas, quando o aumento do atraso de fala e escuta torna-se degradante, diminui-se o tempo de agrupamento dos pacotes, evitando a deterioração da qualidade das conversações.

Realizaram-se avaliações e os resultados demonstraram que a solução proposta possibilita uma maior taxa de compressão em cenários positivos e obtém um melhor nível de qualidade das ligações em contextos desfavoráveis.

Palavras-chave: VoIP, segurança, multiplexação, modelo E, qualidade.

Adaptive Multiplexing Based on E-model
For Reducing Network Overhead on VoIP Calls Over IP Security
Ensuring Conversation Quality

ABSTRACT

The Voice over IP (VoIP) technology rises new opportunities but also new security risks. Therefore, many solutions have been proposed aiming to commit security premises on conversations. However, applying security on VoIP is a challenging task due to the unique features of this technology. One of the proposals lies in protecting conversations using IPSec, reducing significantly the amount of bandwidth available.

Multiplexing connections seems to be an alternative to improve performance of IPSec on VoIP. On this method, as many package grouped together as higher will be the performance. In spite of that, this method does not get good results with few simultaneous connections scenarios.

On this work is shown an evolution to multiplex method, where is proposed an adaptive model based on E-Model, aiming to reduce network overhead and also guarantying reasonable quality for conversations.

When the connection performance factors are favorable, the retention time is extended, increasing the number of packages being grouped, thus allowing a higher compression ratio. This increase is done respecting quality thresholds of each connection involved. In critical situations, when the connection delay is prominent, the retention time is reduced, avoiding more loss of connection quality.

The results obtained demonstrate an improvement on compression ratio in positive scenarios and a better quality level on connections, considering unfavorably scenarios.

Palavras-chave: VoIP, security, multiplexing, E-model, quality.

1 INTRODUÇÃO

Sistemas de VoIP (*Voice over Internet Protocol*) realizam a transmissão de voz sobre o protocolo IP possibilitando a conversação por meio da Internet. A telefonia sobre a Internet apresenta-se como uma alternativa ao sistema telefônico convencional. Essa nova tecnologia agrega às conferências novas possibilidades, tais como a transmissão de vídeo e o compartilhamento de dados. Adicionalmente, outras vantagens desse serviço correspondem ao seu baixo custo e a sua flexibilidade em comparação ao sistema de comunicação tradicional (TANENBAUM, 2003).

A utilização de VoIP introduz novas oportunidades como também novos riscos, pois a transmissão da voz na Internet depende de diversos componentes e parâmetros de configuração. Desse modo, invasores têm um amplo número de potencias pontos vulneráveis para atacar (KUHN; WALSH; FRIES, 2005). As vulnerabilidades existentes em VoIP podem estar relacionadas à sinalização das conferências como também à transmissão da mídia. Assim, ataques indevidos podem afetar a privacidade dos participantes, a integridade dos dados e a disponibilidade do serviço (VOIPSA, 2005).

Em relação aos abusos de privacidade, os riscos de escutas indevidas, já existentes no sistema de telefonia convencional, aumentam com a utilização de VoIP. Isso ocorre devido à estrutura da Internet, onde as informações trafegam por diversos dispositivos sem o controle direto dos pontos finais. Esses riscos possibilitam, além da escuta indevida do tráfico de voz, a interceptação e a obtenção de dados como senhas, identificação de usuários, números telefônicos privados, entre outros (VOIPSA, 2005).

A integridade da mídia e dos dados transmitidos também se apresenta como uma preocupação de segurança em VoIP. Alterações intencionais na transmissão são alguns exemplos de riscos que podem afetar a integridade dessa tecnologia (VOIPSA, 2005).

O risco mais evidente da telefonia sobre a Internet corresponde a negação de serviço. Os sistemas de VoIP são altamente sensíveis ao atraso, desse modo, a degradação na transmissão pode acarretar muitos problemas na disponibilidade do serviço (WALSH; KUHN, 2005).

Como ocorrido historicamente com as demais tecnologias, a preocupação com segurança em VoIP tem aumentado após o seu sucesso e crescimento (VOIPSA, 2005). Assim, diversos esforços vêm sendo realizados para evitar vulnerabilidades nos sistemas de VoIP. Entre eles, a aliança VoIPSA (VoIP *Security Alliance*) tem o objetivo de descobrir e reduzir os riscos de segurança em VoIP. Ela foi constituída em fevereiro de 2005 por diversos segmentos da industria, tais como 3Com, *Columbia University*, CISCO, Siemens, Symantec, entre outras (VOIPSA, 2005). Além disso, os comitês da ITU-T (*International Telecommunication Union*) e os grupos de trabalhos do IETF (*Internet Engineering Task Force*) definiram algumas formalizações de segurança nas arquiteturas de conferência multimídias pela Internet.

Para a arquitetura H.323 (INTERNATIONAL TELECOMMUNICATION UNION, 2000) foi definida a recomendação H.235 (INTERNATIONAL TELECOMMUNICATION UNION, 2003a), a qual apresenta diversos procedimentos, mensagens, estruturas e algoritmos para prover autenticação, privacidade e integridade na sinalização, no controle e na comunicação de mídia nas conferências (INTERNATIONAL TELECOMMUNICATION UNION, 2003a). A obtenção dos mesmos requisitos é desejável com a aplicação de alguns mecanismos de segurança na arquitetura do IETF, tais como a utilização do método de autenticação *Digest* (FRANKS et al., 1999) e de S/MIME (*Secure/Multipurpose Internet Mail Extensions*) (RAMSDELL, 2004) nas mensagens do protocolo de sinalização SIP (*Session Initiation Protocol*) (ROSENBERG et al., 2002), e a definição do protocolo SRTP (*Secure Real-time Transport Protocol*) (SCHULZRINNE et al., 2003) para a transmissão segura da mídia.

Adicionalmente, é proposta a utilização de mecanismos em níveis mais baixos na pilha de protocolos ((KUHN; WALSH; FRIES, 2005). Alguns exemplos consistem no uso dos protocolos TLS (*Transport Layer Security*) (BLAKE-WILSON et al., 2006) e IPsec (*Internet Protocol Security*) (KENT; ATKINSON, 1998a).

Apesar da existência de várias propostas, a aplicação de segurança em VoIP consiste em uma tarefa complexa (BARBIERI; BRUSCHI; ROSTI, 2002). Isso se deve à existência de vários componentes e protocolos, ao balanceamento entre segurança e baixa complexidade e, principalmente, ao impacto desses mecanismos de segurança na qualidade do serviço (WALSH; KUHN, 2005).

Cita-se como exemplo dessa dificuldade a expansão dos pacotes com a aplicação do IPSec. Esse protocolo é amplamente utilizado na construção de canais privados, como VPNs (*Virtual Private Network*) (KUHN; WALSH; FRIES, 2005). Com o IPSec é possível aplicar mecanismos para se obter privacidade, integridade, verificação de autoria e proteção contra ataques de negação de serviço (KENT; ATKINSON, 1998a). Contudo, a necessidade da inserção de informações de segurança agrava a já baixa performance de ocupação da banda em VoIP, aumentando o *overhead* na rede (BARBIERI; BRUSCHI; ROSTI, 2002).

Em aplicações de VoIP, devido à necessidade de tempo real, são usados *codecs* com baixo tempo de amostragem e alta taxa de compressão, o que resulta em pacotes com *payloads* pequenos em relação aos cabeçalhos necessários para sua transmissão. Desse modo, com a inserção dos cabeçalhos de segurança do IPSec, essa relação é agravada, deteriorando o uso da banda, o que pode diminuir em até 40% a capacidade de ligações simultâneas na rede (BARBIERI; BRUSCHI; ROSTI, 2002).

Para esse comportamento específico são sugeridas soluções objetivando diminuir o envio dos dados redundantes na rede. Basicamente, existem duas abordagens: compressão dos cabeçalhos e multiplexação dos pacotes. A primeira mantém informações entre as duas pontas envolvidas, enviando somente os dados que diferem do contexto mantido (BARBIERI; BRUSCHI; ROSTI, 2002).

Na técnica de multiplexação, pacotes de diversas ligações são agrupados em um único *payload*, compartilhando cabeçalhos adjacentes. Essas duas técnicas podem ser aplicadas em conjunto (SZE et al., 2002).

A performance da multiplexação está relacionada ao número de pacotes agrupados, ou seja, quanto mais pacotes agrupados, menor será o número adicional de cabeçalhos enviados (SZE et al., 2002). Para se alcançar um número ideal de pacotes agrupados e com isso melhorar o desempenho da aplicação do IPSec em VoIP, é necessário um contexto com um número grande de ligações concomitantes (acima de 100 ligações). Contudo,

existem cenários onde esses números não podem ser alcançados. Isso pode ocorrer em ambientes com restrições de números de usuários presentes, ou com baixa capacidade de roteamente dos equipamentos utilizados, que não são capazes de gerir um número grande de ligações simultâneas.

Desse modo, nesses ambientes pode-se tornar desencorajador a aplicação da multiplexação em VoIPSec (*Voice over IPSec*). Assim, surge como alternativa para melhoria do desempenho dessa técnica nesses cenários a possibilidade da expansão do tempo em que o processo de multiplexação aguarda e agrupa os pacotes para formação do pacote multiplexado. Com isso, mais pacotes serão agrupados, mesmo com um número reduzido de ligações simultâneas ativas.

Por sua vez, é importante notar que essa solução esbarra nos altos critérios de qualidade nas conversações em VoIP. Isso se deve ao fato da expansão desse tempo pode aumentar o atraso fim-a-fim das ligações envolvidas, o que pode deteriorar a qualidade das ligações (INTERNATIONAL TELECOMMUNICATION UNION, 2003b).

Desse modo, apresenta-se a seguinte problemática: A possibilidade de aumentar a performance da técnica de multiplexação aplicado em VoIPSec em cenários com poucas ligações esbarra na necessidade de manter a qualidade das ligações.

Baseada nessa problemática, é apresentada nesta dissertação uma proposta de solução para evolução do modo atual de aplicação da técnica de multiplexação com o objetivo de melhorar a sua performance na aplicação em VoIPSec em cenários restritos (com baixo número de ligações simultâneas e com contextos diversos de qualidade) mantendo-se a qualidade das ligações envolvidas.

O texto está organizado da seguinte maneira: no Capítulo 2 é apresentada a base teórica desse trabalho, no Capítulo 3 mostra-se com mais detalhes a problemática abordada, em seguida no Capítulo 4, é apresentada a solução de melhoria proposta. O ambiente para validação dessa proposta é mostrado no Capítulo 5. Os resultados e a discussão são apresentados no Capítulo 6. Por fim, as considerações finais e propostas de trabalhos futuros são descritas no Capítulo 7.

2 BASE TEÓRICA

2.1 Considerações Iniciais

Apesar da Internet não ter sido projetada para a transmissão de dados de aplicações em tempo real, o aumento do poder de processamento dos computadores modernos e da largura de banda, estimulou o desenvolvimento de aplicações de multimídia (TANENBAUM, 2003).

Dentre essas soluções está a tecnologia de Voz sobre IP (VoIP - *Voice over Internet Protocol*), que permite ligações telefônicas sobre a Internet. Essa solução vem se apresentando como uma alternativa flexível e de baixo custo em relação à telefonia convencional (TANENBAUM, 2003).

A arquitetura VoIP possibilita que sejam agregadas às conversações novas funcionalidades, tal como a possibilidade dos usuários poderem ser encontrados em qualquer ponto na rede, ou a oportunidade do compartilhamento de dados entre os participantes (TANENBAUM, 2003).

Apesar dessas vantagens, o sucesso dessa tecnologia depende de fatores como a qualidade do serviço e a disponibilidade de banda (BARBIERI; BRUSCHI; ROSTI, 2002). Adicionalmente, um requisito emergente consiste na aplicação de mecanismos que garantam a segurança das conversações (WALSH; KUHN, 2005).

Nas próximas sessões serão apresentados o conteúdo teórico como base para esse trabalho. Serão descritos protocolos da arquitetura VoIP (Seção 2.2), os fatores relacionados com a qualidade do serviço (Seção 2.3) e um modo de estimativa de qualidade das conversações (Seção 2.4). Também serão descritos alguns exemplos de riscos e vulnerabilidades de segurança em VoIP (Seção 2.5), mecanismos de segurança para essa arquitetura (Seção 2.6), assim também como a degradação da aplicação dessas soluções em VoIP (Seção 2.7) e, por fim, alternativas de melhoria da performance da segurança em VoIP (Seção 2.8).

2.2 Visão geral sobre VoIP

Os fatores determinantes para uma arquitetura de conferência são as comunicações entre grupos de pessoas e a entrega de informações em tempo real (HANDLEY et al., 2000). As principais arquiteturas de conferência multimídia pela Internet são as recomendadas pelo ITU - *Internet Telecommunication Union* e pelo IETF - *Internet Engineering Task Force* (TANENBAUM, 2003).

A recomendação H.323, feita pelo ITU, faz referências a um conjunto de protocolos específicos para configuração de chamadas, sinalização, codificação, transmissão de dados, entre outros. Essa arquitetura é amplamente utilizada devido a sua interoperabilidade com a rede de telefonia pública comutada. Entretanto, para permitir essa interoperabili-

dade, o H.323 apresenta-se como uma arquitetura complexa e rígida, tornando-se difícil a sua adaptação em aplicações futuras (TANENBAUM, 2003).

Devido a essas características, o IETF estabeleceu um comitê com o objetivo de projetar uma arquitetura mais simples e modular para as conferências de mídia sobre a Internet (Figura 2.1) (TANENBAUM, 2003).

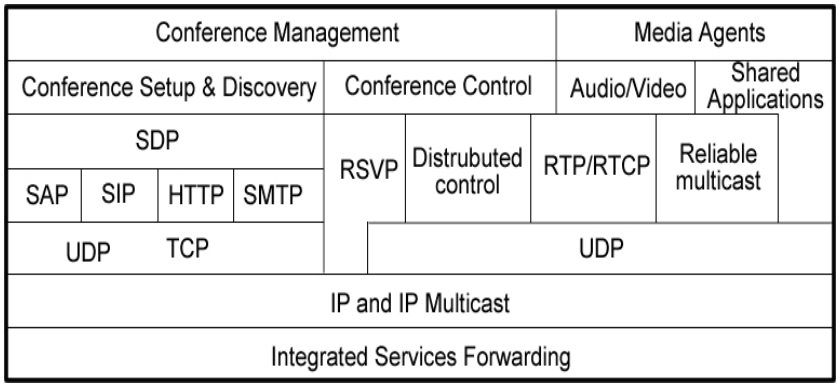

Figura 2.1: Pilha de Protocolos da Arquitetura de Conferência Multimídia. Adaptado de (HANDLEY et al., 1997).

O principal protocolo dessa arquitetura é o SIP (*Session Initiation Protocol*), o qual realiza o controle, no nível de aplicação, de sessões entre um ou mais participantes. Definiram-se também protocolos para o anúncio (SAP - *Session Announcements Protocol*) (HANDLEY; PERKINS; WHELAN, 2000) e a descrição (SDP - *Session Description Protocol*) (HANDLEY; JACOBSON, 1998) das sessões, entre outros.

Em ambas arquiteturas utiliza-se os protocolos RTP (*Real-time Protocol*) e RTCP (*Real-time Control Protocol*) para a transmissão e envio de informações estatísticas da mídia das sessões, respectivamente.

Nas próximas sessões serão apresentados detalhes dos principais protocolos para o funcionamento da arquitetura VoIP. Enfocou-se na arquitetura proposta pelo IETF, a qual apresenta-se com ampla expansão de aplicabilidade (TANENBAUM, 2003).

2.2.1 SIP - *Session Initiation Protocol*

O SIP é um protocolo de aplicação que pode estabelecer, modificar e terminar sessões (conferência) multimídia, tais como, ligações telefônicas sobre a Internet. Além disso, permite outras funcionalidades como mensagens instantâneas (ROSENBERG et al., 2002). Pode-se estabelecer sessões de duas partes (ligações telefônicas comuns), sessões de várias partes (onde todos podem ouvir e falar) e sessões de multidifusão (com um transmissor e vários receptores). Essas sessões podem conter áudio, vídeo e outros tipos de dados (TANENBAUM, 2003).

Esse protocolo não é baseado em serviços, mas provê primitivas que podem ser utilizadas para implementar diversos serviços. Entre eles estão:

- Localização do chamado - localização do terminal para estabelecimento de comunicação;

- Disponibilidade do chamado - determinação da disposição do usuário em estabelecer uma sessão de comunicação;

- Recursos do usuário - definição da mídia e dos parâmetros a serem utilizados na sessão;
- Configurações da sessão - estabelecimento de parâmetros da sessão entre os participantes;
- Gestão da sessão - possibilidade de transferir, colocar em modo de espera ou finalizar sessões, assim como, modificar os seus parâmetros e invocar serviços.

Esses serviços são utilizados apenas para o gerenciamento, configuração e encerramento das sessões. Desse modo, para construir uma arquitetura completa de multimídia, freqüentemente são utilizados outros protocolos, tais como, RTP/RTCP, para o transporte de dados, e o SDP, para a descrição das sessões multimídias (ROSENBERG et al., 2002).

2.2.1.1 Componentes SIP

O protocolo SIP é baseado na arquitetura Cliente/Servidor, sendo composta pelas seguintes entidades (ROSENBERG et al., 2002):

- Agentes de usuário (UA):
  - *User Agent Client* (UAC) - entidade lógica responsável por gerar requisições SIP e receber respostas a esses pedidos;
  - *User Agent Server* (UAS) - entidade lógica responsável por gerar respostas às requisições SIP.
- Servidores:
  - *Proxy Server* - entidade intermediária que pode agir como servidor ou cliente com o propósito de estabelecer chamadas entre os utilizadores. A sua tarefa compreende o encaminhamento dos pedidos recebidos para as entidades que fazem parte do caminho até o destino;
  - *Registrar Server* - servidor que aceita requisições de registros de usuários e armazena informações sobre esses pedidos. Com isso, oferece-se um serviço de localização e tradução de endereços de domínio que controla;
  - *Redirect Server* - consiste em um UAS que gera respostas de redirecionamento com o endereço do usuário requisitado.

2.2.1.2 Mensagens SIP

O SIP é um protocolo de texto baseado no modelo de transação requisição/resposta do protocolo HTTP - *Hypertext Transfer Protocol*. Os *User Agents Client* fazem as requisições e os *User Agents Server* retornam respostas a esses pedidos.

Uma mensagem consiste em um nome de método na primeira linha, seguida de um ou mais cabeçalhos (headers), uma linha vazia indicando o fim dos cabeçalhos e o corpo da mensagem que é opcional (ROSENBERG et al., 2002).

Os cabeçalhos transportam parâmetros necessários para as entidades SIP poderem processar as requisições e respostas. Muitos dos cabeçalhos são tirados do MIME - *Multipurpose Internet Mail Extensions*, permitindo ao SIP interoperar com outras aplicações Internet (TANENBAUM, 2003).

Algumas das mensagens definidas pelo protocolo são:

- INVITE - Solicita o início de uma sessão.
- ACK - Sinaliza o recebimento da resposta de aceitação ou de erro de uma mensagem INVITE;
- BYE - Solicita a finalização da sessão;
- OPTIONS - Consulta opções e capacidades de um host;
- CANCEL - Cancela uma requisição pendente;
- REGISTER - Registra em um servidor Register a localização atual do usuário;
- REFER - Referencia o endereço de outro usuário.

Existem outras sinalizações nas extensões SIP, as quais acrescentam funcionalidades ao protocolo.

As respostas no SIP são códigos numéricos divididos em seis classes:

- 1xx - *Provisional* ou *Informational*: Requisição em processo, mas ainda não finalizada;
- 2xx - *Success*: Requisição foi completada com sucesso;
- 3xx - *Redirection*: Requisição será tentada em outra localização;
- 4xx - *Client Error*: Pedido não foi completado por erro na requisição;
- 5xx - *Server Error*: Pedido não foi completado por erro no recipiente.
- 6xx - *Global Failure*: Falha na requisição, não é possível tentar novamente.

2.2.1.3 Endereço SIP

A identificação de uma entidade final na arquitetura SIP é realizada por meio de um endereço chamado SIP URI - *Uniform Resource Identifier*. Esse endereço utiliza um formato equivalente a um endereço de e-mail comum, tais como: sip:utilizador@dominio, sip:utilizador@IP-*address*, sip:numero-telefone@*gateway* ou sips:utilizador@domínio para comunicações seguras (SIPS URI). A primeira parte do SIP ou SIPS URI está associada ao utilizador, serviço ou número de telefone e a segunda parte consiste no domínio ou endereço de rede (ROSENBERG et al., 2002).

Em geral, o processo de tradução de endereços envolve múltiplos passos, como DNS SRV *lookup*, ENUM *lookup* e *Location server lookup* e várias mensagem SIP. Cada *proxy* consulta um *Location server*, modifica o *Request*-URI e o encaminha para o próximo *hop*. Esse processo se repete até que o *request* seja entregue ao destino. Usualmente esses passos são realizados somente no estabelecimento da sessão. Os endereços traduzidos são armazenados para serem utilizados nas demais requisições posteriormente (ROSENBERG et al., 2002).

2.2.1.4 Funções SIP

A maioria das funções SIP envolve configurações das sessões ou ocorrem durante o seu estabelecimento.

Para o estabelecimento das sessões o SIP utiliza uma requisição INVITE para iniciar uma sessão entre dois *user agents*. A mensagem INVITE usualmente contém um corpo que descreve o tipo de sessão que o agente de usuário deseja estabelecer (ROSENBERG et al., 2002).

A conexão é efetivada, como pode-se observar na Figura 2.2, aplicando um *handshake* de três vias. Primeiramente, o usuário chamador envia um INVITE *request* sinalizando que deseja iniciar uma sessão. Caso o usuário chamado aceite participar, é enviada como resposta a mensagem OK (200). Por fim, o protocolo é finalizado com a confirmação ACK (TANENBAUM, 2003).

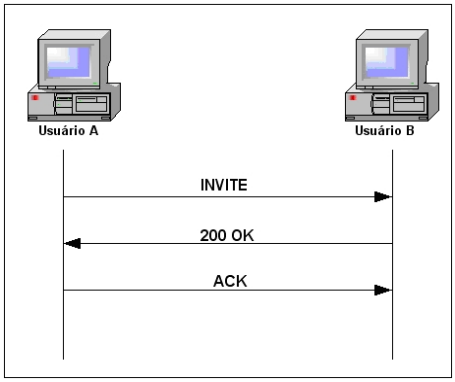

Figura 2.2: Exemplo do Estabelecimento de Sessão (ROSENBERG et al., 2002).

Quando o chamador não conhece o endereço da localização do chamado, a requisição INVITE é direcionada inicialmente a um servidor de *proxy* ou de redirecionamento.

Nesse cenário, ao receber a mensagem INVITE, o servidor consulta o serviço de localização para encontrar o endereço específico do chamado. Assim, quando utiliza-se um servidor de redirecionamento, esse endereço é repassado ao chamador para que ele possa estabelecer a sessão (Figura 2.3).

Quando um *proxy* é utilizado, o servidor atua como uma ponte entre os usuários para as mensagens subseqüentes no estabelecimento da sessão (Figura 2.4).

A negociação de mídia faz parte do estabelecimento de uma sessão entre dois usuários. O SIP não provê mecanismos para a negociação de mídia, mas a negociação é realizada com o uso do protocolo SDP.

Uma vez estabelecida uma sessão, ela pode ser modificada com outra seqüência INVITE/ 200/ACK - referenciada como re-INVITE. O re-INVITE pode realizar modificações nas características da mídia, incluindo o tipo da sessão, *codec* utilizado, endereço IP e o número da porta. Em caso de falha ou recusa no re-INVITE, os parâmetros originais permanecem iguais até o envio da mensagem BYE.

A Finalização e o Cancelamento de sessão são duas operações distintas do SIP. A finalização ocorre quando um agente de usuário, que deseja sair de uma sessão, envia uma mensagem BYE após o seu estabelecimento com sucesso (INVITE/200/ACK).

A mensagem CANCEL é utilizada quando se deseja cancelar o processo de estabelecimento de uma sessão, antes de ser completado. Nesse cenário, um agente de usuário que enviou uma mensagem INVITE, mas que ainda não recebeu a resposta final (2xx,

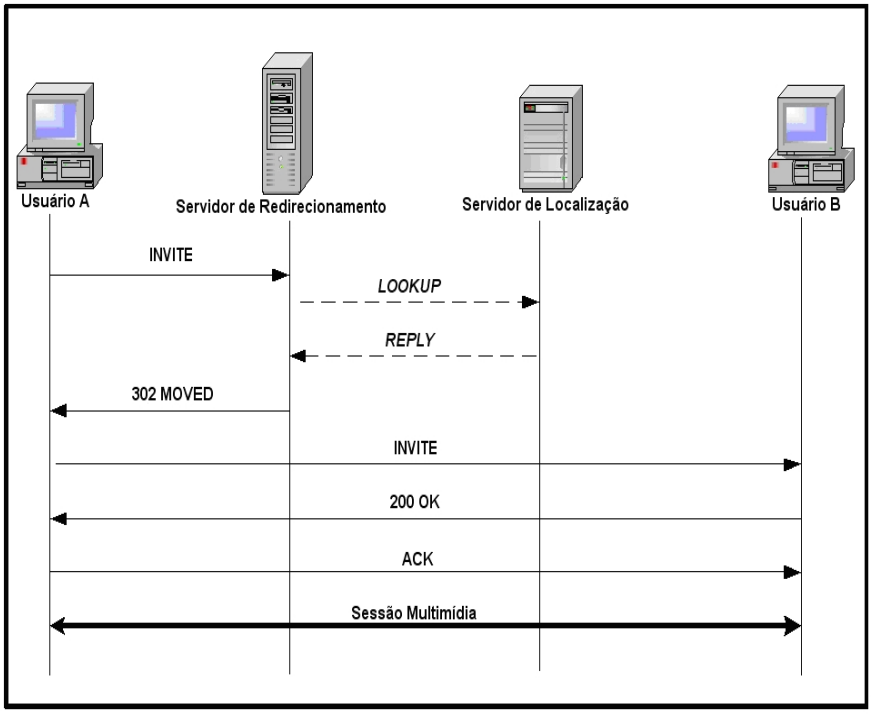

Figura 2.3: Exemplo de Servidor de Redirecionamento (ROSENBERG et al., 2002).

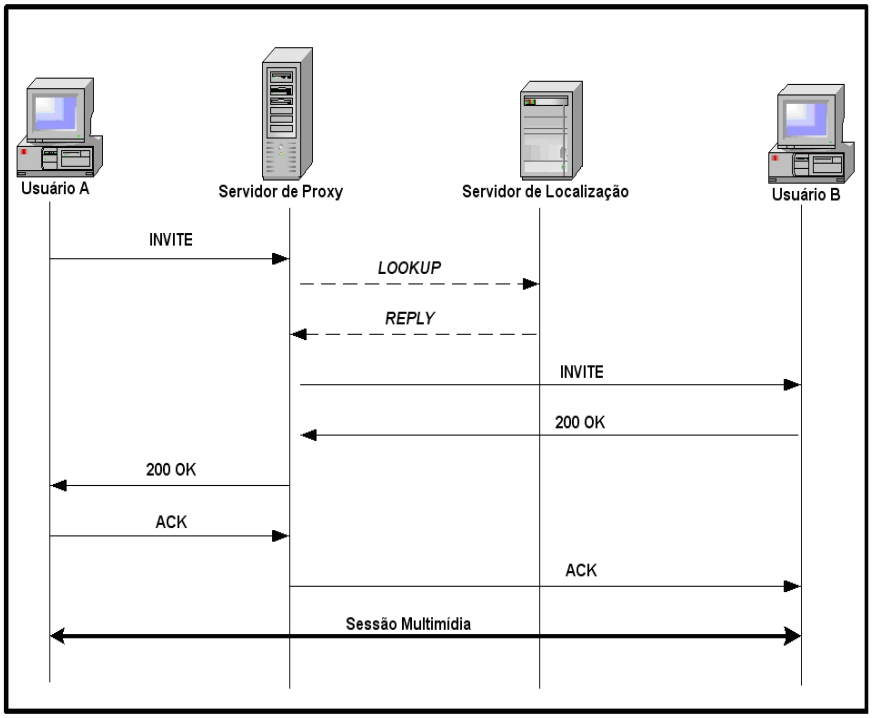

Figura 2.4: Exemplo de Servidor de *Proxy* (ROSENBERG et al., 2002).

3xx, 4xx, 5xx ou 6xx), envia uma mensagem CANCEL solicitando o cancelamento do convite (ROSENBERG et al., 2002).

2.2.2 RTP - Real-time Transport Protocol

O RTP é um protocolo que provê serviços de transporte fim-a-fim de dados em tempo real, tais como áudio, vídeo, texto e outros tipos de dados. Entre os seus serviços estão: identificação do tipo de mídia transportada, numeração dos pacotes, marcação de tempo e monitoramento da qualidade de transmissão. Esses serviços são utilizados em aplicações de tempo real, como telefonia na Internet e videoconferências (SCHULZRINNE et al., 2003).

O protocolo funciona no nível de usuário e normalmente é empregado com o protocolo de transporte UDP - *User Datagram Protocol* (Figura 2.5-a), multiplexando diversos fluxos de dados de tempo real sobre um único fluxo de pacotes UDP (Figura 2.5-b). Esse fluxo pode ser enviado a um único destino (unidifusão), ou, se suportado pela rede, a vários destinos (multidifusão) (TANENBAUM, 2003).

O UDP é mais utilizado nesse tipo de aplicação, pois é preferida a entrega imediata dos pacotes ao atraso decorrente dos serviços providos do protocolo TCP - *Transmission Control Protocol*. O protocolo RTP não contém controle de erros, confirmação e mecanismo para qualidade de serviço, deixando essa responsabilidade às camadas adjacentes (TANENBAUM, 2003).

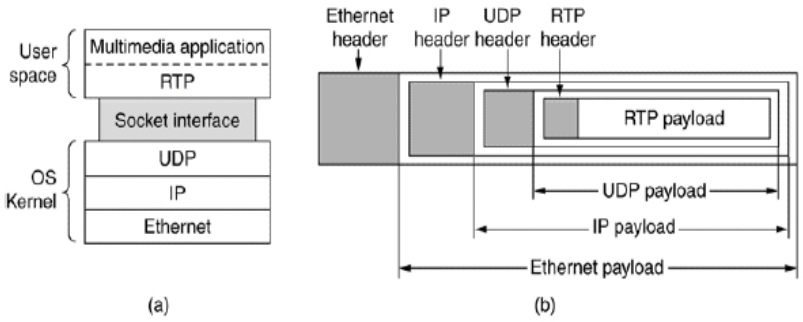

Figura 2.5: (a) Posição do RTP na pilha de protocolos. (b) Alinhamento de pacotes (TANENBAUM, 2003).

O cabeçalho do RTP consiste em três palavras de 32 bits e, potencialmente, algumas extensões para aplicações que necessitam de funcionalidades adicionais. Ele é seguido pelo campo de carga útil, o qual tem o tamanho máximo definido de acordo com o limite dos protocolos das camadas subjacentes (UDP ou TCP). A Figura 2.6 apresenta o formato do pacote RTP, cujos campos são descritos a seguir (SCHULZRINNE et al., 2003):

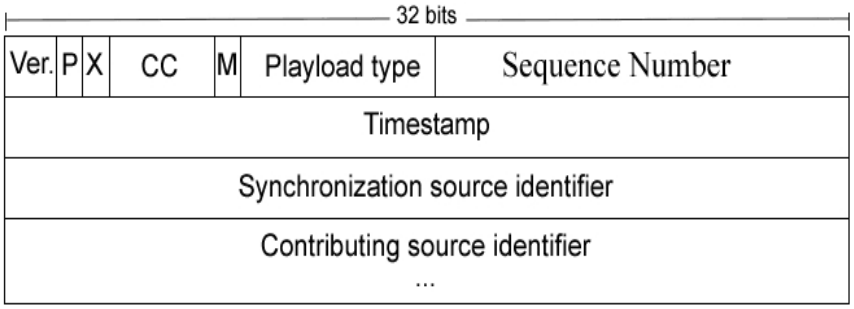

Figura 2.6: Formato do Pacote RTP (SCHULZRINNE et al., 2003).

- (Ver.) *Version* - 2 bits: identifica a versão do RTP;

- (P) *Padding* - 1 bit: se estiver habilitado, indica que o pacote contém dados adicionais que não fazem parte da carga útil. O número de bytes que devem ser ignorados estão definidos no último byte do pacote. O *padding* pode ser utilizado em algoritmos de criptografia que necessitam de um bloco de bytes com um tamanho fixo;

- (X) *Extension* - 1 bit: se habilitado, o cabeçalho fixo é seguido de um cabeçalho de extensão que possibilita adicionar informações suplementares ao cabeçalho;

- (CC) CSRC *count* - 4 bits: informa o número de origens de contribuição estão presentes;

- (M) *Marker* - 1 bit: utilizado como marcador pela camada de aplicação. Possui a capacidade de determinar, por exemplo, o início de um quadro de vídeo, o qual é enviado em vários pacotes;

- (*Payload type*) *Payload Type* - 7 bits: informa o formato de codificação da carga útil, tais como MP3, codificação GSM ou áudio não-compactado. O Playload type pode ser modificado durante a transmissão, possibilitando que um fluxo possa ser transmitido em amostras com codificações diferentes. Entretanto, isso não deve ser utilizado para multiplexar diferentes formatos de mídia, como vídeo e áudio;

- (*Sequence Number*) *Sequence Number* - 16 bits: contador que identifica cada pacote RTP em um fluxo. Seu valor inicial pode, para evitar colisões, ser escolhido aleatoriamente e é incrementado a cada pacote enviado. Essa numeração permite à aplicação efetuar a ordenação dos pacotes recebidos e a detecção de perdas;

- (*Timestamp*) *Timestamp* - 32 bits: consiste em um campo com a função de marcar a hora em que cada pacote foi gerado, possibilitando à aplicação destinatária reproduzir o fluxo corretamente, ou seja, armazenar e executar cada fração de mídia no tempo correto. Para isso, não é considerado o valor absoluto do *Timestamp*, mas a diferença entre cada pacote e o pacote inicial, obtendo-se assim o intervalo em milissegundos para a reprodução da amostra;

- (SSRC) *Synchronization Source Identifier* - 32 bits: informa a qual fluxo o pacote pertence. Com isso, é possível multiplexar e desmultiplexar diversos fluxos de dados de tempo real sobre um único fluxo de pacotes UDP. O seu valor normalmente é definido de modo aleatório;

- (CSRC *List*) *Contributing Source Identifier* - 0 a 15 itens, 32 bits cada: identifica todas as fontes que contribuíram para gerar o *payload* contido no pacote. Exemplificando: em cenário de uma conferência de áudio, vários participantes podem enviar *streams* ao mesmo tempo, um *mixer* (misturador) sincronizará todos os pacotes recebidos combinando-os em um único pacote para enviá-lo aos destinatários. Nesse caso, o campo SSRC identificará o mixer, enquanto o CC (CSRC count) conterá o número de participantes que contribuíram para a geração daquele *stream* e a lista CSRC irá identificá-los (TANENBAUM, 2003).

2.2.2.1 RTCP - Real-time Transport Control Protocol

Com o intuito de realizar o monitoramento de qualidade e um controle mínimo de sessão, é utilizado, em conjunto com o RTP, o protocolo RTCP (SCHULZRINNE et al., 2003). Esse protocolo não transmite quaisquer dados de mídia, mas com a constante troca de pacotes de controle entre os usuários da sessão, realiza as seguintes funções (TANENBAUM, 2003):

- Fornecer um *feedback* sobre a qualidade de serviço da transmissão (retardo, flutuação, largura de banda, congestionamento e outros). Com essas informações é possível adaptar dinamicamente a codificação, aumentando ou diminuindo a taxa de transmissão conforme a qualidade da rede. Além disso, em um ambiente *multicast* elas podem ser críticas no diagnóstico de falhas na distribuição e como os pacotes RTCP vão para todos os usuários, é possível identificar se os problemas são locais ou globais. Uma terceira entidade que não faz parte da sessão, tal como um sistema que aplica QoS (*Quality of Services*) (HARDY, 2001), pode também receber esses relatórios (SCHULZRINNE et al., 2003).

- Manter um identificador em nível de transporte para uma fonte RTP. Esse identificador, chamado *Canonical Name* ou CNAME, permite aos receptores associarem e sincronizarem múltiplos fluxos de dados, tais como, vídeo e som de um mesmo emissor.

- Calcular e condicionar a taxa de envio dos pacotes RTCP, uma vez que, com o aumento de participantes, essa taxa pode elevar substancialmente (SCHULZRINNE et al., 2003).

- Transportar informações mínimas de controle de sessão (ex.: nome, email e número de telefone). Essa funcionalidade é útil em sessões onde existe pouco controle, em que os participantes entram e saem sem nenhum tipo de negociação.

Os relatórios sobre a qualidade da sessão são enviados nos pacotes *Sender Report* (SR) ou *Receiver Report* (RR). Os dois pacotes transportam estatísticas de recepção, porém, o pacote SR contém informações adicionais sobre transmissões, sendo essas utilizadas somente por participantes emissores ativos (SCHULZRINNE et al., 2003). Os principais campos que compõem os relatórios são:

- NTP *timestamp* - 64 bits: captura o horário local do sistema da máquina fonte no momento do envio da mensagem SR. Para isso, usa-se a formatação do NTP - *Network Time Protocol*;

- RTP *timestamp* - 32 bits: corresponde ao instante de tempo (normalmente múltiplo da unidade de amostragem) em que o último pacote RTP foi enviado por determinada fonte;

- *Sender's packet count* - 32 bits: contador cumulativo do total de pacotes de dados RTP transmitidos pelo emissor do início da transmissão até o momento da geração de um pacote SR;

- *Fraction lost* - 8 bits: essa fração é definida pelo número de pacotes perdidos dividido pelo número de pacotes esperado;

- *Cumulative number of packets lost* - 24 bits: número total de pacotes de dados RTP que foram perdidos do início da recepção (valor cumulativo). Esse número é definido com a subtração do número de pacotes esperado pelo número de pacotes atualmente recebidos. A quantidade de pacotes esperado é definido pelo campo *Extended Highest Sequence Number Receive* do pacote RR;

- *Interarrival Jitte*r - 32 bits: esta é uma estimativa da variância estatística do tempo entre chegadas dos pacotes de dados RTP, medido em unidade de amostragem e expresso sempre como um inteiro positivo (MARCONDES et al., 2003).

O protocolo RTCP ainda contém os pacotes (SCHULZRINNE et al., 2003):

- *Source Description* (SDES), para distribuição de controle de sessão e do CNAME;

- *Application* (APP), dependente da aplicação;

- *Bye* (BYE) que indica o fim da participação.

2.3 Qualidade de serviço em VoIP

A qualidade de serviço é fundamental para a operabilidade da arquitetura VoIP (HARDY, 2001). Nas próximas seções são apresentados os fatores relacionados diretamente com a qualidade das ligações.

2.3.1 Atraso

O tempo total da propagação da voz, desde a sua captura até a reprodução no usuário final, não pode ultrapassar determinados limites para não torna-se perceptível aos participantes, o que dificulta a interatividade em uma conversação (INTERNATIONAL TELECOMMUNICATION UNION, 2003b).

O atraso completo, também conhecido como tempo de fala-e-escuta, é constituído por várias partes (Figura 2.7).

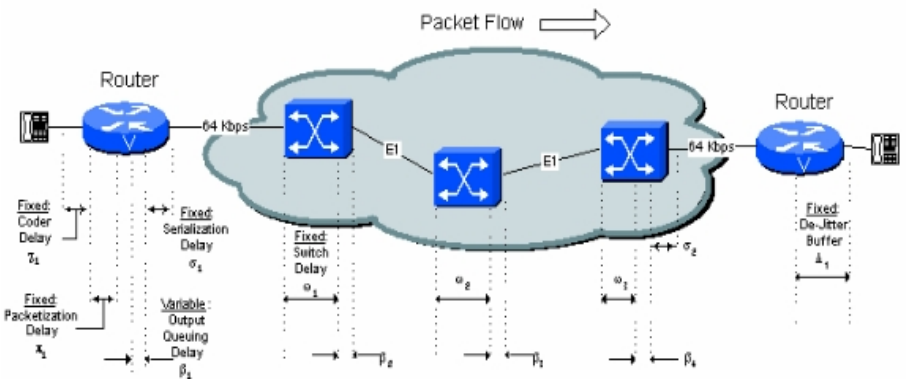

Figura 2.7: Atrasos total de fala-e-escuta (CISCO, 2007a).

Inicialmente, a voz é amostrada e processada pelo *codec*. Em seguida, existe o tempo de empacotamento e serialização do pacote antes dele ser encaminha para a rede. Uma vez na rede, existe o tempo de propagação até o destino final além de atrasos variados nos *buffers* dos roteadores. Ao chegar no *host* final, o pacote ainda pode permanecer retido em

buffers de amortização de *jitter*. Até que finalmente o pacote é processado e reproduzido (CISCO, 2007a).

A seguir serão apresentados alguns detalhes de cada um desses componentes.

2.3.1.1 Atraso no Codec

Os *codecs* trabalham com amostragem de voz, chamados de *frames*. O tempo de amostragem pode variar dependendo do *codec* utilizado. Além do tempo de coleta da voz, alguns *codecs* realizam a análise (*look ahead*) do início da próxima amostragem para otimização da codificação dos *frames*. Um pacote pode ser formado por vários *frames*. Assim, o tempo de codificação da voz pode ser definida com a seguinte equação (INTERNATIONAL TELECOMMUNICATION UNION, 2003b):

$$(N * Tamanho\ \textbf{Frame}) + Look\ ahead \tag{2.1}$$

sendo N o número de *frames* agrupados.

Na Figura 2.8 é mostrada um exemplo de empacotamento de três *frames* com o tamanho de 10 ms.

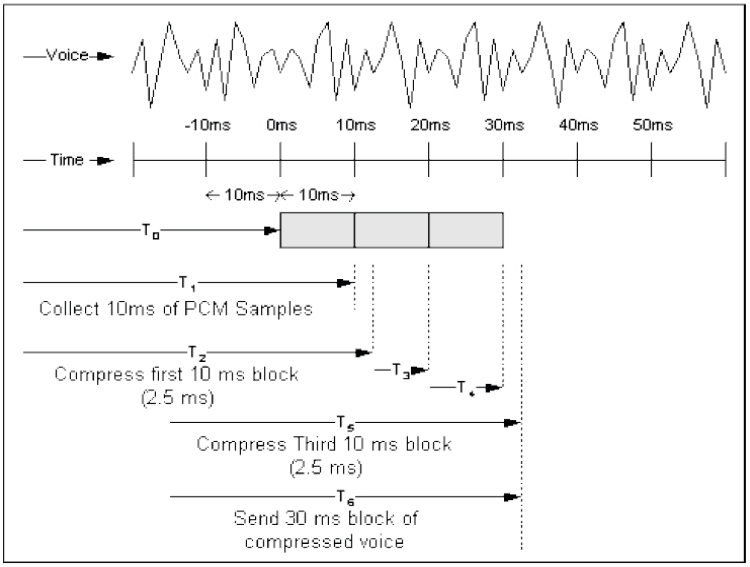

Figura 2.8: Ilustração de empacotamento de *frames* VoIP (CISCO, 2007a).

Nota-se que a amostragem de um novo *frame* é realizada concomitante ao processamento do *frame* anterior (CISCO, 2007a).

Na Tabela 2.1 são apresentados alguns exemplos de atrasos inseridos pela codificação de um *frame* de voz para diferentes *codecs*.

Tabela 2.1: Atraso com a codificação dos *payloads* de voz.

Codec	*Frame (ms)*	*Look-ahead*	*Atraso (ms)*
G.711 (64 Kbit/s)	0.125	0	0.25
G.729 (8 Kbit/s)	10	5	25
G.723.1 (5.3 Kbit/s)	30	7.5	67.5

Fonte (INTERNATIONAL TELECOMMUNICATION UNION, 2003b).

O tempo de descompressão é, aproximadamente, dez por cento do tempo de compressão de cada bloco (CISCO, 2007a).

2.3.1.2 Serialização

Uma vez formado o pacote, ele é serializado para o transporte na rede. Dependendo da velocidade do enlace e do tamanho do pacote o tempo de serialização pode ser significativo (CISCO, 2007a).

2.3.1.3 Propagação na Rede

A transmissão do pacote na rede é constituída por dois componentes: um tempo fixo correspondente à propagação nos enlaces e um tempo variável em que os pacotes ficam aguardando nos *buffers* dos roteadores.

O atraso de propagação depende do enlace atravessado. Na Tabela 2.2 são apresentados alguns exemplos de tempo de propagação em diferentes tipos de enlaces.

Tabela 2.2: Tempo de propagação em diferentes enlaces.

Tipo de enlace	*Tempo de propagação*
Enlace com cabo coaxial	0.006 ms/km
Enlace com fibra óptica	0.005 ms/km
Satélite (14 000 km altitude)	110 ms

Fonte: (INTERNATIONAL TELECOMMUNICATION UNION, 2003b).

A baixa capacidade de alguns enlaces somado ao congestionamento ocasionam a aglomeração dos pacotes na saída dos roteadores. Esse atraso varia conforme o cenário encontrado.

Devida a variação no atraso sofrida pelos pacotes (Seção 2.3.2), normalmente, são aplicados *buffers* nos *hosts* finais para amortização dessa variação.

Ao considerar que a variação do atraso pode fazer com que um pacote chegue com atraso maior ou menor que o antecessor, é comum considerar o tempo em que o pacote permanecerá no *buffer* de *de-jitter* um tempo equivalente à metade do tamanho do *buffer* (INTERNATIONAL TELECOMMUNICATION UNION, 2006).

2.3.2 *Jitter*

A disputa por determinados recursos, como um enlace de baixa velocidade, faz com que os pacotes de um mesmo fluxo experimente tempos diferentes nos *buffers* de saída dos roteadores na rede, o que provoca uma variação no tempo de chegada (*jitter*) de cada pacote (TANENBAUM, 2003). Essa variação provoca uma sensação de descontinuidade no áudio (INTERNATIONAL TELECOMMUNICATION UNION, 2003b).

Buffer de amortização de *jitter* transformam os atrasos variáveis em atrasos constantes. Para isso, os pacotes são retidos e repassados à aplicação em uma taxa constante. Assim, caso um pacote chegue com alguma variação no atraso, ele é realocado em seu tempo correto de reprodução (CISCO, 2007a).

A definição do tamanho do *buffer* é uma tarefa altamente relevante. Isso porque um *buffer* muito grande faz com que aumente o atraso final de fala-e-escuta, degradando as conversações. Já um *buffer* pequeno pode fazer com que os pacotes com uma variação alta no atraso sejam descartados por chegarem tarde de mais (RAMJEE et al., 1994).

Desse modo, existem diversas abordagens para definição ideal do tamanho dos *buffers* de amortização do *jitter*. Basicamente, pode-se definir um buffer fixo, ou um com tamanho variável conforme algum critério de adaptabilidade. A Figura 2.9 ilustra um exemplo de um *buffers* dinâmico.

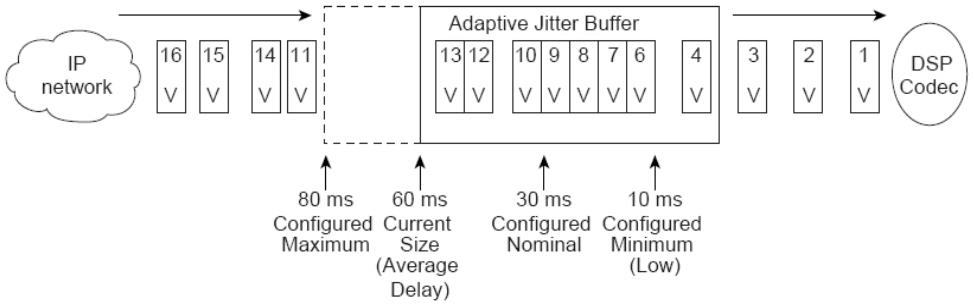

Figura 2.9: Ilustração de um *buffer* dinâmico de amortização de *jitter* (CISCO, 2007b).

Nesse exemplo são determinados três parâmetros: os tamanhos mínimo (*Configured Minimum*) e máximo (*Configured Maximum*) que o *buffer* pode assumir, além do tempo nominal (*Configured Nominal*), ou seja, o atraso de retenção aplicado no início da ligação. O tamanho do *buffer* é adaptado conforme a variação encontrada na rede, onde se expande rapidamente quando identificado uma variação muito longa, e é reduzida mais lentamente para diminuir o tempo de *playout* [1], mas ao mesmo tempo tentar evitar o descarte dos pacotes que chegarem muito tarde. Observa-se que algumas lacunas são deixadas no lugar de pacotes que foram perdidos ou ainda não chegaram. Desse modo, o *buffer* também ordena os pacotes e sempre os repassam ao DSP (*Digital Signal Processor*) na taxa esperada (CISCO, 2007b).

2.3.3 Perda de pacotes

Em VoIP, normalmente, não é recomendada a retransmissão dos pacotes perdidos, porque devido às características de tempo real um pacote retransmitido provavelmente não chegaria a tempo de ser aproveitado. Pelo mesmo motivo, pacotes que não foram descartados na rede, mas chegaram tarde demais também são desconsiderados pelas aplicações (TANENBAUM, 2003).

Assim, além dos problemas tradicionais de descarte de pacotes como congestionamento na rede e enlaces com erros, o atraso e o *jitter* em demasiado também podem fazer com que os pacotes VoIP sejam descartados (RAMJEE et al., 1994).

O tamanho reduzido dos pacotes VoIP, com amostras de 12.5 ms a 62.5 ms, minimiza a perda de um pacote. Contudo, um descarte normalmente está relacionado com algum problema na rede, assim as perdas não ocorrem isoladamente (KUHN; WALSH; FRIES, 2005).

A ausência de trechos da voz, ocasionada pelo descarte dos pacotes, provoca um desconforto alto aos participantes. Em (TELECOMMUNICATIONS INDUSTRY ASSOCIATION, 2006) foi demonstrado que perdas acima de 5% em tráfegos codificados pelo *codec* G.711 degrada a qualidade para patamares abaixo dos encontrados na telefonia PSTN. Esse quadro pode ser encontrado com perdas menores, entre 1% a 2%, quando aplicado *codecs* de alta compressão como G.723.1 e G.729 (TELECOMMUNICATIONS INDUSTRY ASSOCIATION, 2006).

[1] Tempo em que um pacote permanece retido no *buffer* de amortização de *jitter*

2.3.4 Banda

Aplicações de VoIP apresentam uma baixa performance na ocupação efetiva da banda. Isso ocorre devido às características de tempo real dessa tecnologia, em que o tempo de amostragem da voz não pode ser demorado para evitar o aumento do atraso fim-a-fim. Além disso, normalmente, também é realizada a compactação dos dados, resultando em um número reduzido de bytes em um *frame* (CISCO, 2007c).

Desse modo, o *overhead* com os cabeçalhos necessários para transmissão dos pacotes VoIP é altamente significativo (BARBIERI; BRUSCHI; ROSTI, 2002). Na Figura 2.10 é apresentado um gráfico com a percentagem da relação entre o tamanho do *payload* e dos cabeçalhos VoIP.

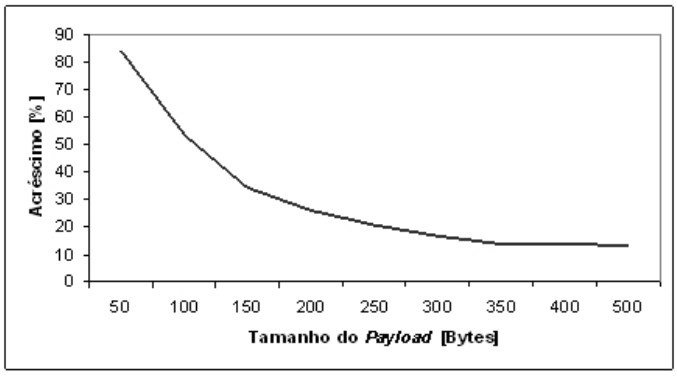

Figura 2.10: Expansão no tamanho do pacote em relação ao *payload* VoIP.

Como pode-se observar os pacotes com *payloads*, comumente, utilizados em VoIP (abaixo de 50 bytes) apresentam um acréscimos de mais de 80% com a inserção dos cabeçalhos.

Na Tabela 2.3 são apresentados a ocupação da banda com diferentes *codecs*.

Tabela 2.3: Ocupação da banda para diferentes *codecs*.

Codec	*Tamanho do payload (ms)*	*Banda Ethernet (Kbit/s)*
G.711 (64 Kbit/s)	160	87.2
G.729 (8 Kbit/s)	20	31.2
G.723.1 (6.3 Kbit/s)	24	21.9
G.726 (24 Kbit/s)	60	47.2

Fonte: (CISCO, 2007c)

O cálculo do consumo da banda é realizado da seguinte maneira (CISCO, 2007c):

- Tamanho total do pacote = (Cabeçalho da Camada de Enlace) + (Cabeçalhos IP/UDP/RTP) + (Tamanho do *payload* de voz);
- Pacotes por Segundo = (Taxa de transmissão de bit do *codec*) / (Tamanho do *payload* de voz);
- *Banda* = Tamanho total do pacote * Pacotes por Segundo.

2.4 E-model

Como visto nas Seção 2.3, diversos fatores influenciam na qualidade das ligações VoIP. Isso demonstra a importância da possibilidade de se mensurar o impacto de todos esses fatores na percepção final dos usuários nas conversações.

Desse modo, em (INTERNATIONAL TELECOMMUNICATION UNION, 1996) foi determinado o método para Pontuação de Opinião Média, ou MOS (*Mean Opinion Score*), o qual apresenta um mecanismo para avaliação subjetiva do efeito dos sistemas e componentes nas transmissões. Para isso, os usuários estipulam uma nota em uma escala de 1 (pobre) a 5 (excelente) conforme a sua percepção de qualidade dos sistemas avaliados.

Entretanto, um método subjetivo é de difícil reprodução e de complexa aplicabilidade em larga escala (SUN; IFEACHOR, 2006).

Um método alternativo consiste no Modelo-E (INTERNATIONAL TELECOMMUNICATION UNION, 2005a), o qual por sua vez apresenta um modelo computacional para determinação da qualidade das ligações de modo objetivo. Nele são considerados diversos fatores, tais como distorções com os equipamentos utilizados, degradações com atrasos na rede, eco nas conversações, entre outros.

Nas próximas seções será apresentada a definição desse modelo.

2.4.1 Visão geral sobre E-model

No Modelo-E as deteriorações são consideradas como fatores aditivos, podendo ser calculadas separadamente e combinadas com a seguinte formulação (INTERNATIONAL TELECOMMUNICATION UNION, 2005a):

$$R = Ro - Is - Id - Ie + A \tag{2.2}$$

O fator Ro representa os efeitos da relação sinal-ruído; Is as perdas simultâneas ao sinal de voz; Id as perdas associadas ao atraso fim-a-fim; Ie mapeia as perdas associadas ao equipamento utilizado; e A corresponde ao fator de vantagem, ou fator de expectativa. Como resultado dessa equação tem-se um valor escalar R, o qual varia na escala de 0 até 100 (INTERNATIONAL TELECOMMUNICATION UNION, 2005a).

O resultado dessa equação pode ser convertido à escala MOS (*Mean Opinion Score*) (INTERNATIONAL TELECOMMUNICATION UNION, 1996) representando o nível de satisfação dos participantes nas ligações (Tabela 2.4).

Tabela 2.4: Relação entre o valor de R e a escala MOS.

R	**MOS**	**Satisfação dos Usuários**
90	4,34	Muito satisfatório
80	4,03	Satisfatório
70	3,60	Alguns usuários insatisfeitos
60	3,10	Muitos usuários insatisfeitos
50	2,58	Quase todos usuários insatisfeitos

Fonte: (INTERNATIONAL TELECOMMUNICATION UNION, 2005a).

Nas próximas seções são apresentados os modos de obtenção de cada fator.

2.4.2 Relação sinal-ruído básica, Ro

Ro representa a relação sina-ruído básica, incluindo ruídos na fonte e no lado do emissor, no circuito de transmissão, no ambiente e o teto de ruído correspondente à sensibilidade do sistema auditivo humano. Em (INTERNATIONAL TELECOMMUNICATION UNION, 2005a) são apresentadas formulações para estipular esse fator, sendo que utilizando os valores padrão tem-se Ro equivalente a 94,77.

2.4.3 Perdas simultâneas, Is

O fator Is é uma combinação de todas as degradações que ocorrem mais ou menos simultaneamente com o sinal de voz. Entre elas estão a degradação na qualidade causada pelo volume de áudio muito alto, pela interferência da voz do locutor em seu próprio fone de ouvido (*sidetone*) e pela quantização decorrente da digitalização do sinal de voz (INTERNATIONAL TELECOMMUNICATION UNION, 2005a).

Utilizando os valores padrão definidos em (INTERNATIONAL TELECOMMUNICATION UNION, 2005a) estipula-se o fator Is em 1,41.

2.4.4 Fator de Vantagem, A

O fator de vantagem A permite compensar os fatores degradantes quando existem outras vantagens no uso (INTERNATIONAL TELECOMMUNICATION UNION, 2005a). Isso permite que a tolerância dos usuários em relação à tecnologia ou ao ambiente de uso podem ser consideradas para a determinação final do cálculo de qualidade.

Um exemplo consiste na diferença da expectativa dos usuários em conversações na telefonia fixa e na telefonia móvel, onde nessa última os usuários aceitam uma deterioração maior que na primeira.

Valores provisórios do fator A são especificado em (INTERNATIONAL TELECOMMUNICATION UNION, 2005a) para alguns tipos de sistemas de comunicação (Tabela 2.5).

Tabela 2.5: Exemplos de valores provisórios do fator de vantagem A.

Exemplo de sistema de comunicação	**Máximo valor para A**
Telefonia fixa	0
Telefonia móvel para redes celular *in-door*	5
Telefonia móvel em redes geográficas	10
Locais de difícil acesso (ex.: redes com enlaces de satélites).	20

Fonte: (INTERNATIONAL TELECOMMUNICATION UNION, 2005a).

2.4.5 Perdas com atraso, Id

As perdas causadas pelo atraso são computadas pelo fator Id, o qual também é subdividido em três componentes:

$$Id = Idte + Idle + Idd \tag{2.3}$$

Os fatores $Idte$ e $Idle$ obtêm uma estima do impacto com o echo no lado do emissor e do receptor, respectivamente. A deterioração com o atraso total é representada pelo fator Idd (INTERNATIONAL TELECOMMUNICATION UNION, 2005a).

A seguir são apresentadas as equações para cálculo dos componentes do Id.

2.4.5.1 *Eco no transmissor* $Idte$

$$Idte = \left[\frac{Roe - Re}{2} + \sqrt{\frac{(Roe - Re)^2 +}{4} + 100} - 1\right]\left(1 - e^{-T}\right) \tag{2.4}$$

onde:

$$Roe = -1,5(No - RLR) \tag{2.5}$$

$$Re = 80 + 2,5(TERV - 14) \tag{2.6}$$

$$TERV = TELR - 40log\frac{1 + \frac{T}{10}}{1 + \frac{T}{150}} + 6e^{-0,3T^2} \tag{2.7}$$

2.4.5.2 *Eco no receptor* $Idle$

$$Idle = \frac{Ro - Rle}{2} + \sqrt{\frac{(Ro - Rle)^2}{4} + 169} \tag{2.8}$$

em que:

$$Rle = 10,5(WEPL + 7)(Tr + 1)^{-0,25} \tag{2.9}$$

2.4.5.3 *Perdas com atrasos longos* Idd

$$Idd = 25\left\{\left(1 + X^6\right)^{\frac{1}{6}} - 3\left(1 + \left[\frac{X}{3}^6\right]\right)^{\frac{1}{6}} + 2\right\} \tag{2.10}$$

onde:

$$X = \frac{log\left(\frac{Ta}{100}\right)}{log2} \tag{2.11}$$

Na Tabela 2.6 são apresentados os valores padrão recomendados em (INTERNATIONAL TELECOMMUNICATION UNION, 2005a) dos parâmetros utilizados nas Equações 2.4 - 2.11.

Tabela 2.6: Valores padrão para cálculo do fator Id.

Abreviatura	**Parâmetro**	**Valor padrão**
No	Ruído total	-61,18 dBm0p
Ro	Relação sinal-ruído básica	94,77
RLR	Nível de intensidade no receptor	+2 dB
$TELR$	Nível de intensidade do eco no emissor	65
$WELP$	Caminho do eco ponderado	110 dB

Fonte: (INTERNATIONAL TELECOMMUNICATION UNION, 2005a).

Nota-se ainda que no cálculo do fator Idd são considerados três parâmetros diferentes associados com o tempo na transmissão (Figura 2.11).

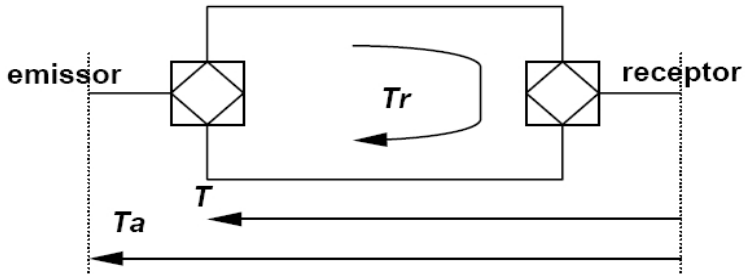

Figura 2.11: Ilustração de diferentes tipos de atrasos num sistema de comunicação de voz (LUSTOSA et al., 2004).

Ta representa o atraso total entre o emissor e o receptor e é usado para estipular a degradação com atrasos muito longos. T representa o atraso médio percorrido pelo eco. Tr consiste no atraso de ida e volta (*round-trip delay*) no circuito a quatro fios [2] (INTERNATIONAL TELECOMMUNICATION UNION, 2005a).

2.4.6 Deterioração com equipamentos, Ie

Os valores para o fator Ie são determinados de duas maneiras, uma para cenários sem perdas e outra quando existem perdas de pacotes.

Na primeira definição os valores de Ie não estão relacionados com nenhum outro parâmetro. Define-se Ie em redes sem perdas baseado na opinião subjetiva baseada na experiência da rede. Os valores são constantemente atualizados em (INTERNATIONAL TELECOMMUNICATION UNION, 2001). Na Tabela 2.7 são apresentados exemplos do fator Ie para algumas famílias de *codecs*.

Tabela 2.7: Exemplos de valores provisórios do fator Ie.

Tipo de *codec*	**Referência**	**Taxa de operação (kbits/s)**	**Ie**
PCM	G.711	64	0
ADPCM	G.726, G.727	40	2
	G.721 (1988), G.726, G.727	32	7
	G.726, G.727	24	25
	G.726, G.727	16	50
CS-ACELP	G.729	8	10
	G.729-A + VAD	8	11
RPE-LTP	GSM 06.10, Full-rate	13	20
VSELP	GSM 06.20, Enhanced Full Rate	12.2	5
ACELP	G.723.1	5.3	19

Fonte: (INTERNATIONAL TELECOMMUNICATION UNION, 2001).

Já em cenários onde os *codecs* operam sobre perda aleatória de pacotes define-se os valores de Ie baseado em uma formulação que relaciona a probabilidade de perda de pacotes do canal (Ppl) com a robusteza de cada *codec* (Bpl), a qual também é definida em (INTERNATIONAL TELECOMMUNICATION UNION, 2001) para alguns *codecs*. O resultado dessa equação é chamado de Fator Efetivo de Perdas do Equipamento (*Effective Equipment Impairment Factor*, Ie_eff), sendo derivado do valor de Ie do determinado

[2] circuito com separação física entre os dois sentidos de comunicação.

codec quando não existem perdas (Equação 2.12) (INTERNATIONAL TELECOMMUNICATION UNION, 2005a).

$$Ie\_eff = Ie + (95 - Ie).\frac{Ppl}{\frac{Ppl}{BurstR}.Bpl} \tag{2.12}$$

O parâmetro $BurstR$ (***Burst Ratio***) - Taxa de Rajadas, representa à razão entre o tamanho médio observado das rajadas em um seqüência entregue e o tamanho médio esperado de rajadas para a rede com perdas aleatórias. Assim, quando as perdas apresentam características aleatórias, independentes, assume-se $BurstR$ igual à 1, e o inverso, ou seja, quando as perdas são dependentes uma das outras (em rajadas), define-se $BusrtR$ com um valor maior que 1 ($BurstR$ > 1) (INTERNATIONAL TELECOMMUNICATION UNION, 2005a).

2.5 Vulnerabilidades e Riscos de Segurança em VoIP

A arquitetura VoIP inclui uma variedade de componentes, entre eles, controladores de chamadas, *gateways*, roteadores, *proxys*, protocolos entre outros. Essa infra-estrutura não apresenta uma mesma proteção como encontradas em redes de telefones tradicionais (KUHN; WALSH; FRIES, 2005). Pacotes trafegam na Internet por um número maior de pontos, os quais não estão sobre controle total dos *hosts* finais. Assim os atacantes podem aproveitar possíveis vulnerabilidades em algum desses componentes para realizar ataques que comprometam as premissas de segurança nas conversações (KUHN; WALSH; FRIES, 2005).

Nas próximas seções serão apresentadas alguns riscos e vulnerabilidades nas aplicações VoIP.

2.5.1 Confidencialidade

A confidencialidade refere-se ao requisito de manter as informações privadas inacessíveis à terceiros não autorizados. Um exemplo de ataques contra a privacidade das conferências VoIP consiste na escuta indevida das conversações. Nesse tipo de risco um atacante é capaz de monitorar as sinalizações e/ou o *stream* entre dois ou mais componentes VoIP, contudo os dados não são alterados. (VOIPSA, 2005).

Sniffer são exemplos de mecanismo que podem ser utilizados pelos atacantes para ter acesso aos dados das conversações. Os atacantes podem agir como um intermediário (*man-in-the-middle*), capturando as informações sem serem notados pelos participantes, ou também pode forjar uma identidade e com isso obter acesso às conversações de maneira indevida (KUHN; WALSH; FRIES, 2005).

O domínio em que uma requisição é destinada, geralmente, é especificado em uma *Request*-URI. UAs normalmente contatam um servidor deste domínio para a entrega de uma requisição. Entretanto, isso abre a possibilidade de um atacante imitar o papel de um servidor remoto, e com isso interceptar as requisições enviadas a aquele domínio (INTERNET2 VIDMID VIDEO MIDDLEWARE GROUP, 2005).

Uma vez obtido acesso às conversações o atacante pode, sem autorização, capturar, analisar, gravar e reconstruir os *streams* de áudio e/ou os demais dados privados de uma comunicação, como a identificação dos usuários, senhas, números e endereços de contatos, entre outros (KUHN; WALSH; FRIES, 2005).

2.5.2 Integridade

A integridade refere-se à necessidade das informações não sofrerem alterações sem autorização (KUHN; WALSH; FRIES, 2005).

Atacantes podem interceptar e modificar o tráfego como um intermediário na conversação. Também existe o risco do descarte, absorção ou redirecionamento dos pacotes. Esse último pode incluir em uma conversação um nó não autorizado ou também pode excluir nós autorizados (VOIPSA, 2005).

Um exemplo de vulnerabilidade se enquadra no fato em que um SIP UAs pode utilizar um *proxy* seguro (utilizando autenticação) para garantir a entrega correta das mensagens, mas não pode impedir a modificação maliciosa do conteúdo das mensagens (*bodies*). Um *proxy* malicioso pode : modificar a chave da sessão, atuar como um *man-in-the-middle*, ou modificar as características de seguranças definidas pela origem. Esse risco ocorre em todas as informações fim-a-fim do SIP: MIME, SDP, entre outros (INTERNET2 VIDMID VIDEO MIDDLEWARE GROUP, 2005).

2.5.3 Deterioração e Negação de Serviço

Devido a alta sensibilidade em relação à performance em aplicações VoIP, o risco tradicional na Internet de negação de serviço, ou DoS (*Denied of Service*), aumenta para esse tipo de aplicação (KUHN; WALSH; FRIES, 2005).

Existem diversos modos de degradação das conversações. Entre eles pode-se atingir os requisitos de QoS, por exemplo aumentando a latência ou o número de pacotes perdidos, pode-se também sobrecarregar os componentes vitais ou ainda utilizar as vulnerabilidades dos protocolos de sinalização para tornar algum *endpoint* ou servidor indisponível (VOIPSA, 2005).

A seguir são citados alguns exemplos desse tipo de risco.

2.5.3.1 Inundação de mensagens

Atacantes podem inundar os sistemas VoIP com mensagens válidas ou inválidas. Isso pode sobrecarregar os sistemas, tornando inoperante o serviço. Esse ataque pode ser direcionado a um UAC, encaminhando diversas solicitações de início ou restabelecimento de chamadas (SIP INVITE e re-INVITE), ou aos controladores das ligações, atingindo um número grande de *endpoints*. Os ataques ainda podem visar os componentes vitais para o funcionamento do serviço, como servidores de DHCP, DNS, etc (VOIPSA, 2005).

Em muitas arquiteturas, servidores SIP (*proxyes*) apresentam um interface pública com a Internet para aceitar requisições, criando com isso potenciais oportunidades de negação de serviço. Por exemplo, atacantes podem falsificar a origem de uma requisição e envia-la para diversos elementos SIP, fazendo com que esses componentes causem uma sobrecarga em algum alvo da rede SIP. Atacantes também podem registrar diversos contatos falsos degradando a memória dos servidores de registro. A utilização de multidifusão para transmitir requisições SIP podem aumentar o potencial de ataques de DoS (INTERNET2 VIDMID VIDEO MIDDLEWARE GROUP, 2005).

2.5.3.2 Mensagens mal formadas

Outro tipo de ataque DoS consiste no envio de mensagens mal formadas com o objetivo de impedir a operabilidade do serviço. Nesse tipo de ataque usuários com acesso permitido aos servidores podem enviar mensagens com algum erro de formação e, com isso, provocar o erro ou o mal funcionamento dos dispositivos, degradando ou impedindo

o processamento das mensagens corretas (VOIPSA, 2005).

Esse risco está ligado ao fato dessa arquitetura normalmente estar em evolução. Assim os sistemas não apresentam uma robusteza satisfatória, o que possibilita a falha dos sistemas com mensagens inesperadas (VOIPSA, 2005). Algumas mensagens inválidas podem consumir um consumo considerado da capacidade de processamento, e pode corromper a máquina de processamento do protocolo com um *overflow* do *buffer* de mensagens.

2.5.3.3 Falsificação de mensagens

Os atacantes podem também usar mensagens válidas do protocolo com um modo mal intencionada (VOIPSA, 2005).

Pode-se, por exemplo, terminar indevidamente uma sessão acrescentando um BYE em uma mensagem legitima. É possível também forjar um re-INVITE, modificando parâmetros de segurança ou re-encaminhando sessões RTP (VOIPSA, 2005).

Um criminoso pode enviar uma mensagem "*Busy Here*"quando responde uma chamada, ou alterar permanentemente ("*Moved Permanently*") o domínio para um local indevido, fazendo com que a vítima não receba mais nenhuma chamada (INTERNET2 VIDMID VIDEO MIDDLEWARE GROUP, 2005).

2.5.3.4 Abuso de QoS

Atacantes podem enfocar diretamente nos parâmetros de qualidade. Por exemplo, pode-se usar um *codec* diferente do declarado na negociação da chamada. Também é possível o abuso ou uso impróprio do QoS definido para o fluxo de voz, afetando as premissas de qualidade (VOIPSA, 2005).

2.5.3.5 Ataque aos serviços da rede

Existe também a oportunidade de ataques aos componentes e recursos da rede essenciais para o funcionamento do serviço de VoIP. Ex.: *buffer overflow* dos componentes da rede (roteador, *switch*, *proxy*, etc) derrubando, reiniciando, ou ocupando toda a rede disponível (SYN *attack*, *Smurf Attack*, etc.) ou uma re-configuração não autorizada dos comportamento dos dispositivos (DHCP, AAA, TFTP, etc) (VOIPSA, 2005).

2.5.4 Abuso do serviço

Uma outra categoria de riscos em aplicações VoIP se enquadra no uso indevido do serviço, onde usuários maliciosos utilizam-se de recursos dos sistemas VoIP para obter alguma vantagem ou prejudicar alguma vítima (VOIPSA, 2005).

Entre os exemplos desse tipo de riscos está na possibilidade de algum usuário malicioso aproveitar a flexibilidade dos protocolos para esconder ou alterar a sua identificação com a meta de não ser identificado em algum ato irregular. Pode-se ainda usar mecanismos para impedir a bilhetagem das sessões, ou ainda, aumentar o seu tráfego sem autorização (VOIPSA, 2005).

2.6 Mecanismos de Segurança em VoIP

A partir das ameaças de segurança em VoIP pode-se citar como serviços de proteção necessários mecanismos que possibilitem (KUHN; WALSH; FRIES, 2005):

- Preservar a confidencialidade e integridade das mensagens e dos fluxos de áudio;

- Prover autenticação dos participantes em uma sessão;
- Prevenir ataques de DoS.

Desse modo, diversas soluções vêm sendo propostas para alcançar essas premissas (KUHN; WALSH; FRIES, 2005). A seguir serão descritos alguns mecanismos para assegurar segurança em VoIP no âmbito da arquitetura SIP. Apresenta-se métodos de segurança para proteção da sinalização e da mídia das sessões.

2.6.1 Proteção das sinalizações SIP

Ao invés de definir novos mecanismos de segurança específicos, o SIP reutiliza modelos já existentes derivados dos protocolos HTTP e SMTP. Dentre eles estão o uso do método de autenticação *Digest* do HTTP, a proteção dos corpos das mensagens com S/MIME, entre outros. Devido à dificuldade da aplicação de mecanismos, como criptografia, de modo fim-a-fim, para esse propósito são recomendados serviços de segurança nas camadas de mais baixo nível (Seção 2.6.3 (KUHN; WALSH; FRIES, 2005).

Nas próximas seções são apresentados os principais mecanismos de segurança aplicáveis ao SIP.

2.6.1.1 Autenticação HTTP

A autenticação é a técnica utilizada por um UA assegurar que um servidor ou um agente de usuário é realmente quem deve ser e não um impostor. Com a identificação certificada o receptor da requisição pode conceder ou não a autorização para a realização do pedido (ROSENBERG et al., 2002)).

O SIP utiliza o método de autenticação *Digest* que se baseia no esquema de identificação do protocolo HTTP. Nesse método aplica-se o mecanismo de *challenge/reponse* (desafio/reposta), utilizando uma chave compartilhada. A chave normalmente contém o nome do usuário e uma senha encriptados (ROSENBERG et al., 2002).

Nesse mecanismo, quando um usuário tenta iniciar uma sessão, o *proxy* escolhe um desafio, um número aleatório muito extenso (*nonce*), e o envia ao usuário com a mensagem *Proxy Authorization Required*. Em seguida o usuário que recebeu o desafio, utiliza a chave compartilhada para encriptar o número recebido e o envia juntamente com um novo INVITE (TANENBAUM, 2003). Assim, como apenas aquele usuário poderia realizar a criptografia corretamente, a sua identidade é autenticada e o estabelecimento da sessão é autorizado. O *Digest* não garante a confidencialidade e integridade das mensagens (FRANKS et al., 1999).

Autenticação Usuário-para-Usuário

Quando um UAS recebe uma requisição de um UAC, o UAS autentica o originador antes de processar a requisição. Se uma credencial não estiver presente na requisição, o UAS pode enviar um desafio, com o uso da mensagem 401 (*Unauthorized*), para o originador prover a credencial (ROSENBERG et al., 2002).

O Campo de resposta WWW-*Authenticate* pode ser incluído na mensagem 401, o qual contém, ao menos, um desafio que indica uma autenticação e parâmetros aplicados ao *realm*.

Ao receber a resposta 401 (*Unauthorized*) o UAC pode, se estiver apto, reenviar a requisição com a credencial apropriada. Se não for obtida uma credencial o UAC pode tentar refazer a requisição com o nome do usuário como "anônimo"sem senha (INTERNET2 VIDMID VIDEO MIDDLEWARE GROUP, 2005).

Uma vez que uma credencial seja obtida, qualquer UA que deseja se autenticar com um UAS ou um servidor de registro, usualmente, mas não necessariamente, após receber uma resposta 401 (*Unauthorized*), pode realizar isso incluindo o campo de Autorização (*Authorization*) no cabeçalho da requisição. O campo *Authorization* consiste na credencial contendo informações de autenticação do UA (INTERNET2 VIDMID VIDEO MIDDLEWARE GROUP, 2005).

Autenticação Proxy-para-Usuário

Igualmente, quando um UAC envia uma requisição para um servidor de *Proxy*, o servidor autenticará o originador antes de processar a requisição. Se não existir uma credencial, no campo *Proxy-Authorization* do cabeçalho da requisição, o *proxy* pode desafiar o originador rejeitando a requisição com o código de resposta 407 (*Proxy Authentication Required*) (ROSENBERG et al., 2002).

Quando a origem UAC recebe a resposta 407 ela pode, se apta, reenviar a requisição adicionando a credencial. Isso é realizado incluindo o campo *Proxy-Authorization* no cabeçalho da requisição. Esse campo permite o cliente se identificar, contendo informações de autenticação do UA para o *proxy*.

Se uma credencial para o *realm* não pode ser alocada, o UAC pode tentar se autenticar com o nome de usuário como anônimo ("*anonymous*") e sem senha () (INTERNET2 VIDMID VIDEO MIDDLEWARE GROUP, 2005).

No cenário em que uma requisição bifurca-se (*forking*) em vários servidores de *proxys* e/ou Uas querem realizar a autenticação do UAC, o servidor de *proxy*, que realiza o *fork*, é responsável por reunir essas autenticações em uma única resposta. Se o UAC não prover uma credencial para cada desafio, o servidor de *proxy* que enviou os desafios não realizara a entrega das requisições.

Ao reenviar as requisições para respostas de 401 (*Unauthorized*) ou 407 (*Proxy Authentication Required*) que contenha múltiplos desafios de autenticação, um UAC deve incluir um valor de *Authorization* e de *Proxy-Authorization* para cada valor de WWW-*Authenticate* e *Proxy-Authenticate*, respectivamente (ROSENBERG et al., 2002).

2.6.1.2 S/MIME

As mensagens SIP são transmitidas em corpos de MIME - *Multipurpose Internet Mail Extensions*. Esse protocolo apresenta mecanismos de segurança para proteção da integridade e criptografia dos conteúdos, conhecido como S/MIME - *Secure* / MIME (RAMSDELL, 2004). Desse modo, o SIP pode utilizar o S/MIME para habilitar mecanismos como distribuição de chave pública, autenticação e integridade, ou confidencialidade nos dados de sinalização (KUHN; WALSH; FRIES, 2005).

O S/MIME pode ser considerado como uma substituição ao PGP - *Pretty Good Privacy*, provendo meios para a proteção de integridade e criptografia nas mensagens do SIP (RAMSDELL, 2004).

Para poder também proteger os campos do cabeçalho do SIP um tunelamento das mensagens SIP no corpo MIME é especificada. Nota-se que esse tunelamento, geralmente, acarreta um adicional aumento de *overhead* (KUHN; WALSH; FRIES, 2005).

O S/MIME requer certificados e chaves privadas para ser utilizado, apesar de que o certificados podem ser distribuído por um terceiro (*third-party*) confiável ou ser autogerada. Esse cenário pode não prover uma autenticação real de usuário, mas pode ser usada para prover uma maneira limitada de proteção da integridade das mensagens (KUHN; WALSH; FRIES, 2005).

2.6.2 RTP Seguro (SRTP)

Com o objetivo de prover requisitos de segurança na transmissão das mídias foram definidos os protocolos *Secure Real-time Transport Protocol* (SRTP) e *Secure* RTCP (SRTCP), os quais são extensões para os protocolos RTP e RTCP, respectivamente. Com esses padrões é possível realizar a criptografia, autenticação e integridade no transporte dos dados de mídia em aplicações unidifusão e multidifusão (BAUGHER et al., 2004).

A utilização do SRTP ou SRTCP é opcional ao uso do RTP ou RTCP, mas se forem utilizados em conjunto, todos os recursos de segurança podem ser separadamente habilitados ou desabilitados. A única exceção é o recurso de mensagem de autenticação a qual é requerida quando se utiliza o SRTCP (BAUGHER et al., 2004).

2.6.2.1 Pacote - Secure RTP

O formato da mensagem do SRTP é apresentado na Figura 2.12. Pode-se observar que somente o *payload* do RTP é encriptado (incluindo qualquer RTP *padding*, se presente). Nota-se também que a criptografia não adiciona nenhum *padding*, ou seja, o tamanho do *payload* do RTP não é aumentado com a criptografia (BAUGHER et al., 2004).

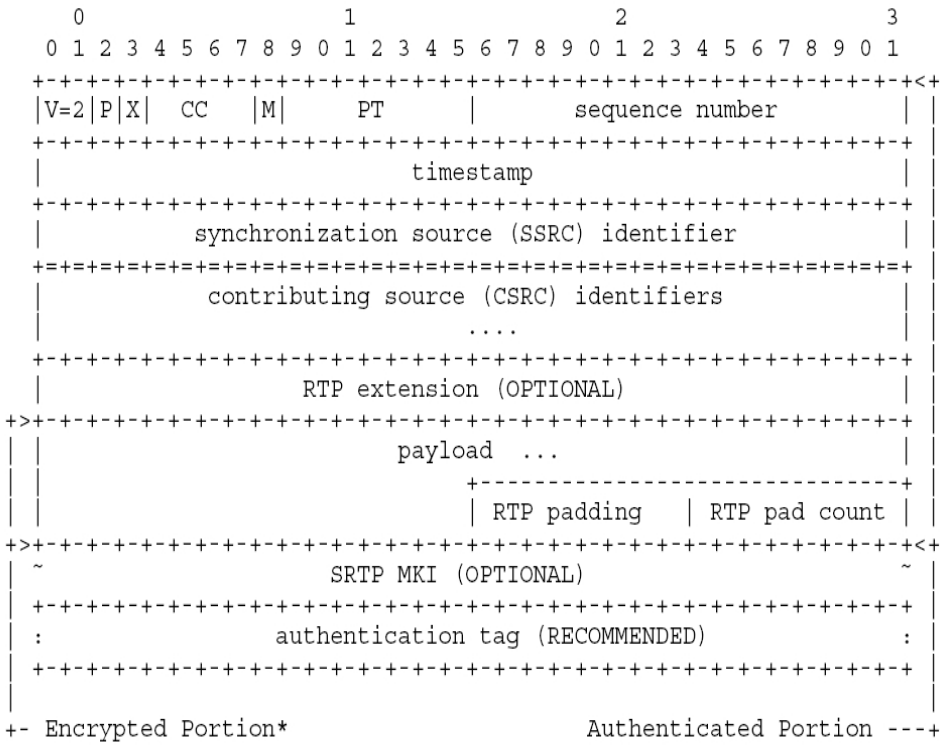

Figura 2.12: Formato do Pacote SRTP (BAUGHER et al., 2004).

O *Master Key Identifier* (MKI) e o *Authentication tag* são os únicos campos adicionados pelo SRTP ao formato original do pacote RTP. O MKI é opcional e identifica a chave mestre da qual a chave de sessão foi derivada. Esse campo pode ser utilizado pelo receptor para recuperar a chave mestre correta quando existe a necessidade de um evento de *re-keying*.

A *tag* de autenticação é um *checksum* encriptado computado pelo cabeçalho e o corpo do pacote RTP. O seu uso é altamente recomendado, uma vez que, protege o pacote contra modificações não autorizadas (BAUGHER et al., 2004).

O campo *sequence number* (16 bit), já presente em um pacote RTP é utilizado em conjunto com um *rollover counter* (ROC - 32bit), o qual é parte do contexto criptográfico para a sessão SRTP prevenir ataques de repetições (BAUGHER et al., 2004).

2.6.2.2 Pacote - Secure RTCP

A Figura 2.13 mostra que os pacotes do RTCP são seguros de modo similar aos pacotes do RTP, mas com a diferença da obrigatoriedade no uso da *tag* de autenticação. Caso contrário seria possível a um atacante, por exemplo, encerrar indevidamente um *stream* RTP com o envio de um pacote BYE (BAUGHER et al., 2004).

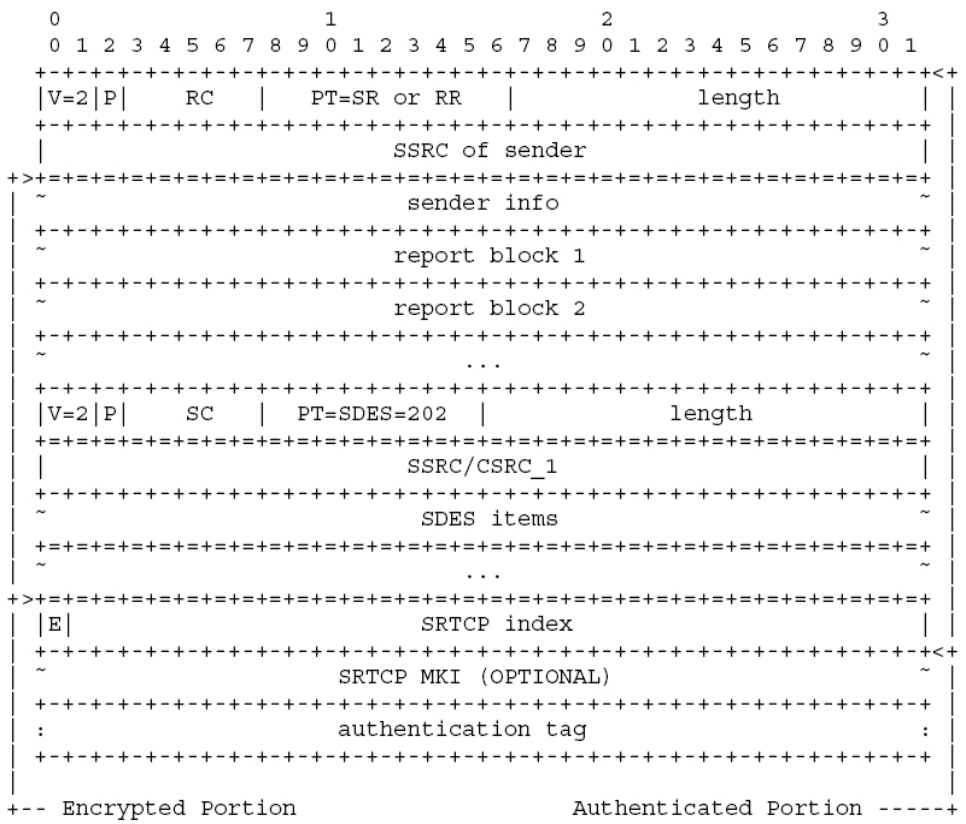

Figura 2.13: Formato do Pacote SRTCP (BAUGHER et al., 2004).

Um campo adicional no pacote é o SRTCP *index*, o qual é utilizado com um contador de seqüência para prevenir ataques de repetição. O inicio do campo de *index* é usado como um *Encryption flag* (E) que indica se o corpo do RTCP está encriptado (BAUGHER et al., 2004).

2.6.2.3 Algoritmos de Criptografia e Autenticação

Em princípio qualquer método de criptografia e autenticação podem ser utilizados com o SRTP. Como padrão são definidos os algoritmos NULL *cipher* (não confidencial) e o *Advanced Encryption Standard* em modo de contador (AES-CTR) e o HMAC-SHA-1 (KRAWCZYK, et al. 1997), baseado na função *hash* SHA de 160 bits, para criptografia e autenticação, respectivamente (BAUGHER et al., 2004).

Esses métodos usam chaves simétricas de sessão que devem ser conhecidas por todos os agentes de usuários participantes em uma sessão SIP. Isso leva ao problema lógico de geração e distribuição de chaves de sessão. O SRTP oferece uma solução parcial,

derivando todas as chaves de sessão necessárias de uma chave mestre comum, mas deixa em aberta a distribuição da chave mestre (BAUGHER et al., 2004).

Existem algumas propostas para a distribuição da chave mestre para os agentes de usuários de uma sessão. Entre elas, há a proposta da utilização da *Multimedia Internet KEYing* (MIKEY) (ARKKO, et al. 2003) para estabelecer um contexto de criptografia no SRTP. MIKEY é um protocolo de troca similar ao IPsec's *Internet Key Exchange* (IKE).

Uma outra opção, para a transmissão da chave mestre, é o uso do parâmetro de chave (k) definido pelo *Session Description Protocol* (SDP). Contudo, deve-se notar que a transmissão da chave mestre em texto limpo (*cleartext*) pode causar um severo risco de segurança. Baseado na suposição que pode-se confiar no servidor de *proxy*, um SIP INVITE pode ser encriptado completamente no *hop-by-hop* utilizando TLS - *Transporte Layer Security* (BLAKE-WILSON, et al. 2003) ou IPsec. Se for necessário a confidencialiadade de fim-a-fim o corpo MIME do SDP necessitará da utilização de uma proteção como o S/MIME (Seção 2.6).

2.6.3 IPsec

Como alternativa aos métodos de proteção implementados no nível de aplicação existe a possibilidade da aplicação de mecanismos de segurança em níveis mais baixos. Um exemplo consiste na utilização do protocolo IPsec para proteção tanto da sinalização como da mídia (KUHN; WALSH; FRIES, 2005).

O IPsec consiste em um padrão que, por meio de criptografia e autenticação, proporciona segurança nas comunicações IP. IPsec é dividido em duas partes principais. A primeira parte define dois novos cabeçalhos independentes: *Authentication Header* (AH) e *Encapsulation Security Payload* (ESP), os quais possibilitam a verificação de confidencialidade, integridade e autenticação nas transmissões. A segunda parte, o ISAKMP (*Internet Security Association and Key Management Protocol*) trata do estabelecimento das chaves (KENT; ATKINSON, 1998a).

Apesar de ser um padrão na camada de rede, o IPsec é orientado a conexão. Isso se deve a necessidade da utilização de uma chave por um determinado período de tempo. Uma conexão no contexto do IPsec é chamada de *Secuity Associantion* (SA) (KENT; ATKINSON, 1998a).

Foram definidos dois modos de utilização do IPsec: modo de transporte e modo de túnel. No modo de transporte um cabeçalho IPsec é inserido no *payload* IPv4 antes do cabeçalho TCP. Esse cabeçalho é formado por informações de segurança, tais como o identificador SA e, possivelmente, uma verificação de integridade de carga útil. A presença desse cabeçalho é identificada alterando-se o campo *Protocol* do cabeçalho IP (KENT; ATKINSON, 1998a).

No modo de túnel, todo o conteúdo do pacote IP é encapsulado e um pacote com um novo cabeçalho IP. Com esse modo, é possível, por exemplo, que diversas conexões TCP sejam encapsuladas em uma única transmissão segura, impedindo a identificação dos emissores e receptores originais (Figura 2.14) (KENT; ATKINSON, 1998a).

Nas próximas sessões serão apresentados os dois novos cabeçalhos descritos pelo IPsec

2.6.3.1 AH - Authentication Header

O cabeçalho *Authentication Header* possibilita verificações de integridade e segurança contra ataques de repetição, não oferecendo sigilo para os dados. A Figura 2.15 apresenta o cabeçalho AH, no modo de túnel, inserido no IPv4, cujos campos são descritos a seguir

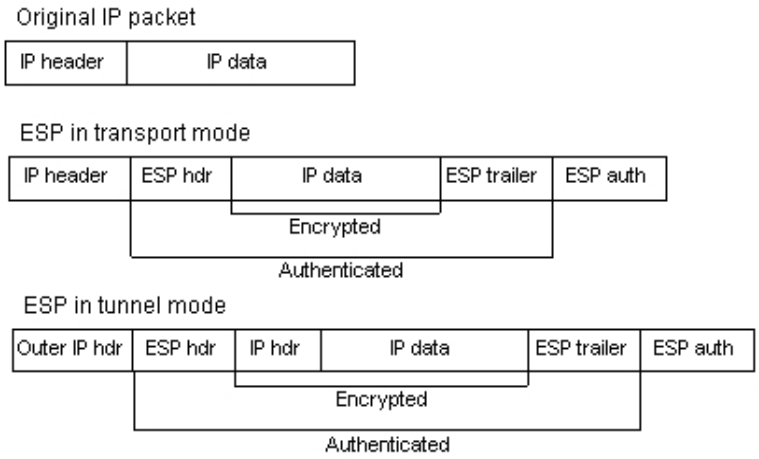

Figura 2.14: Cabeçalhos ESP nos modos de transporte e túnel (KENT; ATKINSON, 1998a).

(KENT; ATKINSON, 1998b):

0	1	2	3
0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
Next Header	Payload Length	RESERVED	
Security Parameters Index (SPI)			
Sequence Number			
Authentication Data (variable)			

Figura 2.15: Formato do cabeçalho AH (KENT; ATKINSON, 1998b)).

- *Next header* - Utilizado para armazenar o valor original do campo *Protocol*, o qual é alterado para indicar a presença do cabeçalho AH;
- *Payload len* - Define o número de palavras de 32 bits (menos duas unidades) que forma o cabeçalho AH;
- *Security parameters index* - Identificador da conexão definido pelo transmissor. Esse campo aponta para um registro no receptor, o qual contém, dentre outras informações da conexão, a chave compartilhada usada na conexão;
- *Sequence number* - Numera os pacotes enviados em uma SA. Cada pacote, mesmo retransmitido, recebe um número exclusivo. Isso é utilizado para evitar ataques de repetição;
- *Authentication data* (HMAC) - Campo de tamanho variável que contém a assinatura digital da carga útil. Para isso, depois do estabelecimento da SA, os dois lados da transmissão negociam o método de assinatura digital. Um modo de assinatura consiste na utilização da chave compartilhada, a qual é somada a um hash

sobre o pacote. Esse esquema é chamado de *Hashed Message Authentication Code* (HMAC).

2.6.3.2 ESP - Encapsulation Security Payload

O cabeçalho AH não realiza criptografia nos dados não fornecendo sigilo na transmissão. Para isso, existe a alternativa da utilização do cabeçalho *Encapsulation Security Payload* (ESP). Além de proporcionar sigilo, esse cabeçalho também apresenta verificações de integridade e segurança contra ataque de repetição (KENT; ATKINSON, 1998c).

A Figura 2.16 apresenta o formato do *payload* IP com a utilização do ESP.

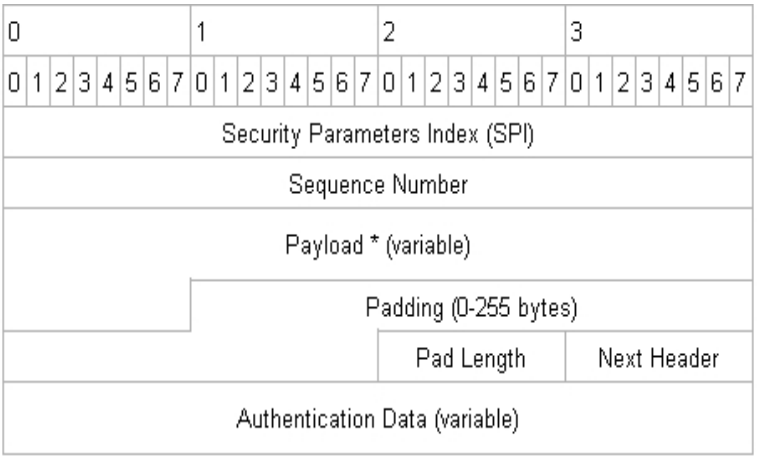

Figura 2.16: Formato do Pacote IP com ESP (KENT; ATKINSON, 1998c).

Os campos *Security parameters index* e *Sequence number* são equivalentes aos presentes no AH. Para verificações de integridade no ESP também é utilizado o HMAC. Entretanto ao invés de serem incluídas no cabeçalho, essas verificações (*Authentication data*) são adicionadas no fim da carga útil, facilitando com isso a sua implementação em *hardware*. Os demais campos consistem em (KENT; ATKINSON, 1998c)):

- *Payload* - Carga útil a ser transmitida;
- *Padding* - Campo de tamanho variável (0 -255 bytes) utilizado para preencher o tamanho do Payload com o objetivo de se adequar a alguma necessidade, tal como blocos fixos para criptografia.
- *Pad Length* - Define a quantidade de bytes preenchidos no *Padding*;
- *Next Header* - Identifica o tipo de dados contido no *Payload*.

Uma palavra adicional (*Initialization vector*) é utilizada quando é aplicada criptografia nos dados, a qual é opcional. Esse valor não está presente no cabeçalho, mas é acrescentado ao *Payload*.

Vários algoritmos podem ser utilizados para a aplicação dos serviços de segurança do ESP. Entretanto uma implementação do ESP deve, obrigatoriamente, suportar os seguintes métodos (KENT; ATKINSON, 1998c):

- 3DES em CBC;
- HMAC com MD5;

- HMAC com SHA-1;
- Algoritmo de autenticação NULL;
- Algoritmo de criptografia NULL.

Os algoritmos de autenticação e criptografia NULL correspondem ao suporte da não utilização desses métodos de segurança.

2.7 Degradação com Segurança em VoIP

Apesar das diversas soluções para obtenção de segurança na arquitetura VoIP, essa apresenta-se como uma tarefa complexa devido as especificidades da arquitetura e principalmente aos critérios de qualidade do serviço (WALSH; KUHN, 2005)(KUHN; WALSH; FRIES, 2005)(VOIPSA, 2005)(BARBIERI; BRUSCHI; ROSTI, 2002)(HONG et al., 2004).

Soluções convencionais de segurança muitas vezes não são aplicáveis em VoIP. Um exemplo é a utilização de *firewalls*, os quais normalmente não suporta portas dinâmicas, tais como as utilizadas no RTP. Outra dificuldade é a aplicações de IPSec em modo fim-a-fim nas sinalizações, que impossibilita aos componentes intermediários da arquitetura VoIP, tais como os SIP *Proxyes*, terem acesso às informações das mensagens SIP para correta transmissão das mensagens (KUHN; WALSH; FRIES, 2005).

Um aspecto que deve ser cuidadosamente avaliado na elaboração de soluções de segurança para VoIP são as premissas de qualidade dessa tecnologia (WALSH; KUHN, 2005). Mecanismos com bastante processamento, como criptografia, autenticação ou vistoria dos *streams* por *proxyes*, podem acrescentar atrasos ou *jitter* e com isso degradar as conversações. A adição de novas informações de segurança podem também degradar a performance do uso efetivo da banda (BARBIERI; BRUSCHI; ROSTI, 2002).

Nas próximas seções serão apresentados alguns aspectos em relação à aplicação de segurança em VoIP. Enfocou-se na avaliação de segurança na mídia devido aos critérios de qualidade não serem significativos na sinalização (KUHN; WALSH; FRIES, 2005). Assim, para cada aspecto foram mostrados os seus impactos nos protocolos SRTP e IPSec, os quais são comumente aplicados para segurança da mídia em aplicações VoIP.

2.7.1 Atraso

Como visto na Seção 2.3 o atraso em demasiado prejudica a interatividade nas conversações provocando uma sensação de desconforto aos participantes.

Mecanismos de segurança como a criptografia e autenticação realizam diversas manipulações nos pacotes, o que pode resultar em um aumento no atraso total de fala-e-escuta. Esse atraso varia em relação ao algoritmo de criptografia e autenticação utilizado (BARBIERI; BRUSCHI; ROSTI, 2002).

Os protocolos IPSec e SRTP diferenciam-se no modo de aplicação dos métodos de segurança. No IPSec é aplicada criptografia em todo *payload* IP, ou seja, além do *payload* de voz a cifragem é aplicada aos cabeçalhos UDP e RTP. A autenticação é aplicada no *payload* encriptado e nos cabeçalhos de segurança do IPSec: ESP e AH (KENT; ATKINSON, 1998a). Já no SRTP, aplica-se a criptografia somente no *payload* de voz RTP e a autenticação em todo pacote RTP (BAUGHER et al., 2004). Isso demonstra que no IPSec um número maior de bits são processados (HONG et al., 2004).

Em (BARBIERI; BRUSCHI; ROSTI, 2002) e (HONG et al., 2004) foram apresentadas avaliações do impacto em VoIP da aplicação de criptografia e autenticação no fluxo de áudio das conversações. Na Tabela 2.8 são apresentados resultados obtidos comparando-se os métodos de criptografia AES e 3DES e os algoritmos de *hash* SHA-1 e MD5.

Tabela 2.8: Atrasos de diferentes algoritmos de criptografia e autenticação em VoIP.

Método de Criptografia	**GSM-1 (33 octetos)**	**GSM-4 (132 octetos)**	**G.711-1 (240 octetos)**	**G.711-5 (1200 octetos)**
AES/SIC	Enc: 0.08 ms Dec: 0.08 ms	Enc: 0.12 ms Dec: 0.12 ms	Enc: 0.16 ms Dec: 0.16 ms	Enc: 0.52 ms Dec: 0.52 ms
AES/f8	Enc: 0.11 ms Dec: 0.11 ms	Enc: 0.16 ms Dec: 0.16 ms	Enc: 0.19 ms Dec: 0.19 ms	Enc: 0.52 ms Dec: 0.52 ms
3DES/CBC	Enc: 0.48 ms Dec: 0.48 ms	Enc: 0.80 ms Dec: 0.80 ms	Enc: 0.94 ms Dec: 0.94 ms	Enc: 2.5 ms Dec: 2.5 ms
HMAC/MD5	Gen: 0.09 ms Ver: 0.09 ms	Gen: 0.11 ms Ver: 0.12 ms	Gen: 0.15 ms Ver: 0.15 ms	Gen: 0.27 ms Ver: 0.28 ms
HMAC/SHA-1	Gen: 0.21 ms Ver: 0.23 ms	Gen: 0.27 ms Ver: 0.29 ms	Gen: 0.30 ms Ver: 0.31 ms	Gen: 0.55 ms Ver: 0.55 ms

Fonte: (HONG et al., 2004).

O métodos AES/SIC e HMAC/MD5 apresentaram um melhor resultado. Contudo, pode-se notar que os atrasos resultantes aos algoritmos não apresentam valores significativos (HONG et al., 2004).

2.7.2 Escalabilidade

Embora, conforme (HONG et al., 2004), os métodos de criptografia e autenticação não apresentarem um atraso substancial em VoIP é necessário observar a escalabilidade desses mecanismos em ambientes sobrecarregados (BARBIERI; BRUSCHI; ROSTI, 2002).

Conforme o tráfego aumenta em relação ao *througput* dos processos de criptografia e autenticação, esses dispositivos podem não ser capazes de processar todos os dados de entrada, ocasionando, com isso, o *jitter* ou até o descarte de pacotes. Esse problema se agrava quando são protegidos tráfegos heterogêneos em conjunto aos pacotes de VoIP (BARBIERI; BRUSCHI; ROSTI, 2002).

Esse tipo de comportamento é encontrado em cenários quando um túnel IPsec é compartilhado para a transmissão de dados de aplicações em tempo real, como o VoIP, em conjunto com tráfegos de aplicações heterogêneas, tal como a transferência de arquivos. Assim, nesse tipo de ambiente o processo do IPSec apresenta-se como um ponto de convergência, podendo tornar-se um gargalo para as transmissões (BARBIERI; BRUSCHI; ROSTI, 2002).

Na Figura 2.17 é ilustrado o comportamento de vários tráfegos utilizando um mesmo processo de IPsec. Os quadros menores representam pacotes de VoIP, os quais são alocados em uma fila FIFO (*Firts-In First-Out*) em conjunto aos demais pacotes (representados pelos quadros maiores) na espera para o acesso ao IPsec. O *jitter* decorrente desse tipo de cenário corresponde a variação entre os tempos de atendimento dos pacotes VoIP, representados por T1 e T2.

Em (BARBIERI; BRUSCHI; ROSTI, 2002) foi apresentada uma avaliação em relação

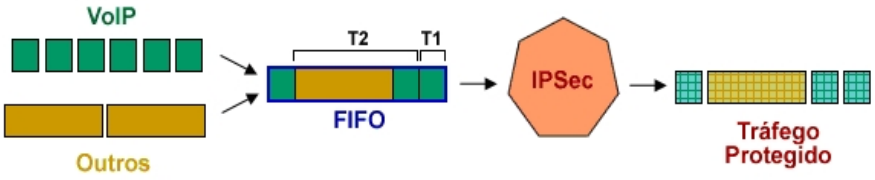

Figura 2.17: Ilustração do compartilhamento do IPSec por tráfegos heterogêneos.

ao *jitter* decorrente da aplicação do IPSec. Na Figura 2.18 são mostradas as variações de chegada dos pacotes protegidos pelo IPsec com criptografia 3DES. Na imagem (a) existe apenas o tráfego VoIP e na segunda b), além dos *streams* das conversações, são transmitidos pacotes com 1200 bytes.

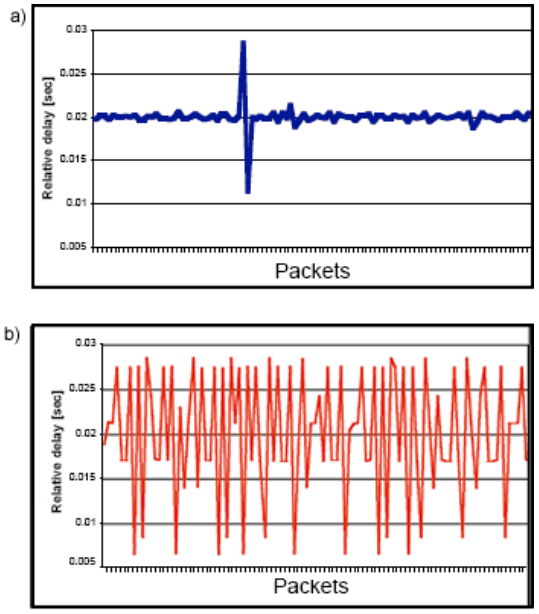

Figura 2.18: Variação do tempo de chegada dos pacotes VoIP sem (a) e com (b) tráfegos concomitantes em um canal protegido pelo IPsec (BARBIERI; BRUSCHI; ROSTI, 2002).

Nota-se que com o aumento de tráfegos heterogêneos no canal, o processo de proteção é saturado e com isso acrescenta uma variação aos pacotes das conversações, o que pode provocar a diminuição da qualidade das conversações.

É importante observar que esse comportamento não é apresentado no protocolo SRTP, o qual é implementado no nível de aplicação e aplica os mecanismos de segurança somente nos pacotes RTP e RTCP (BAUGHER et al., 2004).

2.7.3 Expansão dos Pacotes

O aumento no tamanho dos pacotes pode provocar um maior *overhead* na rede e diminuir o número total de chamadas possíveis (BARBIERI; BRUSCHI; ROSTI, 2002). Esse aumento degrada não somente a ocupação da banda, mas também influência na serialização do pacote, no rotemante, entre outros (CISCO, 2007c).

No IPsec o acréscimo no tamanho do pacote é decorrente à inserção dos novos cabeçalhos AH/ESP. Além disso, pode ser necessária a adição de *padding* ao *payload*

quando preciso a adequação do tamanho do pacote aos métodos que trabalham com um número fixo de bytes, como a criptografia em blocos (KENT; ATKINSON, 1998a).

No SRTP são adicionados um número menor de informações de segurança (Seção 2.6.2) (HONG et al., 2004).

Na Figura 2.19 é apresentada uma comparação do tamanho dos pacotes com um *payload* contendo um *frame* de 20 octetos. Foram observados pacotes sem a aplicação de segurança (sem e com compressão cRTP (KOREN et al., 2003)); pacotes utilizando o protocolo IPsec (em modo de túnel e de transporte); e pacotes aplicando-se o protocolo SRTP.

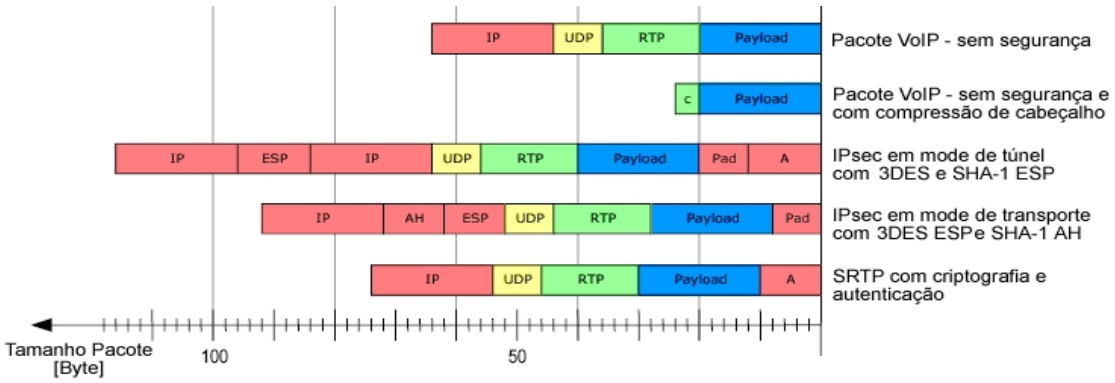

Figura 2.19: Expansão no tamanho do pacote (HONG et al., 2004).

Como pode ser observado, o IPSec apresenta um acréscimo maior que o obtido com o SRTP.

2.8 Métodos para diminuição do impacto da segurança em VoIP

Como apresentado nas seções anteriores o SRTP apresenta-se como uma melhor alternativa, em relação à utilização do IPsec, na aplicação de segurança em VoIP com qualidade de serviço. O SRTP apresenta um framework com baixo custo computacional e de banda, evita uma expansão grande dos pacotes e utiliza algoritmos de criptografia modernos (HONG et al., 2004)(BAUGHER et al., 2004).

Entretanto, para a utilização do SRTP é necessária a alteração dos aplicativos atuais (KUHN; WALSH; FRIES, 2005). O protocolo IPSec, por sua vez, permite a construção de túneis entre duas sub-redes o que possibilita a proteção, no canal protegido, de todos os terminais em seus domínios. Adicionalmente, o IPSec é amplamente utilizado na construção de VPNs, onde comumente são protegidas, além da voz, os tráfegos de dados heterogêneos (TANENBAUM, 2003).

Desse modo, enfocou-se na avaliação do impacto da aplicação do IPSec para proteção das ligações VoIP.

A degradação mais significativa apresentada com a aplicação do IPSec em VoIP consiste na expansão dos pacotes (BARBIERI; BRUSCHI; ROSTI, 2002). Abordagens para minimizar essa problemática consistem em, basicamente, duas soluções: compressão dos cabeçalhos e multiplexação dos pacotes.

2.8.1 Compressão de Cabeçalhos

A técnica de compressão baseia-se no fato em que várias informações dos pacotes não sofrem alterações durante toda a sessão. Assim, são mantidos contextos entre o emissor e

o receptor, submetendo somente o que se diferencia entre eles (KOREN et al., 2003).

Nesses contextos são armazenados os dados constantes dos cabeçalhos, como o endereço IP e as portas UDP, e são enviados somente as informações que se diferenciam para cada pacote, como os campos do RTP *Timestamp* e o *Sequence Number*. É possível ainda reduzir o número de bytes enviados, no momento em que define-se um valor base e são encaminhados apenas a diferença (*delta*) para esse valor (JONSSON, 2004).

Na Figura 2.20 é ilustrado um exemplo da redução do envio de informações mantendo-se um contexto entre as duas partes envolvidas.

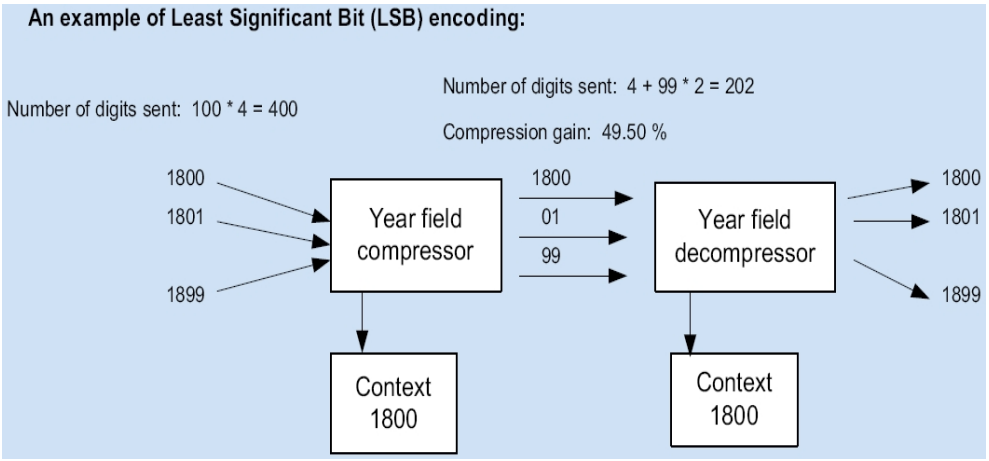

Figura 2.20: Ilustração de compressão.

Nesse exemplo, é transmitida uma seqüência de anos. Ao invés de enviar o valor completo em todas as mensagens, inicialmente é informado o século (1800) e posteriormente envia-se somente o ano. Esse exemplo pode ser aplicado no campo *Sequence Number* do RTP, definido o valor inicial e posteriormente encaminhando somente a diferença.

Os protocolos mais utilizados são o RoHC (*Robust Header Compression*) (JONSSON, 2004) e eCRTP (*enhanced Compressed* RTP) (KOREN et al., 2003). O primeiro apresenta-se como a solução mais completa e alcança uma maior taxa de compressão e robustez a erros. Entretanto, esse protocolo contém diversos mecanismos e métodos que torna complexa a sua implementação (KOREN et al., 2003). O outro protocolo (eCRTP) contém uma implementação mais simples e é comumente utilizado (BARBIERI; BRUSCHI; ROSTI, 2002).

2.8.2 Multiplexação

Em determinados cenários, diversas ligações são realizadas ao mesmo tempo entre trechos em comum. Assim, ao invés de se enviar em pacotes separados, os fluxos de voz de cada ligação podem ser multiplexados em um único pacote. Com isso, pode-se evitar o reenvio de dados, diminuindo o *overhead* na rede.

Na multiplexação os pacotes são obtidos em um ponto de convergência da rede, como, por exemplo, em *gateways* entre uma rede PSTN e IP, e são agrupados para formação de um único *payload*. O pacote resultante é enviado para um ponto responsável pela restauração dos fluxos originais. A Figura 2.21 representar um exemplo de multiplexação em que as ligações de uma rede A são multiplexados e transmitidos à rede B por meio de apenas um fluxo de pacotes.

Essa técnica pode ser aplicada em diferentes níveis na camada de protocolos e com diferentes técnicas. A idéia básica consiste em encapsular diversos fluxos de voz como

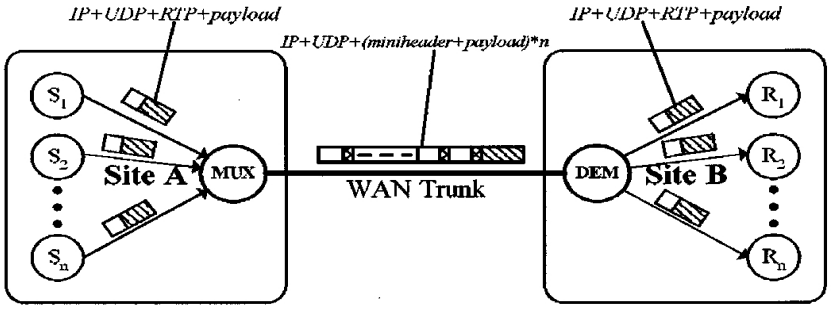

Figura 2.21: Multiplexação de ligações VoIP.

um único *payload* e utilizar alguma técnica para a identificação de cada contribuição. Também podem ser aplicados métodos para a compressão dos cabeçalhos multiplexados, o que permite maior redução na taxa de dados enviados.

Nas próximas seções serão apresentados alguns exemplos de técnicas de multiplexação em VoIP.

2.8.2.1 Modos de Multiplexação

Uma primeira abordagem realiza a multiplexação de modo fim-a-fim no protocolo RTP. Existem várias propostas de implementação dessa alternativa, tais como apresentadas em (EL-KHATIB et al., 2000)(ROSENBERG; SCHULZRINNE, 1999)(TANIGAWA; HOSHI; TSUKADA, 1999). Em (SUBBIAH; SENGODAN, 1999) é sugerida a criação de mine cabeçalhos para identificação de cada fluxo (Figura 2.22).

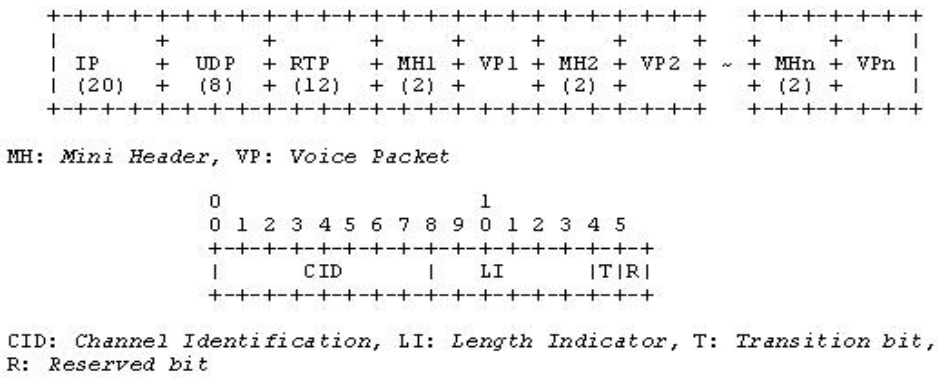

Figura 2.22: a) Formato de pacote com multiplexação dentro do RTP. b) Formato do mine-cabeçalho (MH) (SUBBIAH; SENGODAN, 1999).

Uma segunda abordagem associa portas UDP específicas para a multiplexação dos *streams* (TRAD; AFIFI, 2004).

Pode-se também multiplexar os pacotes dentro de túneis e, com isso, evitar o *overhead* resultante com o tunelamento. A multiplexação do protocolo IPSec em modo de túnel apresenta-se como um exemplo dessa variação (Figura 2.23).

Outra abordagem permite a multiplexação de modo ponto-a-ponto, tal como a utilização da multiplexação do protocolo PPP-Mux (PAZHYANNUR; ALI; FOX, 2001) (Figura 2.24). Esse exemplo de multiplexação é utilizado em (THOMSPON; KOREN; WING, 2005), onde é apresentada uma abordagem em que os pacotes são comprimidos com o protocolo cRTP (KOREN et al., 2003) e multiplexados pelo protocolo PPP-Mux.

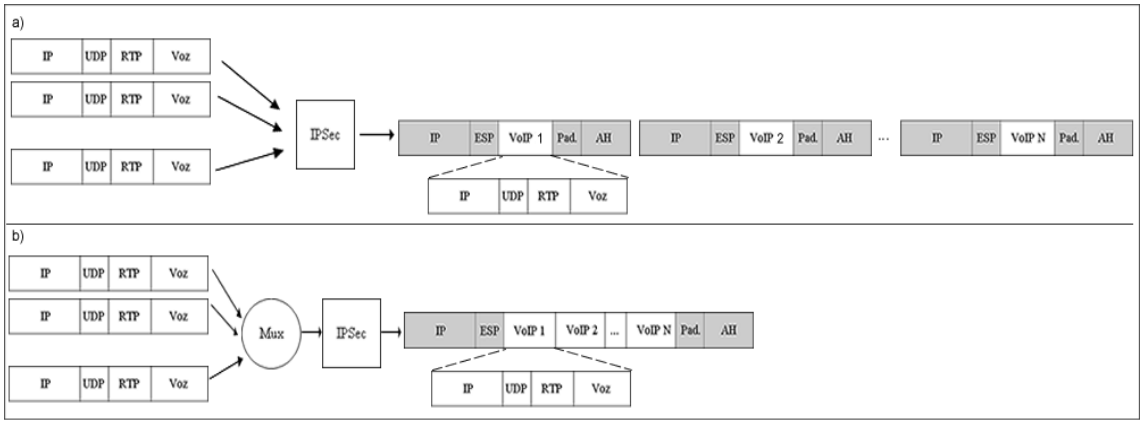

Figura 2.23: a) Fluxo convencional de ligações VoIP sobre IPSec. b) Fluxo de ligações VoIP multiplexados sobre IPSec.

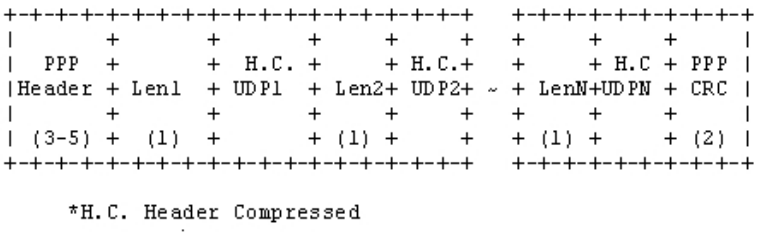

Figura 2.24: Exemplo de multiplexação em um frame PPP (PAZHYANNUR; ALI, 1999).

2.8.2.2 Desempenho

A técnica de multiplexação apresenta fatores positivos e negativos em relação a outros métodos de compressão. Ela consiste em uma técnica com boa escalabilidade, em que aumentando o número de ligações mantém-se constante o número de pacotes enviados, o que evita, entre outros fatores, o *overhead* na fila dos roteadores.

Com a multiplexação pode-se evitar o aumento em demasiado do tempo de empacotamento, uma vez que, ao contrário do método de agrupamento de *frames* (YAMADA; FUKUMOTO, 2006), não é necessário aguardar a amostragem completa de um novo *frame* para a multiplexação do próximo pacote. Outra vantagem da multiplexação é o fato de que o pacote final é formado por um ou poucos *frames* de uma mesma ligação, assim no caso da perda de pacote o impacto final será menor (Figura 2.25).

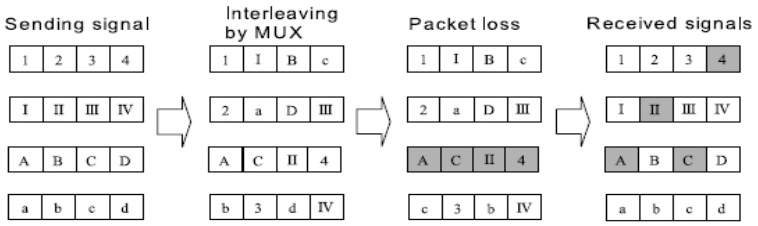

Figura 2.25: Exemplo da perda de pacotes multiplexados (YAMADA; FUKUMOTO, 2006).

Uma desvantagem dessa técnica consiste no tempo de parada-e-spera (*stop-and-wait*). Esse fator refere-se ao tempo de espera para o agrupamento dos pacotes. Na Figura 2.26 é mostrada uma ilustração do processo de amostragem dos pacotes para a formação do

pacote multiplexado.

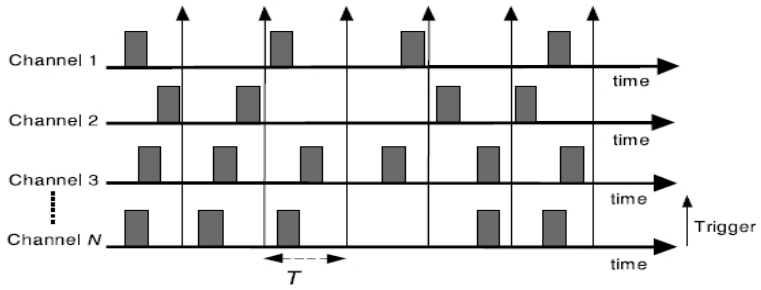

Figura 2.26: Ilustração de várias iterações de agrupamento dos pacotes na multiplexação (YAMADA; FUKUMOTO, 2006).

Nota-se que quanto maior o tempo de agrupamento (T), maior será o pacote multiplexado. Contudo o aumento desse tempo pode inferir negativamente na qualidade das ligações ao aumentar o atraso fim-a-fim.

Em algumas propostas é sugerida a definição desse tempo equivalente ao tempo de um *frame*. Contudo, em alguns cenários essa prerrogativa não é totalmente eficaz.

No próximo capítulo serão apresentados detalhes de um cenário problemático envolvendo a multiplexação em ligações VoIPSec.

2.9 Considerações Finais

Nesse capítulo foi realizada uma visão geral da base teórica do conteúdo abordado nessa dissertação. Nos próximos capítulos serão detalhados a motivação e a contribuição proposta.

3 PROBLEMÁTICA

3.1 Considerações Iniciais

A técnica de multiplexação mostra-se como uma alternativa para diminuição do *overhead* na rede em aplicações de VoIPSec. Entretanto, em cenários onde encontram-se poucas ligações simultâneas, a aplicação dessa técnica pode ser desencorajada. Isso se deve ao fato da performance da multiplexação estar diretamente relacionada com o número de pacotes agrupados. Com um número reduzido de ligações concomitantes os modos tradicionais de multiplexação não alcançam uma redução efetiva do *overhead* na rede.

Existe como alternativa para melhoria do desempenho da multiplexação, nesse tipo de cenário, a abordagem de expansão do tempo de agrupamento, o que aumentaria o número de pacotes multiplexados.

Porém, a expansão do atraso fim-a-fim com o aumento do tempo de retenção pode degradar a qualidade das ligações.

Assim, apresenta-se uma conjuntura onde a possibilidade de aumentar o desempenho da multiplexação em cenários pequenos esbarra na necessidade de garantia de qualidade nas ligações. Nas próximas sessões será detalhada essa problemática.

3.2 Multiplexação em Cenários Reduzidos

O ganho de um mecanismo de multiplexação pode ser medido com a razão entre o número de bytes do pacote multiplexado e a quantidade de bytes necessária para transmissão da mesma informação em pacotes separados.

Em VoIPSec a multiplexação pode ser realizada agrupando-se pacotes de uma mesma SA (*Security Association*), compartilhando os mesmo cabeçalhos ESP/AH (Figura 2.23).

A Equação 3.1 expressa uma formulação do ganho nesse tipo de aplicação.

$$Ganho = 1 - \frac{h + \sum_{i=1}^{n}(c_i + p_i)}{H * n + \sum_{i=1}^{n}(pi)} \tag{3.1}$$

Sendo h igual ao tamanho do cabeçalho do pacote multiplexado, c_i e p_i os tamanhos dos cabeçalhos originais da enésima ligação, H o tamanho do cabeçalho para o envio dos pacotes separadamente e n o número de pacotes agrupados.

Quanto mais pacotes agrupados, maior é a redução do *overhead* na rede. Essa relação pode ser melhor observada se considerarmos um ambiente hipotético, onde a multiplexação é realizada em ligações com o mesmo tamanho de *payload* e de cabeçalho. Assim, pode-se derivar a Equação 3.2 com a seguinte formulação:

$$Ganho' = 1 - \frac{h + (c + p) * n}{(H + p) * n} \tag{3.2}$$

Separando o denominador teremos:

$$Ganho_{max} = 1 - \frac{h}{(H + p) * n} + \frac{(c + p)}{(H + p)} \tag{3.3}$$

Com isso, pode-se observar que o máximo de compressão é equivalente à razão entre o tamanho do pacote original (*payload* do IPSec) e o do pacote quando não aplicada a multiplexação. Esse máximo é aproximado na medida em que o número de pacotes agrupados (n) aumenta em relação ao tamanho do cabeçalho não multiplexado (h).

O número de pacotes (n) depende de diferentes fatores: número de ligações simultâneas, taxa de transmissão do *codec* utilizado em cada ligação e o tempo de agrupamento de pacotes no processo de multiplexação.

Na Figura 3.1 é ilustrado o ganho com a multiplexação variando o número de ligações simultâneas e o tempo de agrupamento (T_{mux}).

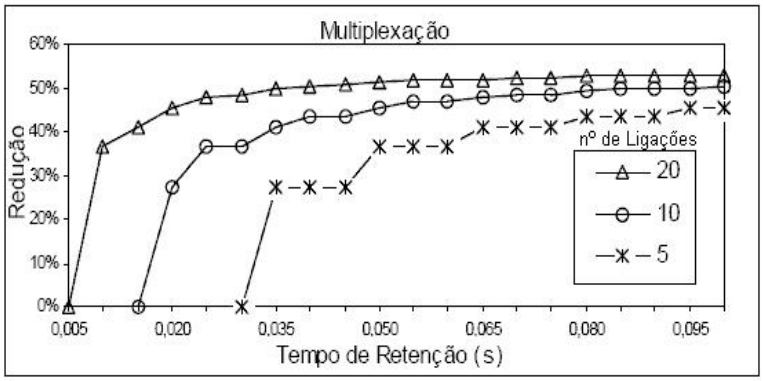

Figura 3.1: Multiplexação de ligações VoIP.

Observa-se que quanto mais ligações simultâneas, menor é o tempo necessário de parada-e-espera. Algumas soluções definem o tempo de retenção igual a um *frame* de voz. Entretanto, existem cenários que essa alternativa não é eficaz.

Em ambientes reduzidos, com restrições de banda, é fundamental a aplicação de métodos para melhoria da ocupação da banda. Contudo, o número reduzido de ligações simultâneas torna pouco eficiente a aplicabilidade da técnica de multiplexação.

Como alternativa, há possibilidade de se expandir o tempo em que o processo de multiplexação agrupa os pacotes, retendo um número maior de pacotes de uma mesma ligação (YAMADA; FUKUMOTO, 2006). O resultado desse procedimento pode ser observado na Figura 3.1, onde em um cenário com 10 ligações simultâneas, é possível aumentar a redução de 30%, com 20 ms de retenção, para aproximadamente 50% definido o tempo de agrupamento em 50 ms.

Entretanto, é importante notar que o aumento do tempo de agrupamento resulta na expansão do tempo de fala-e-escuta, o qual pode degradar a qualidade das ligações. A determinação do tempo máximo que pode-se acrescentar ao tempo de fala-e-escuta está relacionada com diversos fatores, tais como o *codec* utilizado, o tempo de propagação na rede, a taxa de perde de pacotes, entre outros (INTERNATIONAL TELECOMMUNICATION UNION, 2005a). Na Figura 3.2 pode-se observar a influência de alguns desses fatores com a variação do atraso de fala e escuta.

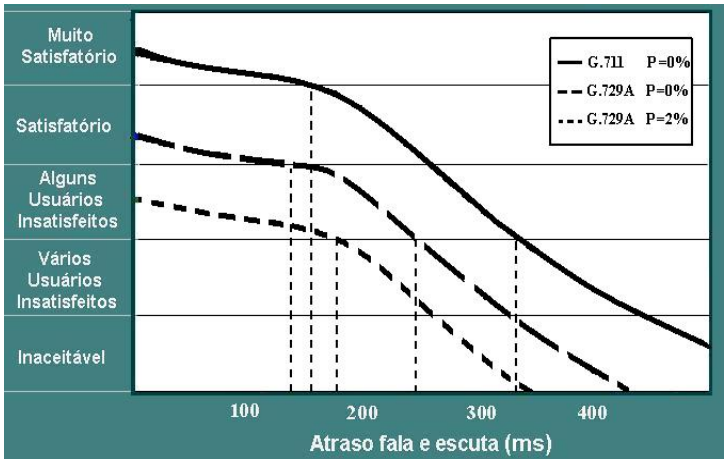

Figura 3.2: Qualidade das conversações com diferentes fatores degradantes Adaptado de (TELECOMMUNICATIONS INDUSTRY ASSOCIATION, 2006).

Pode-se verificar o impacto em relação à satisfação dos usuários entre as seguintes classificações: "Muito Satisfatório", "Satisfatório", "Alguns Usuários Insatisfeitos", "Vários Usuários Insatisfeitos"e "Inaceitável"(INTERNATIONAL TELECOMMUNICATION UNION, 2005a). Em um cenário de referência, onde não se encontra impacto com o *codec* utilizado (G.711) e sem perdas de pacotes (P=0%), é possível manter o nível em Satisfatório com atrasos entre 175 ms à 350 ms. Se alterarmos o *codec* para G.729A os limiares reduzem para 150 ms à 250 ms. Em cenários com perda de pacotes (P) de 2%, os limiares diminuem significantemente, ficando com máximo abaixo de 200 ms.

Isso demonstra que a alternativa de expandir o tempo de retenção para aumentar a eficácia da multiplexação em VoIPSec necessita ser avaliado sobre diferentes fatores para evitar a degradação das ligações em níveis abaixo do desejável.

Adicionalmente, é importante notar que cada ligação multiplexada pode percorrer redes heterogêneas. Assim, o acréscimo do tempo de retenção dos pacotes pode apresentar um impacto diferente para cada uma das conversações estabelecidas.

Desse modo, além de observar os diferentes parâmetros, uma solução robusta deve considerar de modo individual cada ligação envolvida na multiplexação (PEREIRA; TAROUCO, 2007).

Toda essa conjuntura demonstra a necessidade de um mecanismo que possibilite a definição de um tempo ideal de parada-e-espera. Este deve possibilitar a redução no envio de dados redundantes, em cenários com poucas ligações simultâneas, e ao mesmo tempo garantir a qualidade de todas as ligações envolvidas na multiplexação. O mesmo ainda deve considerar os diferentes fatores de degradação, de maneira independente, para cada uma das ligações.

Com base nessa problemática foi elaborada uma evolução do método de multiplexação com o tempo de retenção adaptativo baseado em parâmetros de qualidade. Com essa solução pretende-se obter uma maior taxa de compressão com qualidade nas ligações (PEREIRA; TAROUCO, 2007).

3.3 Trabalhos Relacionados

Em (EL-KHATIB et al., 2000)(PAZHYANNUR; ALI, 1999)(ROSENBERG; SCHULZRINNE, 1999)(TANIGAWA; HOSHI; TSUKADA, 1999)(SZE et al., 2002) é

sugerida a definição do tempo de retenção para a multiplexação com um valor fixo equivalente ao tamanho (em ms) de um *payload* VoIP, o qual normalmente é definindo em 10 ms, 20 ms ou 30 ms (INTERNATIONAL TELECOMMUNICATION UNION, 2003b). Essa abordagem pode ser ineficaz na redução do *overhead* com um número pequeno de ligações, tal como pode ser visto na Figura 3.1. Além disso, essa abordagem não é robusta, uma vez que a inserção de atrasos fim-a-fim, mesmo que pequenos, pode ocasionar impacto negativo em redes com fatores degradantes, tais como atrasos longos (ex.: ligações intercontinentais), e descartes de pacotes (ex.: congestionamento ou enlaces com alta taxa de erros).

Abordagens adaptativas são propostas em (TRAD; AFIFI, 2004) e (KIM; KANG; HWANG, 2005). No primeiro trabalho o tempo de retenção é ajustado dinamicamente por meio do mecanismo de controle de taxa de transmissão do TCP-friendly. No trabalho não são considerados todos os fatores relevantes à qualidade da ligação. Teve como objetivo realizar apenas o controle de congestionamento na rede.

Já em (KIM; KANG; HWANG, 2005) o tempo de retenção para a multiplexação é definido de maneira dinâmica com o objetivo de não ultrapassar um atraso máximo desejável. Para isso, utiliza-se do algoritmo TCP RTO (*Retransmission TimeOut*). Embora o objetivo desta abordagem seja evitar a degradação com um atraso excessivo, não são considerados todos os fatores que influenciam a qualidade das ligações, tais como *jitter*, *codec*e descartes de pacotes.

3.4 Considerações Finais

Nesse capítulo foram apresentados os detalhes da problemática abordada neste trabalho. No próximo capítulo será descrita a solução de melhoria proposta.

4 PROPOSTA

4.1 Considerações Iniciais

No Capítulo 3 foram demonstrados os desafios da aplicação eficaz da multiplexação de ligações VoIP sobre IPSec em cenários com um número reduzido de ligações simultâneas. Neste capítulo será apresentada uma proposta de evolução do método de multiplexação com o objetivo de obter uma maior taxa de compressão mantendo níveis de qualidade nas ligações.

4.2 Esquema de Multiplexação Adaptativa Baseada em Parâmetros de Qualidade

O princípio básico da solução proposta consiste em aumentar o tempo de retenção em cenários favoráveis e diminuí-lo nos desfavoráveis.

Em boa conjuntura dos fatores que influenciam o desempenho das ligações, expande-se o tempo de retenção, aumentando o número de pacotes agrupados e conseqüentemente obtendo-se maior taxa de compressão (Figura 4.1). O acréscimo é realizado respeitando os limiares de qualidade de cada ligação envolvida na multiplexação, processo esse que será explicado nas próximas seções.

Em situações críticas, quando o aumento do atraso de fala e escuta torna-se degradante, diminui-se o tempo de agrupamento dos pacotes, evitando a deterioração da qualidade das conversações.

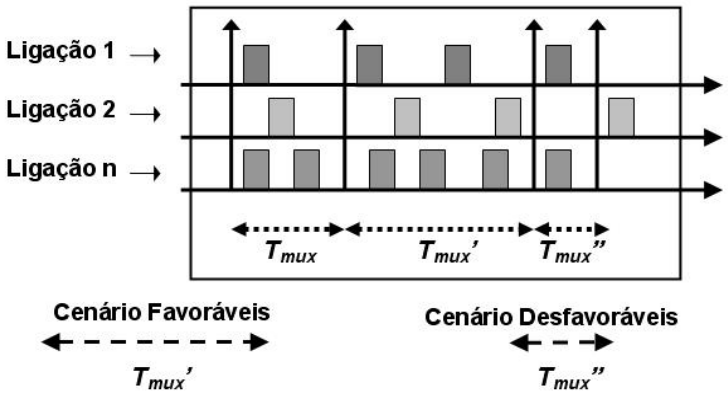

Figura 4.1: Adaptação do tempo de retenção (T_{mux}).

O tempo de retenção é estimado baseado em diversos parâmetros de qualidade de

conversações, utilizando-se para isso as métricas definidas na recomendação do Modelo E (INTERNATIONAL TELECOMMUNICATION UNION, 2005a).

Nas próximas seções são apresentados os detalhes dessa proposta.

4.3 Arquitetura

A arquitetura elaborada é formada, basicamente, por dois componentes: multiplexador adaptativo e monitor de QoS (*Quality of Service*) (Figura 4.2).

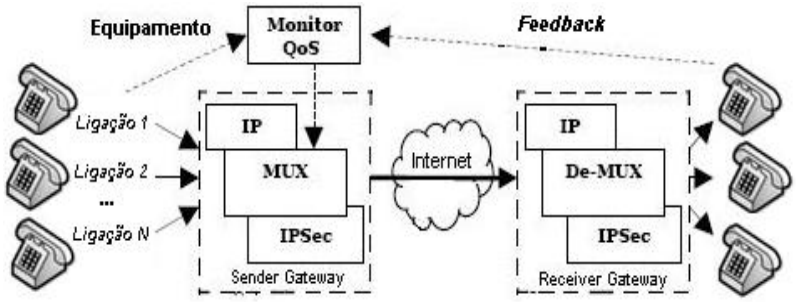

Figura 4.2: Arquitetura Proposta.

O primeiro componente realiza a multiplexação de modo adaptativo dos *streams* de ligações VoIP estabelecidas entre duas sub-redes interligadas por um canal seguro. Sua implementação pode ser realizada antes ou dentro do espaço da camada do IPSec 4.3. Contudo, é necessário aplica-la antes da criptografia e da autenticação, pois não é possível manipular os dados após a aplicação desses métodos (KENT; ATKINSON, 1998a,c,b). Além disso, esse esquema possibilita compartilhar cabeçalhos ESP e/ou AH entre diversas ligações de uma mesma SA (*Security Association*).

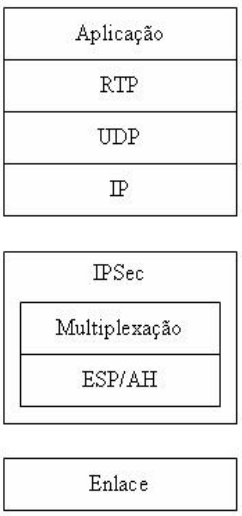

Figura 4.3: Exemplo de implementação da multiplexação adaptativa na pilha de protocolos.

O segundo componente é responsável pelo monitoramento dos parâmetros referentes à qualidade das ligações, os quais são repassados ao módulo de multiplexação para o computo do tempo de parada-e-espera para retenção dos pacotes (Seção 3.2). Esse componente pode ser uma entidade externa, tal como uma aplicação de monitoramento de

QoS, onde são coletados parâmetros dos equipamentos utilizados e das redes fim-a-fim de cada ligação. Essas informações podem ser obtidas por meio dos protocolos RTP (*Real time Transport Protocol*), RTCP (*RealTime Transport Control Protocol* (SCHULZRINNE et al., 2003) e RTCP XP (*Extended Reports*) (FRIEDMAN; CACERES; CLARK, 2003).

4.4 Cálculo do Tempo de Retenção

O tempo máximo de retenção dos pacotes no processo de multiplexação é estimado individualmente para cada ligação estabelecida dentro do canal seguro e é calculado considerando diversos fatores degradantes às ligações. Para isso, utiliza-se a recomendação do Modelo E (INTERNATIONAL TELECOMMUNICATION UNION, 2005a), onde vários parâmetros, como atraso total, perdas de pacotes, eco nas conversações, degradação com equipamentos, entre outros, são computados para a determinação de um valor referente ao nível de satisfação dos usuários em uma ligação (Equação 4.1).

$$R = 93,36 - Ie - Id \tag{4.1}$$

Os parâmetros Ie (*Impairment equipaments*) e Id (*Impairment devices*) representam os fatores degradantes causados pelo equipamento e pela rede, respectivamente. A constante $93,36$ equivale aos demais fatores que influenciam a percepção de qualidade, assumindo-se os valores padrões (INTERNATIONAL TELECOMMUNICATION UNION, 2005a).

Na proposta apresentada neste trabalho o E-model é utilizado no diagnostico da qualidade das ligações em um determinado contexto. Assim, a partir de um limiar pré-definido, onde é estipulado o valor mínimo em que o fator R poderá assumir, define-se o tempo máximo em que é possível de acréscimo à retenção no processo de multiplexação para que possa diminuir a carga na rede e evitar a degradação das ligações abaixo do limiar desejável.

A seguir são apresentados os detalhes dos procedimentos para estimar o tempo de retenção da multiplexação baseado no E-model.

4.4.1 Cálculo do tempo de retenção baseado no E-model

Cada parâmetro da Equação 4.1 apresenta maneiras diferentes de definição, onde alguns são estaticamente determinados e outros são dependentes de diversos fatores e precisam ser estipulados periodicamente.

Os valores do fator Ie, em situações sem perdas de pacotes, são definidos em (INTERNATIONAL TELECOMMUNICATION UNION, 2001, 2002a). Já em cenários com descarte de pacotes estimasse o impacto por meio da Equação 2.12.

O cálculo do fator Id foi resumido em (COLE; ROSENBLUTH, 2001) para a seguinte equação:

$$Id = \begin{cases} 0,024.TA & ,TA \leq 177,3ms \\ \\ 0,024.TA + 0,11.(TA - 177,3) & ,TA > 177,3ms \end{cases} \tag{4.2}$$

sendo TA referente ao tempo de atraso total de fala e escuta.

Pode-se observar na Equação 4.3 uma representação dos componentes que compõem o atraso total de fala e escuta em um ambiente com multiplexação.

$$TA = T_{codec} + T_{mux} + T_{rede} + T_{de-jitter} \tag{4.3}$$

onde T_{codec} representa o tempo gasto com a codificação, T_{mux} o tempo de agrupamento no processo de multiplexação, T_{rede} o tempo de propagação na rede e $T_{de-jitter}$ o tempo de *playout* do pacote no *buffer* de amortização de *jitter*.

Com a Formulação 4.3, pode-se isolar o fator T_{mux}. Assim, uma vez determinado o limiar mínimo de qualidade desejável à ligação (R_{min}) e obtido por meio do componente de monitoramento de QoS os valores dos fatores degradantes dos equipamentos e da rede, torna-se possível a definição do tempo máximo de acréscimo do tempo de retenção sem degradar a qualidade das conversações com a Equação 4.4.

$$T_{mux} \leq \frac{-R_{min} + 93,36 - Ie - 0,134.(T_{codec} + T_{rede} + T_{de-jitter}) + 19,503}{0,134} \quad (4.4)$$

O cálculo do T_{mux} pode ser realizado sempre que um novo pacote RTCP ou RTCP XP com informações de *feedback* do receptor for recebido pelo componente de monitoramento de QoS. Assim, o processo de multiplexação poderá se adequar para cada ligação dependendo das características encontradas.

A seguir são apresentados os detalhes do agrupamento dos pacotes, o qual é realizado baseado no cálculo do fator T_{mux} e realiza o agrupamento de modo a obter o melhor ganho e a ser justo para todas as ligações envolvidas.

4.5 Agrupamento dos Pacotes

A proposta adaptativa, além de considerar os diversos fatores degradantes às conversações, apresenta mecanismos para se adequar a todas as ligações envolvidas na multiplexação visando que nenhum dos pacotes fique retido um tempo maior que o definido na Equação 4.4.

No processo de agrupamento o tempo de retenção é alterado constantemente para que o menor T_{mux} das ligações ativas naquele determinado instante não seja extrapolado. Para isso, é verificado a cada novo pacote recebido qual é o tempo máximo de retenção estipulado para sua ligação. Caso o tempo for menor que o definido para o agrupamento dos pacotes naquele instante, ou seja, uma nova ligação com contexto negativo entrou no sistema ou o contexto de alguma ligação já estabelecida foi deteriorado, o tempo de parada-e-espera do processo de multiplexação é diminuído imediatamente para se adequar ao máximo dessa ligação. Desse modo, um pacote nunca permanecerá retido por tempo maior que o definido no seu T_{mux}.

Adicionalmente, é importante notar que a adaptação do tempo total de retenção entre uma multiplexação e outra pode acrescentar uma variação no atraso fim-a-fim das ligações. Assim, são necessários procedimentos adicionais para evitar que o *jitter* inserido não seja maior que o suportado pelas ligações envolvidas. Tal processo é realizado conforme o pseudo algoritmo ilustrado na Figura 4.4.

Nesse algoritmo, primeiramente, é verificado se o pacote inicia ou encontra-se dentro de um *talkspurt* . Caso o pacote inicie um novo *talkspurt* , não existirá variação na latência uma vez que ele foi precedido por um momento de silêncio. Já no caso do pacote pertencer ao fluxo dentro do *talkspurt*, a variação na latência será caracterizada como *jitter*. Assim, nesse caso o processo de multiplexação deve analisar se poderá expandir o T_{mux} em relação ao T_{mux} da multiplexação que agrupou o pacote do início do *talkspurt*. Essa verificação é realizada observando o *jitter* presente no contexto atual da ligação e o tamanho do *buffer* de de-*jitter*. A diferença entre esses valores indica o máximo que o

```
Para cada nova multiplexação (i), faça:
  A cada novo pacote de uma ligação (k), verifique:
    Se pacote NÃO é novo talkspurt, faça:
       Timer_CU(i,k) = Timer_CU(i-1,k) + (Buffer(k) - jitter(k))
onde:
     Timer_CU (i-1,k) = Tempo de retenção anterior
```

Figura 4.4: Pseudo algoritmo para diminuir o *jitter* com a multiplexação adaptativa.

T_{mux} daquela ligação poderá ser expandido em relação ao T_{mux} inicial. As informações necessárias para essa análise podem ser obtidas por meio dos *feedback* fornecidos pelos protocolos RTCP e RTCP XR, onde são descritos parâmetros como *Round-Trip Time* (RTT) e o tamanho atual do *buffer* de de-*jitter*.

Outra verificação realizada para continuação do agrupamento é a conformidade do tamanho do pacote multiplexado com o MTU (*Maximum Transmission Unit*) definido para o canal. A Figura 4.5 apresenta um fluxograma que ilustra o procedimento total de retenção dos pacotes na multiplexação adaptativa.

Ao fim do processo o pacote multiplexado é repassado ao IPsec para aplicação dos métodos de segurança.

Nota-se que é necessário um modo em que o processo de multiplexação possa associar cada pacote com o T_{mux} da sua referente ligação. Um exemplo dessa associação consiste em manter uma tabela onde seriam relacionados alguns campos dos cabeçalhos UDP e IP com o T_{mux} de uma determinada chamada. Outros modos de associação poderiam ser aplicados, tal como a utilização do identificador de contexto do protocolo de compressão de cabeçalhos, entre outros.

4.5.1 Exemplos de Multiplexação

A seguir serão apresentados alguns exemplos de multiplexação para um melhor entendimento.

Nas Figuras 4.6 e 4.7 são ilustrados exemplos de duas seqüências de iterações do processo adaptativo de multiplexação. Na primeira iteração (Figura 4.6) todos os pacotes agrupados correspondem ao início de *talkspurt*, assim não são feitas adaptações para evitar a expansão de *jitter*. Já na segunda interação (Figura 4.7), a qual corresponde à multiplexação seguinte ao primeiro exemplo, é necessária a verificação de *jitter* já que os pacotes se encontram dentro dos *talkspurt*.

No primeiro exemplo (Figura 4.6), inicialmente (passo 1), é recebido um pacote da ligação "A"(*Lig. A*) em que havia sido determinado o tempo de multiplexação ($T_{mux}(A)$) igual à 30 ms. Nesse instante o processo de multiplexação define como tempo de parada e espera ($T_{mux}(ini)$) igual ao valor do T_{mux} desse pacote. No segundo momento (passo 2), após 5 ms decorridos (T_{pass}), um novo pacote é recebido, agora da ligação "B"(*Lig. B*), em que por sua vez tem o T_{mux} definido em 20 ms. Assim, o tempo restante (T_{rest}) do processo de multiplexação é readequado e diminuído de 25 ms para 20 ms. Um terceiro pacote da ligação "C"(*Lig. C*) é recebido 5 ms depois (passo 3), ou seja, 10 ms do início da iteração. Na sua ligação o tempo de multiplexação ($T_{mux}(C)$) era determinado em 30 ms, porém como esse valor ultrapassa os limiares das demais ligações dos pacotes já retidos, o tempo de retenção não é alterado. Assim, ao final do processo (passo 4) o tempo

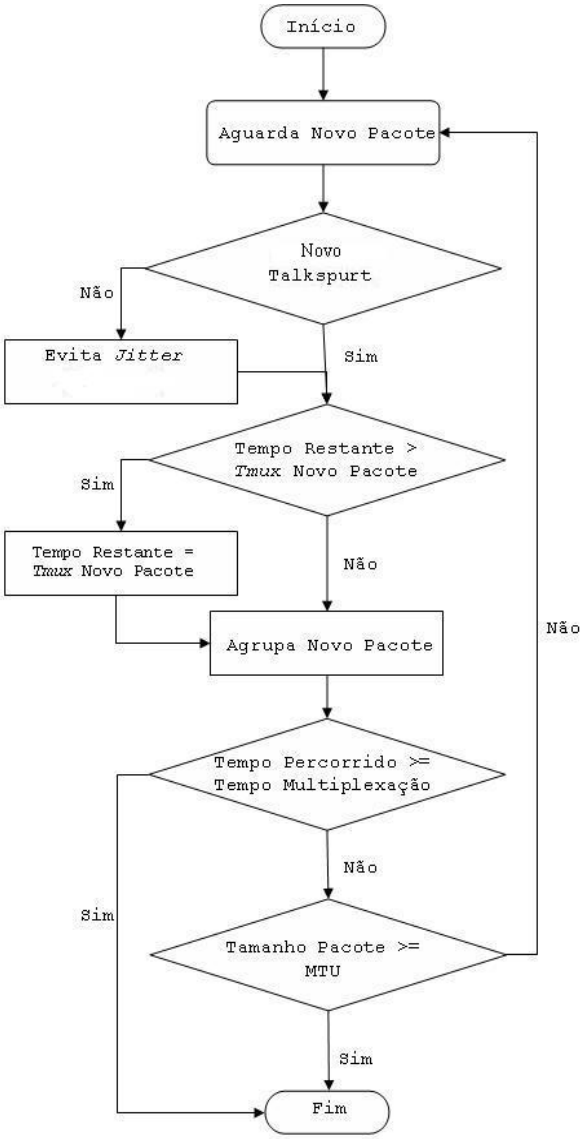

Figura 4.5: Processo de agrupamento dos pacotes.

total de multiplexação ($T_{mux}(fim)$) foi de 25 ms, sendo 20 ms para o segundo pacote e 15 ms para o último. Desse modo, nenhum dos pacotes ultrapassou o limiar estipulado para sua ligação.

Algumas ligações tiveram como tempo de agrupamentos dos pacotes do início do *talkspurt* um tempo menor ao estipulado como máximo para o agrupamento (T_{mux}). Além disso, existe a possibilidade do contexto de alguma ligação melhorar. Assim, nas próximas iterações de multiplexação serão necessárias verificações para averiguar o impacto no *jitter* das ligações envolvidas com uma possível alteração no tempo de retenção. Esse comportamento pode ser observado no próximo exemplo (Figura 4.7).

Em princípio (passo 5), um novo pacote da ligação "A"é obtido. Como esse pacote permaneceu 25 ms na multiplexação anterior e seu máximo era definido em 30 ms é verificado se é possível realizar a expansão do tempo em 5 ms ($Jexp(A)$). Naquele determinado momento a ligação "A"experimenta um *jitter* de 10 ms na rede ($Jrede(A)$). Assim o acréscimo de 5 ms ao tempo de agrupamento resultaria em, aproximadamente, 15 ms de *jitter* na rede, o qual por sua vez é absolvido pelo *buffer* do receptor que se

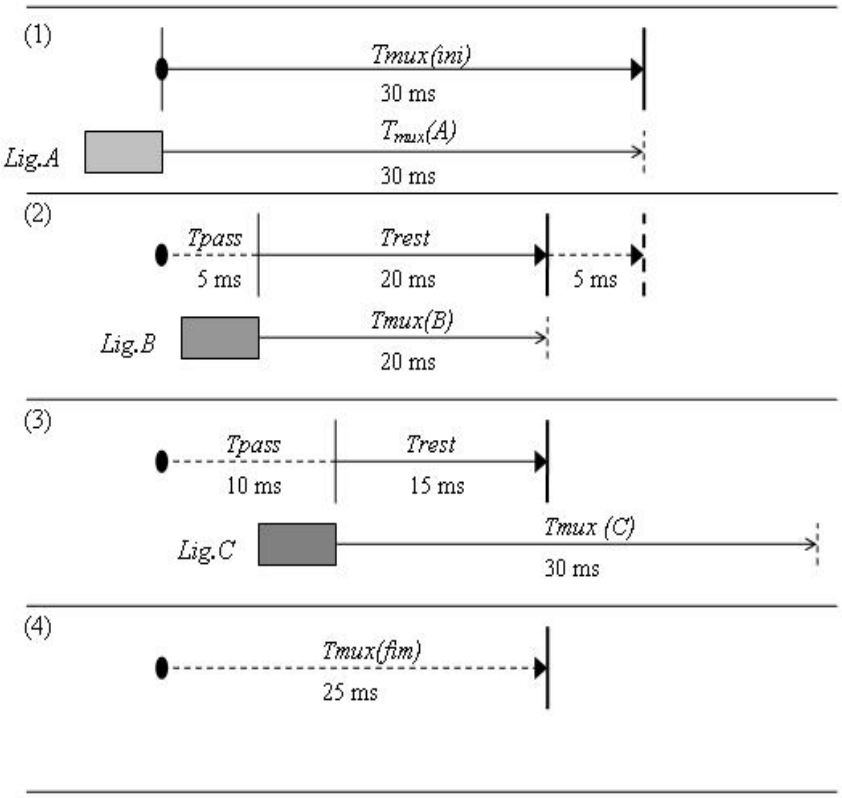

Figura 4.6: Exemplo de Agrupamento 1: Multiplexação no início do *talkspurt*.

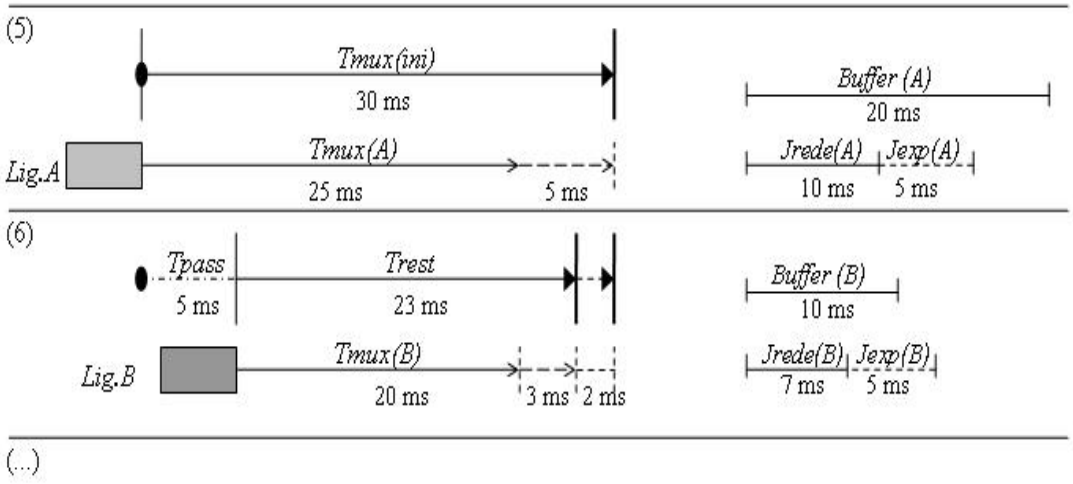

Figura 4.7: Exemplo de Agrupamento 2: Multiplexação dentro de um *talkspurt*.

encontra definido em 10 ms ($Buffer(A)$) conforme a última informação de *feedback* recebida. Desse modo, o tempo máximo ($T_{mux}(ini)$) da ligação é estipulado novamente em 30 ms.

Em seguida (passo 6), ao receber um novo pacote da ligação "B", o processo identifica que essa ligação obteve uma melhora no seu contexto, onde o tempo máximo de agrupamento ($T_{mux}(B)$) fora alterado de 20 ms para 25 ms. Tal como feito para a ligação anterior, realiza-se a verificação do impacto no *jitter*. Nessa ligação é experimentado

um *jitter* de 7 ms na rede ($Jrede(B)$), sendo o *buffer* definido em 10 ms ($Buffer(B)$). Isso demonstra que a expansão ($Jexp(B)$) de 5 ms do tempo de retenção não pode ser realizada por completo. Desse modo, apenas 3 ms são acrescido ao tempo de retenção anterior, definindo-o em 23 ms.

Nos próximos pacotes serão realizados os mesmos procedimentos.

4.6 Sinalização

O estabelecimento da multiplexação entre os *gateways* pode ser realizado aproveitando-se da sinalização inicial do IPSec. Assim, a multiplexação poderia ser inserida aos métodos aplicados ao IPSec, tais como IPComp, AH e ESP.

4.7 Considerações Finais

No presente capítulo foram mostrados os detalhes da solução de melhoria proposta como evolução ao modo tradicional da multiplexação aplicada em VoIPSec. Desse modo, no próximo capítulo serão apresentados os ambientes construídos para validação dessa proposta.

5 SIMULAÇÃO

5.1 Considerações Iniciais

Objetivando validar a solução proposta elaborou-se um ambiente de testes, no qual foram simuladas multiplexações de ligações VoIPSec com características distintas em cenários favoráveis e desfavoráveis.

Esse esquema era formado por quatro componentes (Figura 5.1): um responsável pela simulação do tráfego VoIP (Simulador de Ligações); outro que implementava a solução proposta e abordagens alternativas (Multiplexação); um terceiro que realizava a inserção no ambiente de fatores degradantes (atrasos, *jitter* e perda de pacotes) (Inserção de Degradação); e um último que simulava o comportamento de receptores das sessões VoIP (Receptor).

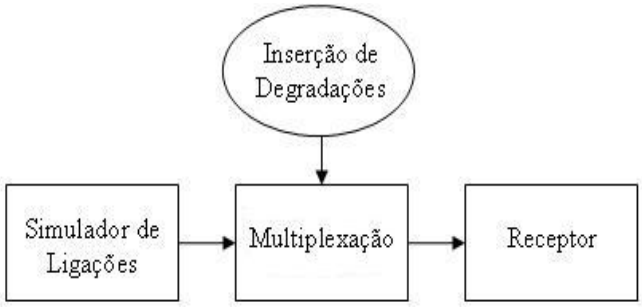

Figura 5.1: Modelo de simulação.

As simulações eram realizadas a partir de *traces* gerados pelo Simulador de Ligações, os quais emulavam o tráfego de diferentes tipos de ligações. Para cada ligação foram definidas diferentes características, tais como *codecs*, propagação de rede, *jitter* e perda de pacotes, conforme os contextos avaliados. Os arquivos gerados eram usados como entrada pelo módulo de Multiplexação, o qual simulava as mesmas dependendo do modelo implementado. Como saída desse sistema eram apontadas, para cada multiplexação simulada, a quantidade de pacotes agrupados, o tempo levado para a construção do pacote multiplexado e o contexto naquele determinado momento (atraso, *buffer* e Ie). Com essas informações era possível avaliar a performance das diferentes abordagens em relação ao ganho com a multiplexação e o nível de qualidade.

Nas próximas seções serão apresentados os detalhes dos componentes das simulações.

5.2 Simulador de Ligações VoIP

Para representação dos tráfegos de áudio foram gerados *traces*, simulando o comportamento de fluxos VoIP, onde cada entrada no arquivo representava o tempo de captura do pacote pelo *gateway* responsável pela multiplexação e aplicação do IPSec. Gerou-se um arquivo para cada ligação, sendo esses utilizados como entrada para simulação do modelo proposto.

Os tráfegos simulados representavam o comportamento de ligações com fluxos não contínuos, ou seja, foi assumido que nos *codecs* das ligações simuladas eram utilizados métodos de detecção de voz e supressão de silêncio. Assim, foram inseridas linhas nos *traces* somente nos momentos em que havia fala.

O comportamento das conversações foi gerado utilizando-se do modelo de geração artificial de conversações proposto em (INTERNATIONAL TELECOMMUNICATION UNION, 1993). Nesse modelo é apresentado um método para reprodução artificial do comportamento temporal de conversações humanas, representando a duração de momentos de fala (unidirecional e bidirecional) e de silêncio mútuo.

No método apresentado em (INTERNATIONAL TELECOMMUNICATION UNION, 1993) as conversações entre duas partes (A e B) são simuladas por meio de transações entre quatro estados:

- Fala individual (A fala e B fica em silêncio);
- Silêncio Mútuo;
- Fala concomitante;
- Fala individual (A fica em silêncio e B fala).

A Figura 5.2 ilustra as direções e probabilidades (em porcentagens) das transições entre os estados.

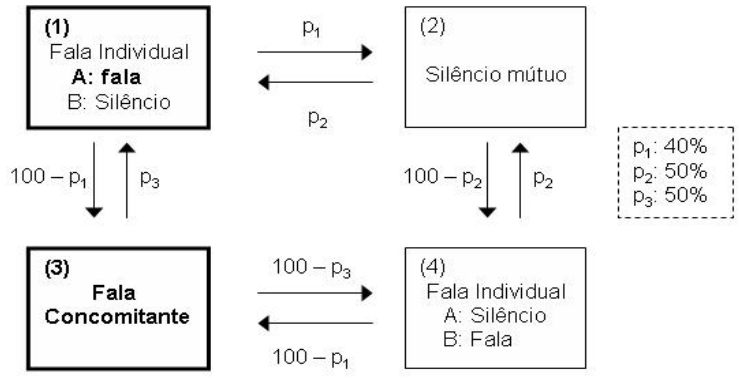

Figura 5.2: Modelo de transição de estados para simulação de conversações. Adaptado de (INTERNATIONAL TELECOMMUNICATION UNION, 1993).

No primeiro estado apenas o lado A encontra-se falando (*Talk*), enquanto o lado B está em silêncio (*Silence*). O oposto desse comportamento é mapeado no estado 4. Já nos estados 2 e 3, os dois lados encontram-se em silêncio (*Mutual silence*) e falando (*Double talk*) ao mesmo tempo, respectivamente. As probabilidades de transição apresentam os seguintes valores: $p_1 = 40\%$, $p_2 = 50\%$ e $p_3 = 50\%$.

A permanência em cada um dos estados varia com as seguintes equações:

- Duração de fala individual (Estados 1 e 4) = $-0.854ln(1-x1)$;
- Duração de falas concomitantes (Estado 2) = $-0.226ln(1-x2)$;
- Duração de silêncio mútuo (Estado 3) = $-0.456ln(1-x3)$.

sendo $0 < x1, x2, x3 < 1$: variáveis randômicas com distribuição uniforme.

Baseados nesse método foram gerados *traces* com o tempo de 50 segundos para cada ligação, onde eram inseridas entradas nos arquivos no momento em que a execução do modelo encontrava-se no primeiro ou terceiro estado, ou seja, nos instantes em que a parte A (que representava nesse trabalho o emissor) apresentava-se falando. A quantidade de pacotes (linhas) inseridos nos momentos de fala variava com a taxa de geração do *codec* simulado.

5.3 Ambiente e Fatores Degradantes

Diferentes aspectos foram definidos nos experimentos com o objetivo de representar contextos favoráveis e desfavoráveis. Os cenários se caracterizavam em relação a fatores que influenciavam diretamente na estimativa da qualidade das ligações ou na execução dos métodos de multiplexação. Desse modo, variou-se para cada ligação aspectos como:

- Atraso fim-a-fim;
- *Jitter*;
- *Codec*;
- Perda de pacotes.

Cada fator foi definido com o objetivo de representar o comportamento de ambientes reais.

Atraso e *jitter* seguiram os valores determinados nos cenários hipotéticos definidos em (INTERNATIONAL TELECOMMUNICATION UNION, 2006). Esses cenários representavam caminhos semelhantes a ligações dentro e entre continentes.

Para os parâmetros relacionados com os *codecs* (tempo de codificação e fator Ie) foram seguidos os valores de diferentes famílias de *codecs* determinados em (INTERNATIONAL TELECOMMUNICATION UNION, 2002b)

A degradação com perda de pacotes foi simulada no canal multiplexado e sua implementação é descrita na Seção 5.3.1.

Os valores específicos de cada fator nas ligações nos contextos simulados são descritos no Capítulo 6.

Outros fatores, como eco, não foram considerados nesse trabalho, considerando que fora aplicados mecanismos de cancelamento. Assim, assumiu-se os valores padrão definidos em (INTERNATIONAL TELECOMMUNICATION UNION, 2005a)

5.3.1 Perda de Pacotes

Para a simulação de um canal com perdas de pacotes seguiu-se o modelo Gilbert-Elliote (GILBERT, 1960). Nesse modelo o comportamento de um canal é representado por meio de dois estados: um chamado de "Bom"(*Good*) (G), onde a probabilidade média de erro de bit é bastante pequena ($Pg \approx 0$) e outro referenciado como "Ruim"(*Bad*) (B),

que apresenta uma probabilidade alta de erro ($Pb = 0,5$). As transições entre os estados são definidas pelas probabilidades P e Q, sendo P a probabilidade de mudança do estado "Bom"para o "Ruim"e Q do "Ruim"para o "Bom". Adicionalmente, $(1 - P)$ e $(1 - Q)$ são as probabilidades de não ocorrerem transições (Figura 5.3).

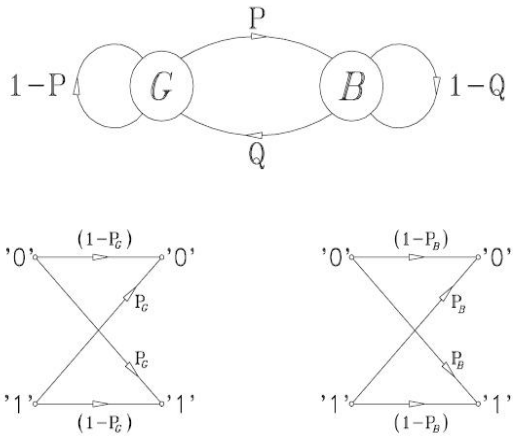

Figura 5.3: Modelo de canal de *Gilbert Elliote* (INTERNATIONAL TELECOMMUNICATION UNION, 2005b).

Em (INTERNATIONAL TELECOMMUNICATION UNION, 2005b) assume-se $Pg = 0$, ou seja, quando o modelo encontra-se no estado "Bom"nenhum bit é alterado, e $Pb = 0,5$. Assim, resume-se a definição das probabilidades de transição entre os estados com a seguinte formulação:

$$P = 2.(1 - \gamma).BER \tag{5.1}$$

$$Q = (1 - \gamma).(1 - 2.BER) \tag{5.2}$$

sendo BER (*Bit Error Rate*) o acronimo em inglês para Taxa de Erro de Bits. O Termo γ equivale-se à:

$$\gamma = 1 - (P + Q) \tag{5.3}$$

O fator γ está relacionado com o comportamento do erro, onde $\gamma \approx 0$ caracteriza erros aleatórios, ou seja, um primeiro erro não tem relação com os posteriores, já para erros associados ($\gamma \approx 0.5$) esse termo representa o comportamento de erros em rajadas (*burst*).

Nesse trabalho assumiu-se que as perdas ocorriam em *burst*. Foi utilizada para implementação do modelo *Gilbert Elliote* o código de livre acesso, disponível em (INTERNATIONAL TELECOMMUNICATION UNION, 2005b).

5.4 Multiplexação

Os *traces* gerados pelo Simulador de Ligações eram utilizados como entrada pelo módulo de Multiplexação. Nesse componente além da proposta desse trabalho também foram implementadas, com intuito comparativo, diferentes abordagens de multiplexação.

O modelo proposto foi confrontado com abordagens onde o tempo de retenção para multiplexação é estipulado de modo fixo e sem considerar parâmetros de qualidade. Os

tempos avaliados foram os valores iguais ao tamanho (em ms) dos *frames* nos principais *codecs*: 10 ms, 20 ms e 30 ms.

Estipulou-se o valor mínimo desejável para R em 70, o qual garante que a maioria dos participantes considere satisfatória a qualidade das ligações conforme recomendado em (INTERNATIONAL TELECOMMUNICATION UNION, 2003b).

5.5 Receptor

No lado do receptor o fator relevante relacionado com a qualidade das sessões VoIP consiste no comportamento do *buffer* de amortização de *jitter*. Um *buffer* pequeno provoca o descarte dos pacotes que chegarem muito tarde para sua reprodução, por sua vez, um *buffer* muito grande acarreta no aumento do tempo total de fala e escuta (TELECOMMUNICATIONS INDUSTRY ASSOCIATION, 2006). Isso demonstra a relevância desse fator na qualidade final das ligações.

Desse modo, foi introduzido ao ambiente um módulo para simulação do comportamento de um *buffer* de compensação de *jitter*.

Pode-se utilizar *buffers* fixos, ou seja, que mantêm o seu tamanho constante por toda sessão, ou dinâmico, que altera a sua capacidade de retenção conforme a variação do atraso presente na rede.

Nesse trabalho foi simulado um *buffer* dinâmico, pois o mesmo, devido as suas características adaptativas, é o mais recomendado por diversos trabalhos (CISCO, 2007b). Além disso, é importante observar se o modelo de multiplexação adaptativa provoca a expansão demasiada de um *buffer* dinâmico ao inserir uma variação no atraso com a adaptação do tempo de retenção.

Foi sumulado o comportamento da abordagem apresenta em (RAMJEE et al., 1994), a qual realiza o ajuste de tamanho do *buffer* nos momentos de silêncio das conversações. O tempo de *playout*, ou seja, o tempo em que um pacote leva entre a sua emissão até a execução no receptor, é determinado conforme a seguinte equação:

$$pi = \hat{d}_i + 4 * \hat{v}_i \tag{5.4}$$

onde $\hat{d}_i$ corresponde a estimativa do atraso total na rede e $\hat{v}_i$ a média da variação do atraso dentro de um *talkspurt*. Os valores desses termos são estimados conforme o algoritmo de retransmissão do TCP: RTO (*Retransmission TimeOut*) (RAMJEE et al., 1994).

5.6 Considerações Finais

A partir dos ambientes de avaliação especificados, no próximo capítulo serão mostrados os resultados obtidos e será discutido cada aspecto observado.

6 RESULTADOS E DISCUSSÃO

6.1 Considerações Iniciais

Nas próximas seções são apresentados os resultados obtidos com as simulações em cenários favoráveis e desfavoráveis.

6.2 Avaliação

A performance do modelo proposto foi avaliada em relação ao nível de satisfação obtido e à taxa de redução do *overhead* na rede.

A satisfação dos usuários foi calculada por meio do Modelo E conforme a Equação 4.1. Para esse cálculo considerou-se os parâmetros de cada ligação, o tempo de retenção resultante do processo de multiplexação, o atraso fim-a-fim e a taxa de perda de pacotes.

Como já visto, o ganho na redução pode ser calculado por meio da razão entre o número de bytes resultantes da multiplexação sobre o número de bytes dos pacotes não multiplexados. Nesse trabalho, assumiu-se que as ligações eram protegidas com criptografia e autenticação por meio do cabeçalho ESP em modo de túnel. Assim, pacotes não multiplexados seriam formados, além dos cabeçalhos originais (Enlace, IP, UDP e RTP), pelos cabeçalhos adicionais do IPSec: IP, ESP e ESP trailer (KENT; ATKINSON, 1998a). Desse modo, o ganho na rede pode ser obtido conforme a Equação 6.1.

$$1 - \frac{(IP + ESP + ESPtrailer) + \sum_{i=1}^{n}(IP + UDP + RTP + p)}{\sum_{i=1}^{n}(IP + ESP + ESPtrailer + IP + UDP + RTP + p)} \tag{6.1}$$

em que n corresponde ao total de pacotes emitidos pelas ligações durante uma iteração de multiplexação. O *payload* dos pacotes é definido por p.

6.2.1 Análise Estatística

Com o intuito de verificar se os resultados da solução proposta apresentaram diferenças, estatisticamente, significativas em relação aos modelos confrontados, foi aplicado o teste não paramétrico U de *Mann-Whitney*, considerando significativo um p-valor < 0,01.

Em todas as avaliações resultaram-se em diferenças significativas com um p-valor menor de 0,0001.

6.3 Cenários Favoráveis

Em princípio, o comportamento da proposta foi observado em um cenário favorável, ou seja, com valores baixos de atrasos e de *jitter*, sem perdas de pacotes na rede e com *codecs* com pouca deterioração. A Tabela 6.1 apresenta um resumo dos valores definidos para representação desse contexto.

Tabela 6.1: Parâmetros das ligações simuladas em um cenário favorável.

Ligação	**Atraso Rede (ms)**	***Jitter (ms)***	***Codec***	Ie
1	100	5	G.711 (62.5 kb/s)	0
2	100	5	G.711 (62.5 kb/s)	0
3	100	5	G.711 (62.5 kb/s)	0
4	100	5	G.711 (62.5 kb/s)	0
5	100	5	G.711 (62.5 kb/s)	0
6	100	5	G.711 (62.5 kb/s)	0
7	100	5	G.711 (62.5 kb/s)	0
8	100	5	G.711 (62.5 kb/s)	0
9	100	5	G.711 (62.5 kb/s)	0
10	100	5	G.711 (62.5 kb/s)	0

Os valores definidos para atraso e *jitter* correspondem aos encontrados em ligações dentro de um mesmo continente, conforme apresentado no cenário hipotético em (INTERNATIONAL TELECOMMUNICATION UNION, 2006). Exemplo de *codec* sem deterioração consiste na aplicação do G.711.

Nesse cenário foi possível expandir o tempo de multiplexação e com isso aumentar a taxa de compressão, como pode ser observado na Figura 6.1.

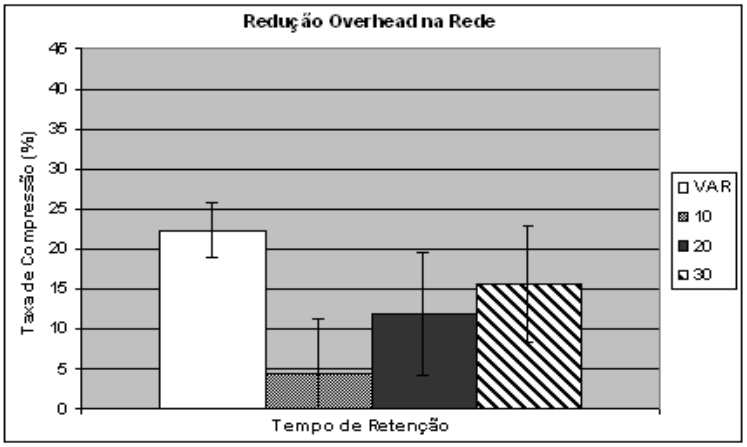

Figura 6.1: Redução do *overhead* com multiplexação fixa e variável em cenário favorável.

Uma vez utilizado somente o *codec* G.711 com 62kb/s em todas as ligações pode-se estimar, por meio da Equação 3.3, que a máxima compressão nesse contexto poderia ser de 25%. O modelo adaptativo ("VAR") alcançou uma compressão em média de, aproximadamente, 22%, sendo maior em relação às demais alternativas. Os valores altos de desvio padrão explica-se devido à característica variável dos tráfegos gerados.

Na Figura 6.2 são apresentados os valores resultantes do nível da qualidade das ligações.

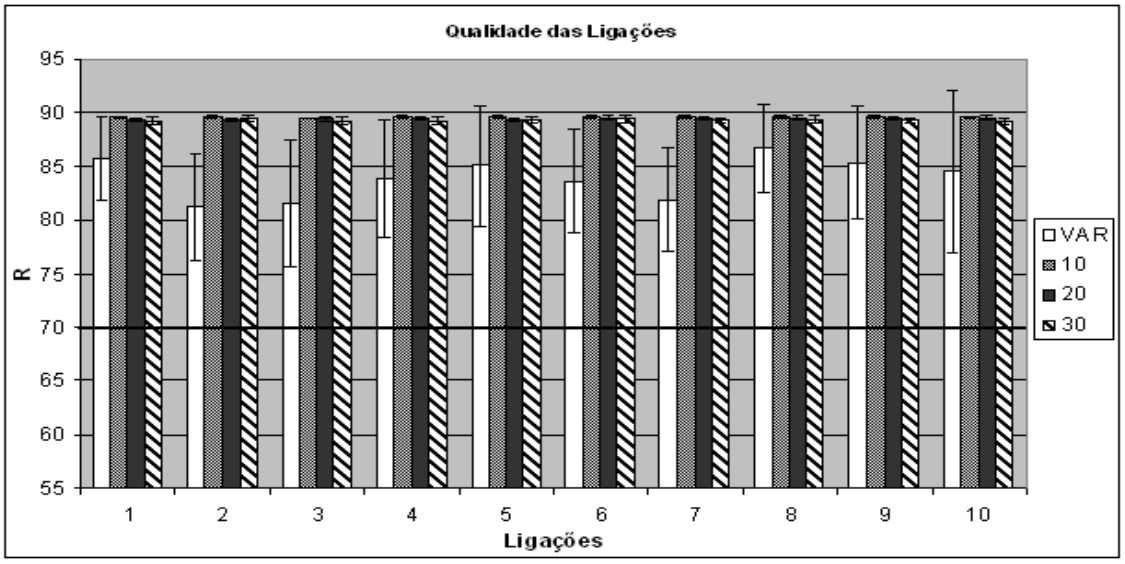

Figura 6.2: Qualidade das conversações com multiplexação fixa e variável em cenário favorável.

Nessas condições, como esperado, todas abordagens apresentam uma alta qualidade das ligações. A proposta adaptativa acarretou um resultado menor, contudo respeitou os limites previamente definidos.

6.4 Cenários Desfavoráveis

A robusteza da proposta foi avaliada em cenários com contextos negativos, onde apresentavam-se degradações com latência, *jitter*, *codecs* e descarte de pacotes. Nas próximas seções são apresentados os detalhes de cada contexto e os resultados obtidos.

6.4.1 Atraso fim-a-fim

Primeiramente, um ambiente com valores altos na propagação dos pacotes e com tempo *jitter* elevado foi simulado para avaliação do desempenho das diferentes abordagens de multiplexação com um valor alto de atraso fim-a-fim.

Ligações entre continentes são exemplos de situações onde os pacotes VoIP poderão experimentar propagações na rede com tempo acima dos recomendados em (INTERNATIONAL TELECOMMUNICATION UNION, 2002b). Conforme (INTERNATIONAL TELECOMMUNICATION UNION, 2006) ligações nesses cenários podem experimentar atrasos na rede de até 233 ms.

Assim, nesse experimento, as ligações foram simulados com atrasos na rede na ordem de 100 ms a 230 ms.

Outro aspecto ligado diretamente ao atraso total é a variação na rede (*jitter*), cujo valor alto resulta na necessidade de um maior *buffer* para sua amortização. Definiu-se *jitter* entre 5 ms a 20 ms, tais como demonstrado nos piores casos em (INTERNATIONAL TELECOMMUNICATION UNION, 2006)

Somando-se ainda o atraso com a codificação (entre 0.125m a 67 ms (INTERNATIONAL TELECOMMUNICATION UNION, 2002b)), os piores casos poderiam chegar, nesse experimento, a valores acima de 300 ms. A Tabela 6.2 apresenta um resumo dos valores estipulados.

Tabela 6.2: Parâmetros das ligações simuladas em um cenário com degradações com atraso fim-a-fim.

Ligação	Atraso Rede (ms)	*Jitter (ms)*	*Codec*	Ie
1	100	5	GSM EFR (12.2 kb/s)	5
2	140	10	GSM EFR (12.2 kb/s)	5
3	180	10	GSM EFR (12.2 kb/s)	5
4	220	15	GSM EFR (12.2 kb/s)	5
5	150	10	GSM EFR (12.2 kb/s)	5
6	110	15	GSM EFR (12.2 kb/s)	5
7	160	10	GSM EFR (12.2 kb/s)	5
8	230	10	GSM EFR (12.2 kb/s)	5
9	190	20	GSM EFR (12.2 kb/s)	5
10	170	10	GSM EFR (12.2 kb/s)	5

Escolheu-se para essa avaliação o *codec* GSM por apresentar um tempo considerável de codificação, mas ao mesmo tempo manter um impacto baixo em relação ao termo Ie .

Nas Figuras 6.3 e 6.4 temos os resultados obtidos nesse ambiente.

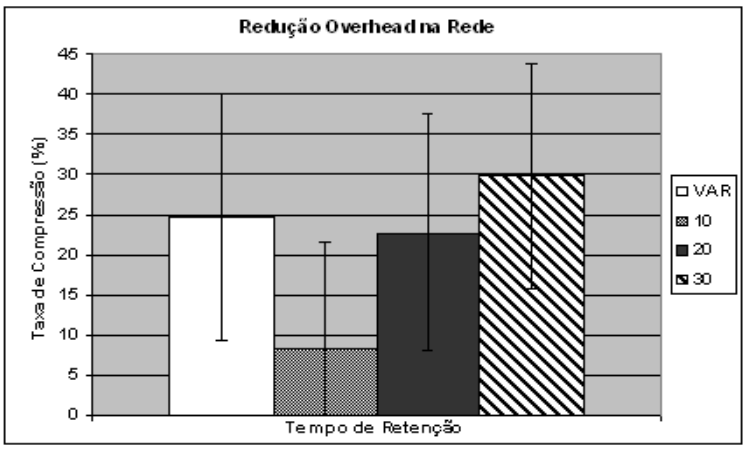

Figura 6.3: Redução do *overhead* com multilplexação fixa e variável em cenário desfavorável com atraso fim-a-fim.

Tal como no exemplo anterior, pode-se utilizar a Equação 3.3 para estipular a redução máxima. Nesse teste o ganho poderia alcançar 50%. Observar-se que para esse tipo de degradação a solução adaptativa apresentou uma compressão maior em relação às alternativas de 10 ms e 20 ms.

Adicionalmente, a característica de basear-se em parâmetros de qualidade fez com que a proposta adaptativa não deteriorasse a qualidade das ligações que estavam abaixo do limiar previamente estipulado (70), ao contrário do que ocorreu com as demais abordagens. Isso se deve ao fato da solução proposta não reter pacotes de ligações que estão com a qualidade abaixo do nível desejado. Os diferentes níveis entre as ligações envolvidas na multiplexação permitem ao método adaptativo compensar as restrições das ligações com baixa qualidade. Assim, como pôde ser observado, o agrupamento dos pacotes das ligações com qualidade acima do estipulado permitiu a redução do *overhead* na rede.

Nota-se também, que a variação na qualidade das ligações com boa qualidade (acima de 70) não foi degradada para valores abaixo do limiar desejado.

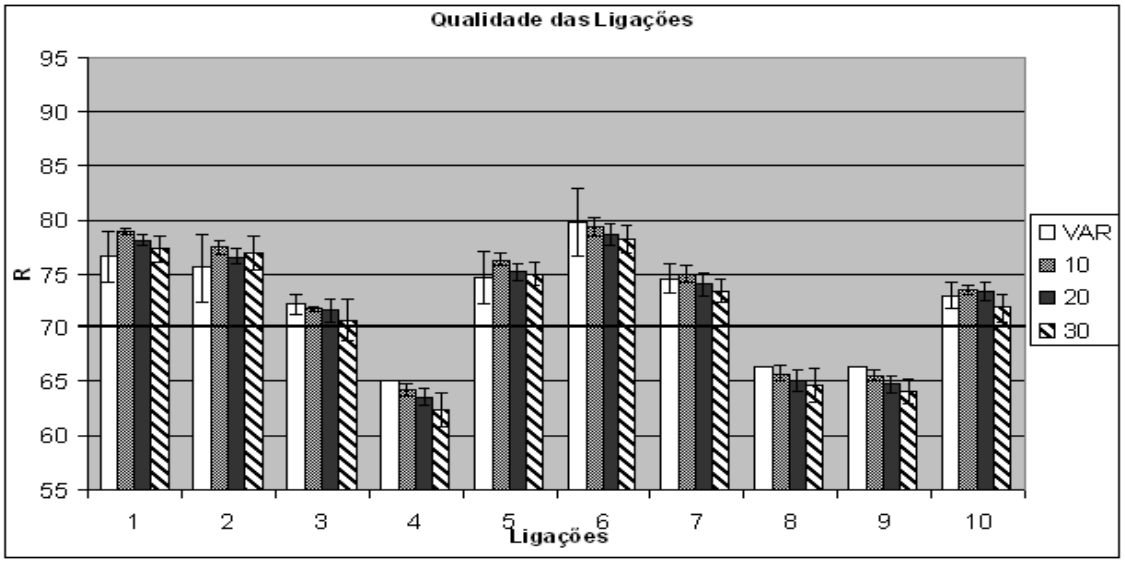

Figura 6.4: Qualidade das conversações com multiplexação fixa e variável em cenário com atraso fim-a-fim.

6.4.2 Codecs

Nos experimentos já descritos, não foram inseridas degradações relevantes decorrentes da codificação (fator Ie). Como poder ser visto na Tabela 2.7 o fator Ie , que representa o impacto com o equipamento, pode alcançar valores acima de 20, o que restringiria o atraso máximo em 100 ms para se manter níveis onde a maioria dos participantes considaressem a qualidade muito satisfatória.

Desse modo, avaliou-se diferentes tipos de *codecs* os quais apresentavam diversos valores de Ie , tais como descritos em (INTERNATIONAL TELECOMMUNICATION UNION, 2002b). Em todas as ligações foram estipulados atrasos de 150 ms e *jitter* de 10 ms (Tabela 6.3).

Tabela 6.3: Parâmetros das ligações simuladas em um cenário com *codecs* degradantes.

Ligação	**Atraso Rede (ms)**	***Jitter (ms)***	***Codec***	Ie
1	100	5	G.726 (24kb/s)	25
2	100	5	G.728 (12.8 kb/s)	20
3	100	5	G.732.1 (5.3 kb/s)	19
4	100	5	G.732.1 (6.3 kb/s)	15
5	100	5	G.729-A (8 kb/s)	11
6	100	5	G.729 (8 kb/s)	10
7	100	5	G.728 (16 kb/s)	7
8	100	5	G.727 (32 kb/s)	7
9	100	5	GSM EFR (12.2 kb/s)	5
10	100	5	G.726 (64 kb/s)	2

Com os diferentes *codecs* utilizados a compressão nesse contexto poderia variar entre 35% até 55%. A maior compressão alcançada foi de 27% com a abordagem de 30 ms (Figura 6.5) . Dessa vez a abordagem adaptativa obteve uma compressão melhor somente que a abordagem de 10 ms. Porém, novamente a técnica proposta apresentou um melhor

rendimento em relação à qualidade das ligações com baixo nível e respeitou o limiar estipulado nas demais ligações (Figura 6.6).

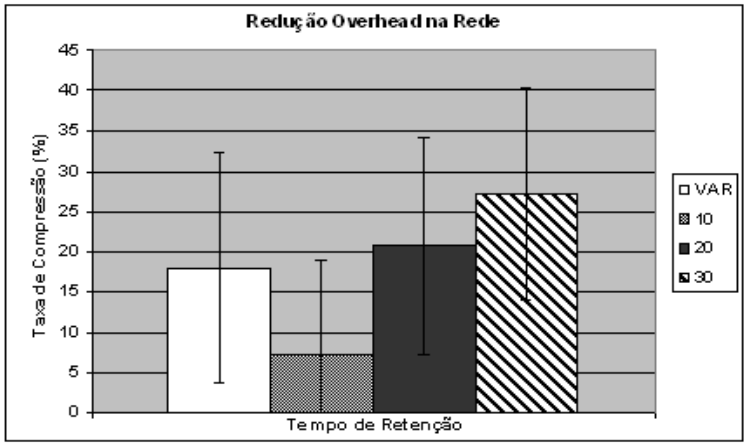

Figura 6.5: Redução do *overhead* com multiplexação fixa e variável em cenário desfavorável com *codecs* degradantes.

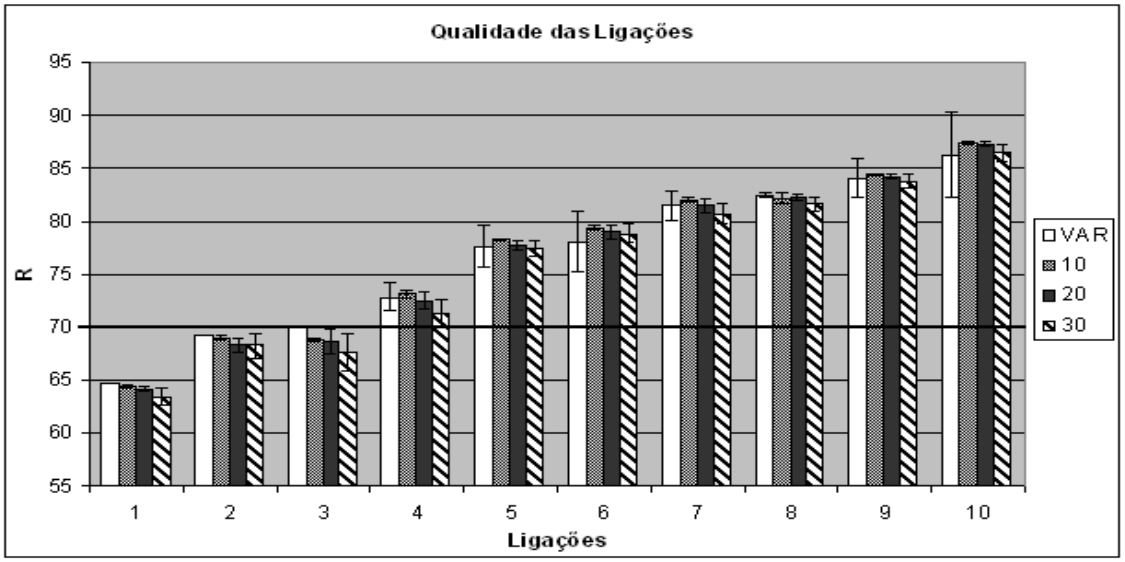

Figura 6.6: Qualidade das conversações com multiplexação fixa e variável em cenário com *codecs* degradantes.

6.4.3 Perda de Pacotes

No experimento anterior foi avaliado o desempenho das abordagens com diferentes *codecs*. Contudo, é importante também observar o impacto com descartes de pacotes na rede. Cada *codec* apresenta uma robustez diferente em relação à perda de pacotes. Desse modo, no próximo experimento foram simuladas ligações em um canal com perda de pacotes com diferentes *codecs*.

As configurações equivalem ao experimento anterior (Tabela 6.3), contudo foi inserido ao canal uma probabilidade de descartes de pacotes equivalente a 10^{-4}. As Figuras 6.7 e 6.8 demonstram os resultados nesse contexto. Observa-se que ocorreu uma diminuição

na compressão da nossa proposta (VAR), contudo ainda apresentou um melhor resultado em relação aos valor de 10 ms.

Houve uma grande deterioração para todas as abordagens, contudo como a proposta apresentada nesse trabalho considera a perda de pacotes, essa solução apresentou resultados melhores nas ligações mais críticas em relação às abordagens com um valores fixos de retenção.

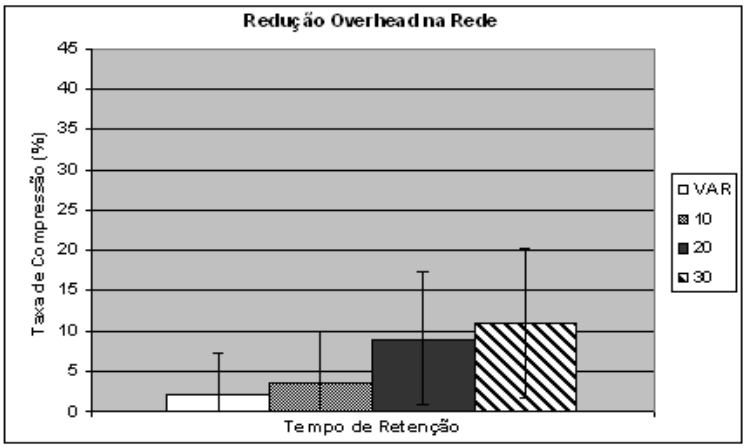

Figura 6.7: Redução do *overhead* com multiplexação fixa e variável em cenário desfavorável com perda de pacotes.

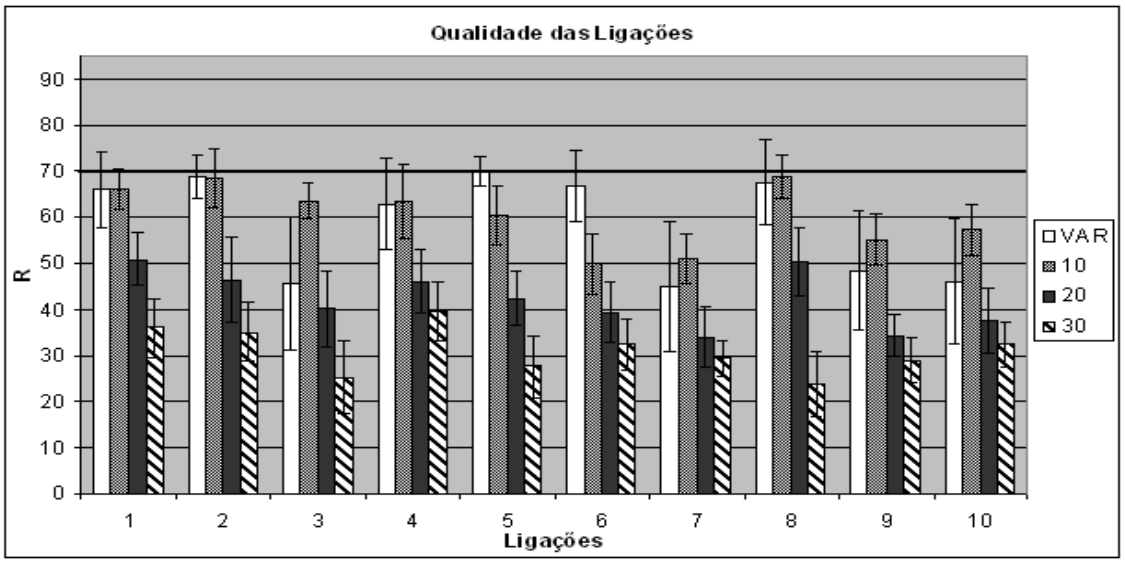

Figura 6.8: Qualidade das conversações com multiplexação fixa e variável em cenário com perda de pacotes.

6.4.4 Atraso + *Codec* + Perda de Pacotes

A relação dos experimentos anteriores permitiu observar, individualmente, o impacto de cada fator degradante. Entretanto, em cenários reais, comumente, encontram-se todos os fatores ao mesmo tempo. Deste modo, uma última avaliação simulou um ambiente onde eram distribuídos entre as ligações diversos fatores de degradação (Tabela 6.4). Nessa simulação as perdas de pacotes tinham uma probabilidade de 10^{-5}.

Tabela 6.4: Parâmetros das ligações simuladas em um cenário com degradações com atraso fim-a-fim, *codecs* e perda de pacotes.

Ligação	**Atraso Rede (ms)**	***Jitter (ms)***	***Codec***	Ie
1	100	5	G.726 (24kb/s)	25
2	140	10	G.728 (12.8 kb/s)	20
3	180	10	G.732.1 (5.3 kb/s)	19
4	220	10	G.732.1 (6.3 kb/s)	15
5	150	10	G.729-A (8 kb/s)	11
6	110	15	G.729 (8 kb/s)	10
7	160	10	G.728 (16 kb/s)	7
8	200	5	G.727 (32 kb/s)	7
9	150	10	GSM EFR (12.2 kb/s)	5
10	100	15	G.726 (64 kb/s)	2

Apesar do contexto negativo, pode-se observar que a diversidade entre as ligações permitiu que a solução proposta reduzisse o *overhead* na rede (Figura 6.9).

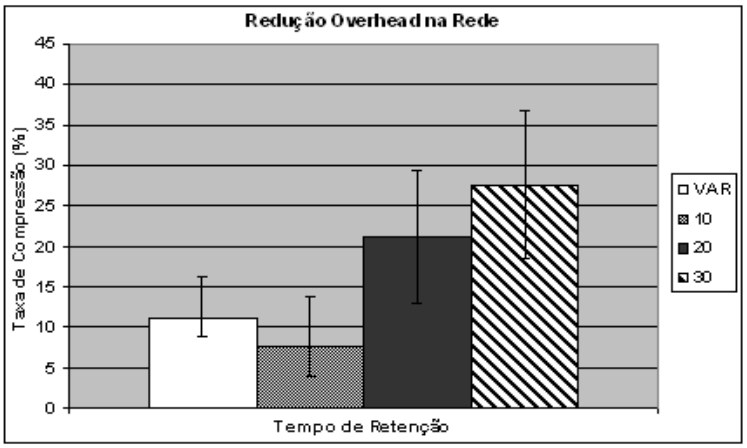

Figura 6.9: Redução do *overhead* com multiplexação fixa e variável em cenário desfavorável com atraso fim-a-fim, *codecs* degradantes e perda de pacotes.

A qualidade na rede apresentou um melhor resultado para as abordagens com a multiplexação com valores baixos (10 ms), sendo que a solução proposta também conservou a sua robustez, mantendo o nível da qualidade das ligações nos patamares desejados (Figura 6.10).

Nota-se que, ao contrário dos primeiros experimentos, a solução adaptativa (VAR) também apresentou variações, em relação à qualidade nas ligações que estavam abaixo do limiar desejado. Isso ocorreu devido à deterioração com as perdas dos pacotes o que diminuiu o nível de qualidade de maneira diferenciada durante os experimentos.

6.5 Implementação e Performance Computacional

Um aspecto relevante nesse tipo de aplicação, onde comumente são utilizados sistemas embarcados, consiste no impacto na carga de processamento. Assim, é importante apresentar alguns pontos em relação a possível carga resultante com o modelo proposto.

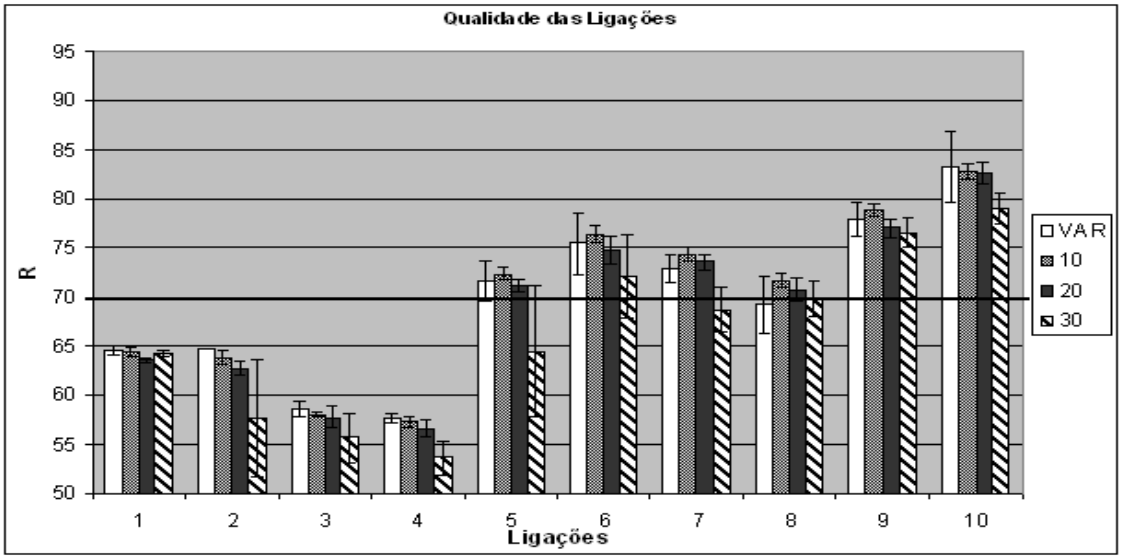

Figura 6.10: Qualidade das conversações com multiplexação fixa e variável em cenário com atraso fim-a-fim, *codecs* degradantes e perda de pacotes.

Como foi apresentado, as alterações propostas são relacionadas em dois aspectos: cálculo do tempo de retenção e agrupamento dinâmico dos pacotes.

6.5.1 Carga de Processamento com Cálculo do tempo de Retenção

Em relação ao cálculo do tempo de retenção, nota-se que esse é baseado nas estatísticas submetidas por meio do protocolo RTCP(XR), sendo sugerido a sua atualização logo quando um novo pacote de estatística for recebido. Assim, observado que na RFC 3530 (SCHULZRINNE et al., 2003) recomenda-se que o fluxo do RTCP não pode ultrapassar 5% do tráfego RTP, é possível notar que a taxa de transmissão desses pacotes fica bem abaixo da capacidade de roteamento dos fluxos de voz. Isso mostra que o cálculo de tempo de retenção será realizado em períodos bem menores que o tratamento dos pacotes RTP.

Considerando que no experimento avaliado foram simuladas 10 ligações simultâneas, as quais enviavam pacotes em média com um taxa de 36 a 50 pacotes/s, o tráfego RTCP, seguindo a recomendação da RFC 3550, não poderia ultrapassar 1,75 a 2,5 pacotes/s.

Adicionalmente, utiliza-se como referência uma otimização do Modelo E para as avaliações, em tempo real, propostas em (COLE; ROSENBLUTH, 2001), as quais alcançam resultados semelhantes às equações definidas em (INTERNATIONAL TELECOMMUNICATION UNION, 2005a), diminuindo o número de cálculos executados.

6.5.2 Carga de Processamento com Agrupamento dos Pacotes

Outro aspecto relevante é o processamento necessário para o agrupamento dinâmico dos pacotes. Para o agrupamento é necessário a verificação permanente dos pacotes RTP. Isso pode ser uma tarefa de impacto negativo na carga de processamento. Alternativas para melhorar o desempenho nesse tipo de abordagem consiste em utilizar implementações que aprimoram a tarefa de *match* dos pacotes. Isso pode ser realizado, por exemplo, utilizando tabelas de acesso direto, como tabelas *hash*.

6.6 Considerações Finais

Nesse capítulo pôde-se observar o comportamento da solução proposta em diversos ambientes. Por fim, no próximo capítulo serão realizadas as considerações finais e propostas algumas oportunidades de pesquisas futuras.

7 CONCLUSÃO

Neste trabalho foi apresentada uma solução onde se buscou atender diferentes requisitos da aplicação de VoIP, combinando utilização efetiva da banda, garantia da qualidade do serviço e segurança nas conversações.

Enfocou-se na melhoria do desempenho do protocolo IPSec na proteção das sessões de VoIP, o qual é amplamente utilizado em VPNs, mas apresenta como desvantagem a expansão dos pacotes protegidos. Nesse contexto, foi identificada a necessidade da evolução da técnica de multiplexação aplicada para redução do *overhead* na rede em aplicações de VoIPSec, onde era necessária a definição de um tempo ideal de multiplexação que possibilitasse a compressão mantendo o nível de qualidade das conversações.

Assim, foi elaborada uma solução, onde se propôs um modelo de multiplexação adaptativo baseado em parâmetros de qualidade. Com isso, buscou-se melhorar o desempenho mantendo os níveis de qualidade.

Para isso, foi utilizada uma recomendação padronizada: o E-model. Esse modelo se apresentou eficaz por possibilitar um diagnóstico objetivo e completo em relação à percepção de qualidade das ligações pelos usuários. Adicionalmente, um outro fator importante em relação ao E-model consiste na contribuição apresentada com o modo de aplicação. Essa recomendação normalmente é implementado nos pontos finais de uma sessão, ou seja, na fonte para adaptação da taxa de transmissão dos *codecs* ou no destino como um modo de diagnóstico para adequação do *buffer* de amortização de *jitter*. Neste trabalho foi demonstrada a aplicabilidade desse método também nos componentes intermediários da rede.

Os resultados demonstraram que a solução conseguiu alcançar o seu objetivo: aumentando a redução do *overhead* na rede em contextos favoráveis, mantendo o nível desejável de qualidade nas ligações nos desfavoráveis. Foram avaliados diferentes aspectos, os quais influenciavam diretamente no diagnóstico de qualidade das ligações. Os contextos negativos representavam ambientes em casos críticos, o que mostrou a robusteza da proposta apresentada.

Notou-se ainda que a performance da solução proposta está relacionada com a multiplicidade dos contextos das ligações, onde ligações de boa qualidade compensam as restrições das ligações com baixa qualidade. Isso demonstra que esta solução mostra-se aplicável em ambientes onde os contextos são comumente alternáveis (variando entre bom e ruim) ou que exista uma multiplicidade entre as ligações. Caso existam restrições permanentes em todas as ligações, sem alterações para momentos bons, a solução proposta não resultará em resultados favoráveis, uma vez que ela não terá a oportunidade de agrupamento e com isso redução do *overhead*.

Em relação aos demais métodos de multiplexação, a proposta apresentou um método que abrange de maneira mais completa e justa as necessidades e restrições. Isso se ca-

racteriza no fato da solução considerar diversos fatores de qualidade e tratar cada ligação individualmente.

A solução enfocou na melhoria do desempenho de VoIP com segurança, contudo ela é facilmente aplicável em contextos sem segurança. Desse modo, a contribuição mostrou-se extensível a outros propósitos.

Pode-se citar como possibilidade de trabalhos futuros aspectos como: métodos para definição da expansão máxima do tempo de retenção em cenários positivos; a determinação de limiares diferentes dependendo da realidade de cada cenário; a avaliação da implementação do proposta; a extensão da solução para outros fins; entre outros.

Em relação a primeira sugestão, uma multiplexação poder alcançar a sua compressão máxima antes do tempo de retenção atingir o seu final. Desse modo, heurísticas poderiam ser desenvolvidas para diagnosticar se o processo de agrupamento alcançou o seu máximo efetivo. Sugestão para essa melhoria consiste na aplicação de métodos que observem a taxa de variação da compressão em tempo real.

Outra evolução possível consiste em determinar valores diferentes para o limiar mínimo, o qual, dependendo da realidade de cada aplicação, pode ser diferente dos recomendados nas bibliografias. Essa determinação poderia ser realizada com sistemas inteligentes, que levariam em conta diversos aspectos, entre ele as sugestões dos usuários, custos, entre outros.

Apresenta-se também como oportunidade interessante a avaliação da implementação em um ambiente real. Apesar de ter sido observada de forma ampla em diferentes aspectos, é importante validar a implementação em cenários reais.

Ainda, propõe-se como trabalho futuro a possibilidade da extensão da solução em outras aplicações, como telefonia móvel digital, entre outros.

REFERÊNCIAS

BARBIERI, R.; BRUSCHI, D.; ROSTI, E. Voice over IPsec: analysis and solutions. In: ANNUAL COMPUTER SECURITY APPLICATIONS, ACSAC, 18., 2002. **Proceedings...** Los Alamitos: IEEE Computer Society, 2002. p.261–270.

BAUGHER, M. et al. **The Secure Real-time Transport Protocol (SRTP)**: RFC 3711. [S.l.]: Internet Engineering Task Force, Network Working Group, 2004.

BLAKE-WILSON, S. et al. **Transport Layer Security (TLS) Extensions**: RFC 4366. [S.l.]: Internet Engineering Task Force, Network Working Group, 2006.

CISCO. **Understanding Delay in Packet Voice Network**. Disponível: <http://www.cisco.com/warp/public/788/voip/delay-details.html>. Acesso em: out. 2007.

CISCO. **Playout Delay Enhancements**. Disponível: <http://www.cisco.com/en/US/docs/ios/12_1t/12_1t5/feature/guide/dt_pod.html>. Acesso em: out. 2007.

CISCO. **Voice Over IP - Per Call Bandwidth Consumption**. Disponível: <http://www.cisco.com/warp/public/788/pkt-voice-general/bwidth_consume.html>. Acesso em: out. 2007.

COLE, R. G.; ROSENBLUTH, J. H. Voice over IP performance monitoring. **SIGCOMM Comput. Commun. Rev.**, New York, NY, USA, v.31, n.2, p.9–24, 2001.

EL-KHATIB, K. et al. **Multiplexing Scheme for RTP Flows between Access Routers**: Internet Draft. [S.l.]: Internet Engineering Task Force, Network Working Group, 2000.

FRANKS, J. et al. **HTTP Authentication - Basic and Digest Access Authentication**: RFC 2617. [S.l.]: Internet Engineering Task Force, Network Working Group, 1999.

FRIEDMAN, T.; CACERES, R.; CLARK, A. **RTP Control Protocol Extended Reports (RTCP XR)**: RFC 3611. [S.l.]: Internet Engineering Task Force, Network Working Group, 2003.

GILBERT, E. N. Capacity of a Burst-Noise Channel. **Bell Syst.Tech.J**, [S.l.], p.1253–1265, 1960.

HANDLEY, M. et al. **The Internet multimedia conferencing architecture**: Internet Draft. [S.l.]: Internet Engineering Task Force, Network Working Group, 1997.

HANDLEY, M.; JACOBSON, V. **SDP - Session Description Protocol**: RFC 4566. [S.l.]: Internet Engineering Task Force, Network Working Group, 1998.

HANDLEY, M.; PERKINS, C.; WHELAN, E. **Session Announcement Protocol**: RFC 2974. [S.l.]: Internet Engineering Task Force, Network Working Group, 2000.

HARDY, W. C. **QoS**: measurement and evaluation of telecommunications quality of service. New York, NY, USA: John Wiley & Sons, 2001. Preface By-Luis Cardoso.

HONG, K. et al. Impacts of Security Protocols on Real-Time Multimedia Communication. In: INTERNATIONAL WORKSHOP ON INFORMATION SECURITY APPLICATIONS, WISA, 5., 2004. **Revised Selected Papers...** Berlin: Springer, 2004.

INTERNATIONAL TELECOMMUNICATION UNION. **ITU-T Recommendation P.59**: artificial conversational speech. [S.l.], 1993.

INTERNATIONAL TELECOMMUNICATION UNION. **ITU-T Recommendation P.800**: methods for subjective determination of transmission quality. [S.l.], 1996.

INTERNATIONAL TELECOMMUNICATION UNION. **ITU-T Recommendation H.323**: packet-based multimedia communication. [S.l.], 2000.

INTERNATIONAL TELECOMMUNICATION UNION. **ITU-T Recommendation G.113 Appendix I**: provisional planning values for the equipment impairment factor ie. [S.l.], 2001.

INTERNATIONAL TELECOMMUNICATION UNION. **ITU-T Recommendation G.113 Appendix I**: provisional planning values for the equipment impairment factor ie and packet-loss robustness factor bpl. [S.l.], 2002.

INTERNATIONAL TELECOMMUNICATION UNION. **ITU-T Recommendation G.113**: transmission impairments due to speech processing. [S.l.], 2002.

INTERNATIONAL TELECOMMUNICATION UNION. **ITU-T Recommendation H.235**: security and encryption for h-series (h.323 and other h.245-based) multimedia terminals. [S.l.], 2003.

INTERNATIONAL TELECOMMUNICATION UNION. **ITU-T Recommendation G.114**: one-way transmission time. [S.l.], 2003.

INTERNATIONAL TELECOMMUNICATION UNION. **ITU-T Recommendation G.107**: the e-model, a computational model for use in transmission planning. [S.l.], 2005.

INTERNATIONAL TELECOMMUNICATION UNION. **ITU-T Recommendation G.191**: itu-t software tool library 2005 user's manual. [S.l.], 2005.

INTERNATIONAL TELECOMMUNICATION UNION. **ITU-T Recommendation Y.1541**: network performance objectives for ip-based services. [S.l.], 2006.

INTERNET2 VIDMID VIDEO MIDDLEWARE GROUP. **A Framework of Requirements, Threat Models, and Security Services for Videoconferencing over Internet2**. [S.l.], 2005.

JONSSON, L. E. **RObust Header Compression (ROHC) Terminology and Channel Mapping Examples**: RFC 3759. [S.l.]: Internet Engineering Task Force, Network Working Group, 2004.

KENT, S.; ATKINSON, R. **Security Architecture for the Internet Protocol**: RFC 2401. [S.l.]: Internet Engineering Task Force, Network Working Group, 1998.

KENT, S.; ATKINSON, R. **IP Authentication Header**: RFC 2402. [S.l.]: Internet Engineering Task Force, Network Working Group, 1998.

KENT, S.; ATKINSON, R. **IP Encapsulating Security Payload (ESP)**: RFC 2406. [S.l.]: Internet Engineering Task Force, Network Working Group, 1998.

KIM, H.; KANG, I.; HWANG, E. Measurement-Based Multi-Call Voice Frame Grouping in Internet Telephony. **IEEE Communications Letters**, [S.l.], v.6, n.5, May 2005.

KOREN, T. et al. **Enhanced Compressed RTP (CRTP) for Links with High Delay, Packet Loss and Reordering**: RFC 3545. [S.l.]: Internet Engineering Task Force, Network Working Group, 2003.

KUHN, D. R.; WALSH, T. J.; FRIES, S. **Security Considerations for Voice Over IP System**. [S.l.]: National Institute of Standards and Technology, 2005.

LUSTOSA, L. C. G. et al. Utilização do Modelo E para avaliação da qualidade da fala em sistemas de comunicação baseados em voz sobre IP. In: SIMPóSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUíDOS, SBRC, 22., 2004. **Anais...** Gramado: II/UFRGS, 2004.

PAZHYANNUR, R.; ALI, I. **Multiplexing Compressed RTP/UDP Packets in a PPP Frame**: Internet Draft. [S.l.]: Internet Engineering Task Force, Network Working Group, 1999.

PAZHYANNUR, R.; ALI, I.; FOX, C. **PPP Multiplexing**: RFC 3153. [S.l.]: Internet Engineering Task Force, Network Working Group, 2001.

PEREIRA, R. M.; TAROUCO, L. M. R. Um Novo Método Para Multiplexação Adaptativa de Ligações VoIPSec com Qualidade Baseado no E-Model. In: BRAZILIAN SYMPOSIUM ON MULTIMEDIA AND THE WEB, WEBMEDIA, 13., 2007, Gramado, Porto Alegre, Brasil. **Proceedings...** New York: ACM, 2007.

RAMJEE, R. et al. Adaptive Playout Mechanisms for Packetized Audio Applications in Wide-Area Networks. In: NETWORKIN FOR GLOBAL COMMUNICATIONS, INFOCOM, 13., 1994. **Proceedings...** [S.l.]:IEEE, 1994. v.2, p.680–688.

RAMSDELL, B. **Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification**: RFC 3851. [S.l.]: Internet Engineering Task Force, Network Working Group, 2004.

ROSENBERG, J. et al. **SIP - Session Initiation Protocol**: RFC 3261. [S.l.]: Internet Engineering Task Force, Network Working Group, 2002.

ROSENBERG, J.; SCHULZRINNE, H. **An RTP Payload Format for User Multiplexing**: Internet Draft. [S.l.]: Internet Engineering Task Force, Network Working Group, 1999.

SCHULZRINNE, H. et al. **RTP A Transport Protocol for Real-Time Applications**: RFC 3550. [S.l.]: Internet Engineering Task Force, Network Working Group, 2003.

SUBBIAH, B.; SENGODAN, S. **User Multiplexing in RTP payload between IP Telephony Gateways**: Internet Draft. [S.l.]: Internet Engineering Task Force, Network Working Group, 1999.

SUN, L.; IFEACHOR, E. C. Voice Quality Prediction Models and Their Application in VoIP Networks. **IEEE Transactions on Multimedia**, [S.l.], v.8, n.4, p.809–820, 2006.

SZE, H. P. et al. A Multiplexing Scheme for H.323 Voice-Over-IP Applications. **IEEE Journal on Selected Areas in Communications**, [S.l.], v.20, n.7, Sept. 2002.

TANENBAUM, A. S. **Redes de computadores**. [S.l.]: Elsevier, 2003.

TANIGAWA, K.; HOSHI, T.; TSUKADA, K. **Simple RTP Multiplexing Transfer Methods for VoIP**: Internet Draft. [S.l.]: Internet Engineering Task Force, Network Working Group, 1999.

TELECOMMUNICATIONS INDUSTRY ASSOCIATION. **TSB-116-A Voice Quality Recommendations for IP Telephony**. [S.l.], 2006.

THOMSPON, B.; KOREN, T.; WING, D. **Tunneling multiplexed compressed RTP (TCRTP)**: RFC 4170. [S.l.]: Internet Engineering Task Force, Network Working Group, 2005.

TRAD, A.; AFIFI, H. TFMC a TCP-Friendly Multiplexing Control Scheme for VoIP Flow Transmission. In: INTERNATIONAL CONFERENCE ON NETWORKING AND SERVICES, 3., 2004. **Proceedings...** [S.l.: s.n.], 2004.

VOIPSA. **VoIP security and privacy threat taxonomy**. [S.l.], 2005.

WALSH, T. J.; KUHN, D. R. Challenges in Securing Voice over IP. **IEEE Security and Privacy**, Piscataway, NJ, USA, v.3, n.3, p.44–49, 2005.

YAMADA, H.; FUKUMOTO, N. Speech Quality Transmitted by Circuit Multiplication Equipment Optimized for IP-Based Networks (IP-CME). **IEICE Transactions**, [S.l.], v.89-B, n.2, p.490–499, 2006.