Universidade de São Paulo

Escola de Engenharia de São Carlos
Departamento de Engenharia Elétrica

Estudo dos Fundamentos de Sistemas de Áudio e Vídeo Conferência de Alto Desempenho

Davi Nóbrega de Sousa

São Carlos - SP

Estudo dos Fundamentos de Sistemas de Áudio e Vídeo Conferência de Alto Desempenho

Davi Nóbrega de Sousa

Orientadora: Profª. Drª. Mônica de Lacerda Rocha

Monografia referente ao Trabalho de Conclusão de Curso dentro do escopo da disciplina SEL0626 – Projeto de Formatura II do Departamento de Engenharia Elétrica da Escola de Engenharia de São Carlos – EESC/USP.

Área de Concentração: Telecomunicações

USP – São Carlos
Novembro/2012

S725e

Sousa, Davi Nóbrega de
Estudo dos Fundamentos de Sistemas de Áudio e Vídeo Conferência de Alto Desempenho / Davi Nóbrega de Sousa; orientadora Mônica de Lacerda Rocha. São Carlos, 2012.

Monografia (Graduação em Engenharia de Computação) -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2012.

1. vídeo conferência. 2. RTP. 3. SIP. 4. H.323. I. Título.

FOLHA DE APROVAÇÃO

Nome: Davi Nóbrega de Sousa

Título: "Estudo dos Fundamentos de Sistemas de Áudio e Vídeo Conferência de Alto Desempenho"

Trabalho de Conclusão de Curso defendido em 28/11/2012.

Comissão Julgadora:	**Resultado:**
Prof. Dr. Marcelo Basílio Joaquim SEL/EESC/USP	Aprovado
Prof. Dr. Maximiliam Luppe SEL/EESC/USP	Aprovado

Orientadora:
Profa. Dra. Mônica de Lacerda Rocha - SEL/EESC/USP

Coordenador pela EESC/USP do Curso de Engenharia de Computação:

Prof. Associado Evandro Luís Linhari Rodrigues

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Siglas

ACF	*Admission Confirm*
ALG	*Application Layer Gateway*
ARJ	*Admission Reject*
ARQ	*Admission Request*
BCF	*Bandwidth Confirm*
BRJ	*Bandwidth Reject*
BRQ	*Bandwidth Request*
CLC	*Close Logical Channel*
CP	*Continuous Presence*
CSRC	*Contributing Source Identifiers*
DCF	*Disengage Confirm*
DCT	*Discrete Cosine Transform*
DDoS	*Distributed Denial of Service*
DoS	*Denial of Service*
DRQ	*Disengage Request*
DTMF	*Dual-Tone Multi-Frequency*
GKRCS	*Gatekeeper Routed Call Signaling*
HIPS	*Host-based Intrusion Prevention System*
IIR	*Infinite Impulse Response*
IVR	*Interactive Voice Response*
MCU	*Multipoint Control Unit*
MitM	*Man-in-the-Middle*
MSD	*Master-Slave Determination*
NAT	*Network Address Translation*
NTP	*Network Time Protocol*
OLC	*Open Logical Channel*
PSTN	*Public Switched Telephone Network*
QoS	*Quality of Service*

RCF	*Registration Confirm*
RR	*Receiver Report*
RRJ	*Registration Reject*
RRQ	*Registration Request*
RTCP	*RTP Control Protocol*
RTP	*Real-time Transport Protocol*
SDES	*Source Description*
SDP	*Session Description Protocol*
SID	*Silence Detection*
SIP	*Session Initiation Protocol*
SNR	*Signal-to-Noise Ratio*
SR	*Sender Report*
SSRC	*Synchronization Source*
SVC	*Scalable Video Coding*
TCP	*Transmission Control Protocol*
TCS	*Terminal Capability Set*
UAC	*User Agent Client*
UAS	*User Agent Server*
UDP	*User Datagram Protocol*
URI	*Uniform Resource Identifier*
VAD	*Voice Activity Detection*
VAS	*Voice-activated Switched*
VFU	*Video Fast Upgrade*
VLC	*Variable Lenght Coding*
VPN	*Virtual Private Network*

Resumo

Este trabalho objetiva a descrição e estudo de fundamentos de conferência de áudio e vídeo no caminho fim a fim da informação. Os diversos dispositivos existentes bem como tipos de estabelecimento de vídeo conferências, com ou sem alocação prévia de recursos, utilizam uma gama de recomendações para uma experiência de qualidade. As padronizações *Session Initiation Protocol* e H.323, bem como *Real-Time Protocol*, possibilitam estabelecer chamadas e canais de transmissão de mídia entre os dispositivos do sistema de vídeo conferência, seja em arquiteturas distribuídas ou centralizadas, e um mecanismo de sincronização fundamentado na utilização de bases de tempo comum possibilita ainda a obtenção de alinhamento de áudio e vídeo. Na intenção de proteger as informações e usuários, a presença de *firewalls*, outros componentes e implementação de *Network Address Translation* promovem a segurança do sistema.

Palavras chave: fundamentos de vídeo conferência, estabelecimento de vídeo conferência, SIP, H.323, RTP.

Abstract

This paper concentrates on describing the audio and video conferencing fundamentals involved in the end to end information's path. The various devices and types of conference establishment, with or without resource allocation, use a range of recommendations to achieve a quality experience. Standardizations *Session Initiation Protocol* and H.323, and *Real-Time Protocol*, allow the establishment of calls and media streaming channels between devices of the conference system, regard the use of centralized or distributed architectures, and a synchronization mechanism based on the use of common time basis also allows obtaining audio and video alignment. With the intent information and users protection, the presence of firewalls, other components and *Network Address Translation* implementation promotes system security.

Keywords: videoconferencing fundamentals, videoconferencing establishment, SIP, H.323, RTP.

Introdução

A redução no custo de transporte no âmbito das comunicações e o avanço do *hardware* dedicado a essas atividades tem possibilitado o uso de avançadas técnicas fornecendo voz e vídeo com alta qualidade e boa relação custo-benefício. Neste contexto, a vídeo conferência se transforma numa ferramenta importante que permite a interação face a face, proporciona uma ótima experiência no âmbito da colaboração, através da transmissão e recepção de voz e vídeo de qualidade, e é capaz de suprir as necessidades do mundo corporativo com credibilidade.

Em particular, tendo como motivação a oportunidade de realizar um estágio supervisionado, para obtenção do título de Engenheiro da Computação, em uma empresa integradora que vivencia uma grande demanda por soluções desse tipo de colaboração para atender uma diversa gama de clientes dos mais variados segmentos, o estudo de tecnologias relacionadas à vídeo conferência tornou-se atraente e relevante, o que resultou na elaboração desta monografia de Conclusão de Curso.

Este trabalho abordará os fundamentos de videoconferência com base na obra de Firestone, Ramalingam & Fry [1], a qual se mostra bastante completa na abrangência dos conceitos gerais que regem um sistema de vídeo conferência, abordando desde os diversos métodos de se estabelecer uma experiência desse tipo e passando por todos os aspectos e conceitos envolvidos com o funcionamento fim a fim da atividade.

No capítulo 1 são abordados os tipos principais de vídeo conferência, tanto para estabelecimento como para operação, com diferenças, pontos positivos e negativos. São também abordados alguns dispositivos para vídeo conferência.

O capítulo 2 engloba o sistema de vídeo conferência, com a descrição de seus componentes e como estão estruturados em diferentes arquiteturas, de forma que a informação caminhe e seja tratada em cada etapa da experiência de colaboração.

O capítulo 3 contém a informação referente à codificação e à decodificação da informação, tanto para transmissão quanto para recepção. São abordados diferentes métodos de compressão da informação, comparação entre esses métodos e soluções mais adequadas para diferentes tipos de ambientes.

No capítulo 4 são tratados os conceitos que envolvem o controle e transporte da informação, abordando em maior detalhe os protocolos *Real-time Transport Protocol* (RTP) e *RTP Control Protocol* (RTCP).

No capítulo 5 são cobertos aspectos referentes aos protocolos de sinalização em sistemas de vídeo conferência, através do detalhamento e comparação dos protocolos *Session Initiation Protocol* (SIP) e H.323.

O capítulo 6 envolve um detalhamento sobre sincronização labial em sistemas de vídeo conferência, com alguns fundamentos base para a ocorrência dessa atividade.

Por fim, o capítulo 7 é responsável por abordar aspectos de segurança em sistemas de vídeo conferência, com descrição de arquiteturas e componentes que permitem a implementação dessa proteção.

Capítulo 1 - Tipos de conferências de áudio e vídeo

As conferências são definidas como chamadas, envolvendo apenas voz ou voz e vídeo, que agregam mais de dois participantes. Muito utilizadas em salas de reunião, são uma ótima ferramenta para discussões e trocas de informações principalmente no ambiente corporativo.

Basicamente, as vídeo conferências são divididas em três principais tipos: *Ad Hoc*, *Reservationless* e *Scheduled*. A principal diferença entre eles está na alocação de recursos para a vídeo conferência: tanto o modo *Ad Hoc* como *Reservationless* não permitem a alocação de recursos, enquanto no modo *Scheduled* os recursos são alocados préviamente. O fato de não envolverem alocação de recursos faz das vídeo conferências *Ad Hoc* e *Reservationless* mais práticas de serem estabelecidas. Porém, a reserva de recursos para uma atividade de vídeo conferência tem um papel bastante interessante e importante que é a garantia de qualidade do serviço durante a colaboração.

Os tipos de vídeo conferência têm sua complexidade aumentada seguindo a ordem em que foram citados, da *Ad Hoc*, a mais simples, até a *Scheduled*, mais complexa [2]. A forma com que vídeo conferências desse tipo são estabelecidas pode ser melhor entendida através de exemplos, que serão dados a seguir.

As vídeo conferências do tipo *Ad Hoc* são bastante simples e tem o funcionamento estabelecido da seguinte maneira: um participante X (*host*) estabelece uma chamada com participante Y e deseja incluir o participante Z na conversação. Para isso, pressiona um botão de conferência, colocando o participante Y em espera e estabelece uma chamada com o participante Z e, após estabelecida essa chamada, pressiona novamente o botão de conferência e dessa forma estarão os três participantes na chamada anteriormente estabelecida. Esses passos podem ser repetidos para incluir mais participantes, respeitando um número máximo definido em cada sistema.

Já o modo *Reservationless*, apesar de também não envolver a alocação de recursos assim como o modo *Ad Hoc*, tem seu funcionamento um pouco diferenciado. Nesse modo, uma vídeo conferência é agendada sem especificar requisitos para alocação de recursos, como duração do encontro ou número de participantes, sendo

assim também um tanto ágil. O organizador cria uma vídeo conferência através de uma interface de voz provida pelo sistema, interface essa denominada *Interactive Voice Reponse* (IVR). Ele então atribui nome e identificador para a vídeo conferência, e outros participantes podem então adentrar ao encontro.

Mais complexo que os dois modos explicitados anteriormente, o modo *Scheduled* pode não ser tão ágil de se estabelecer, mas tem a seu favor um ponto bastante importante que é a alocação prévia de recursos para a vídeo conferência. Com essa alocação, é possível garantir a qualidade do serviço durante a atividade, evitando perdas de qualidade quando existe um grande número de usuários utilizando o sistema de vídeo conferência, por exemplo. O estabelecimento desse tipo de vídeo conferência se dá pela seguinte maneira: através da integração com sistemas de calendário, agenda-se um encontro e definem-se parâmetros que serão utilizados para a reserva de recursos, como número esperado de participantes e duração do encontro. Então, através da discagem de um número referente à vídeo conferência agendada, os participantes podem adentrar ao encontro. É também bastante comum, e um tanto mais prático, o ingresso em vídeo conferências através de uma *Uniform Resource Locator* (URL) distribuída aos participantes, de forma que ao acessar tal *link* o sistema de vídeo conferência já tenta estabelecer a chamada com o usuário. Esses dois modos de adentrar em uma vídeo conferência são também aplicáveis ao modo *Reservationless*.

Uma vez dentro de uma vídeo conferência, os participantes podem em geral usufruir de diversas ferramentas que tornam a colaboração mais rica. Dentre essas ferramentas, uma muito útil é o compartilhamento de apresentações, através do qual os participantes conseguem ter acesso a uma apresentação elaborada e regida por um certo participante, aspecto bastante positivo para o entendimento do que está sendo discorrido num ambiente de reuniões e discussões. Outra ferramenta interessante é a subdivisão de uma vídeo conferência, em que os participantes podem ser subdivididos em outros grupos de forma que cada um desses subgrupos possa discutir diferentes aspectos de um mesmo assunto.

Além dos modos de se estabelecer uma vídeo conferência, existem alguns modos de operação que valem ser abordados. Têm-se dois modos que diferem, basicamente, em como será feita a exibição, são eles: ativação por voz (*Voice-activated Switched*, identificado pela sigla VAS) e presença contínua (*Continuous Presence*, identificado por CP).

O primeiro, VAS, gerencia qual participante será visto pelos outros através da energia do áudio de entrada. Basicamente, o participante que está falando num maior nível de energia será o participante visto pelos demais participantes. Já o modo de presença contínua, como o próprio nome define, exibe simultaneamente os participantes da conferência. A Figura 1 exemplifica a tela desse último modo de exibição.

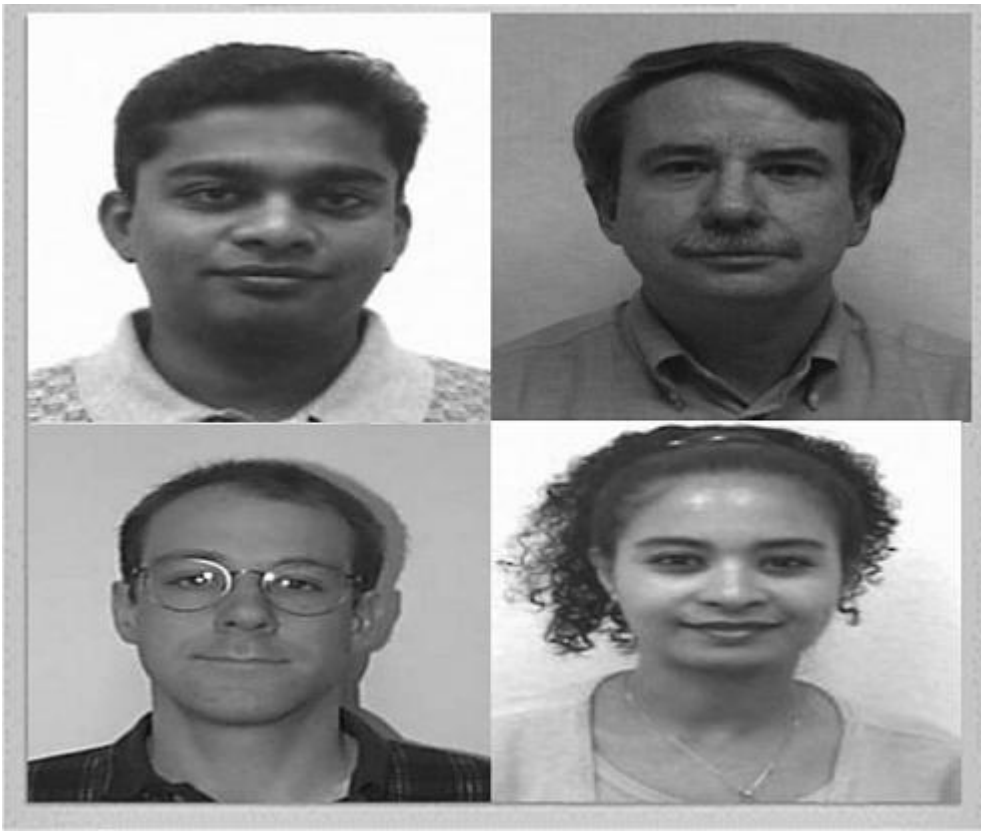

Figura 1 - Modo de exibição Presença Contínua, adaptada de [1].

Basicamente, diversos *streams* provenientes de cada participante são organizados em um único *stream*, o qual geralmente é um único para esses mesmos participantes. A disposição em que cada participante aparece, denominado *layout*, é customizável, e o número de participantes a serem vistos simultaneamente também envolve os recursos disponíveis no sistema: uma vez que todos os *streams* exibidos são decodificados, a exibição de diversos *streams* simultâneos exige alto processamento. A Figura 2 exibe diversas estruturas de *layout*, e a distribuição dos participantes nos quadrantes do *layout* pode ser fixa ou dinâmica.

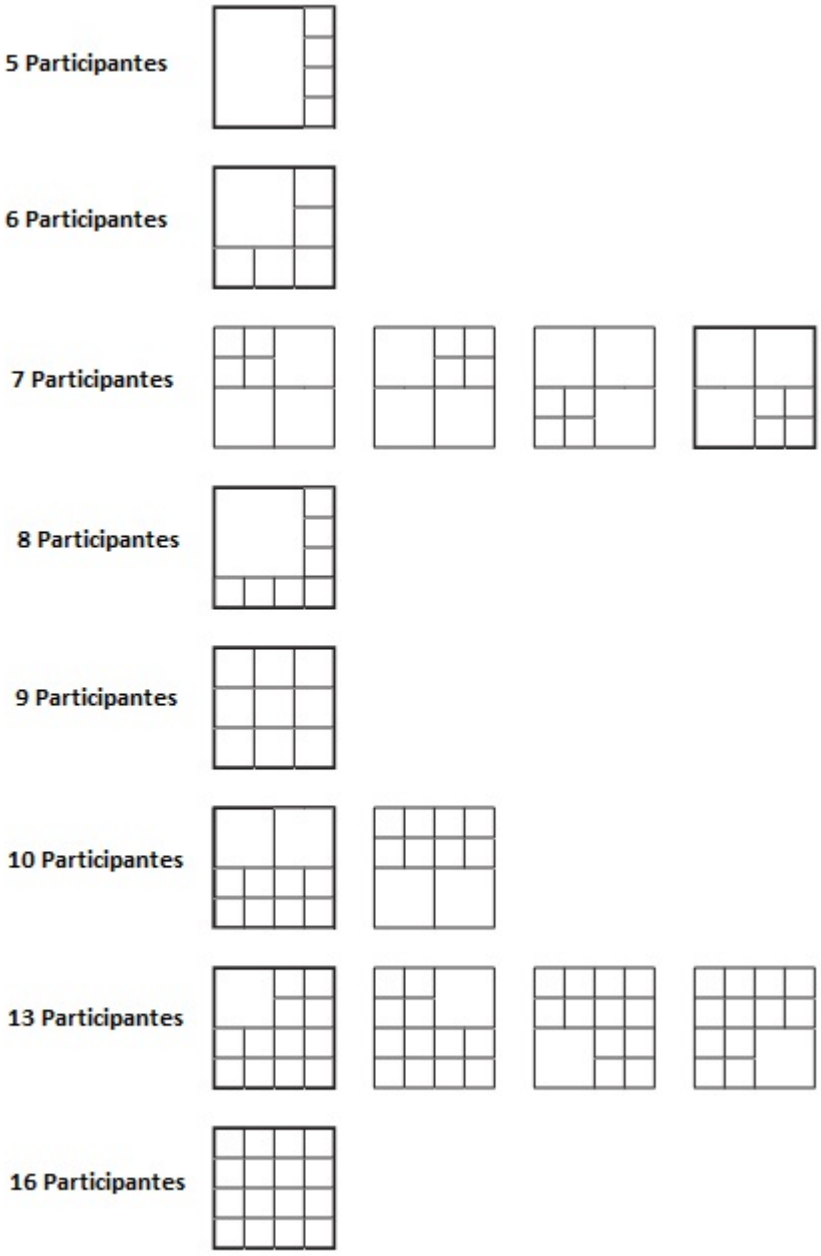

Figura 2 – *Layouts* de disposição dos participantes, adaptada de [1].

Uma variação nos modos de operação é o modo *Lecture Mode*, no qual um quadrante principal é fixo com um apresentador e os outros quadrantes menores contem os outros participantes. É comum também a combinação com ativação por voz.

Para o estabelecimento e operação das conferências, participantes fazem uso de alguns dispositivos, dos mais simples aos de última geração. As diferenças entre os diversos dispositivos envolvem basicamente taxas de transferência de bits, resolução da

imagem, tamanho de telas, tipos de câmera e qualidade de som. É possível estabelecer uma vídeo conferência através de um computador pessoal de mesa, geralmente com uma câmera de baixa qualidade e todo processamento atribuído ao processador do próprio computador, com nenhum *hardware* dedicado. Como de se esperar, a qualidade da experiência não é das melhores.

No outro extremo, estão os sistemas de última geração, tratados como Sistemas de Telepresença. Esses sistemas fornecem uma experiência de altíssima qualidade aos usuários, com *hardware* dedicado à aplicação, câmeras de alta definição, telas grandes, som espacial e ainda toda uma preparação do ambiente envolvendo iluminação e padrões de cores das paredes e modelos de mesas e cadeiras. A Figura 3 ilustra um sistema de telepresença.

Figura 3 - Sistema de Telepresença [3]

Capítulo 2 - Arquitetura e componentes

A experiência de colaboração via vídeo conferência torna-se possível através de toda uma arquitetura do sistema de conferência, com diversos componentes, cada um com seu papel de forma que interagem entre si possibilitando a entrega de uma experiência adequada. Nesse capítulo, serão abordados os diversos componentes do sistema de vídeo conferência e como esse sistema é estruturado.

O sistema de vídeo conferência, conforme ilustrado na Figura 4, é composto basicamente de quatro camadas: interface do usuário, *conference control*, *control plane* e *media plane*. Nos próximos parágrafos será explicado qual o papel de cada uma dessas camadas na arquitetura do sistema de conferência.

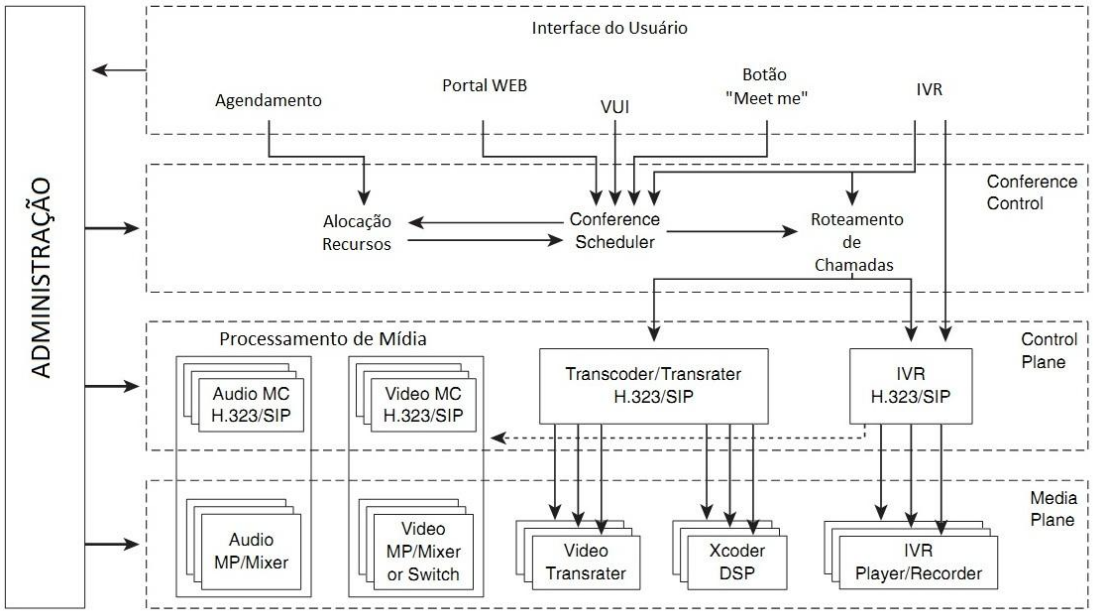

Figura 4 - Camadas do sistema de conferência, adaptada de [1].

A interface do usuário é a visão de mais alto nível do sistema de vídeo conferência. Como o próprio nome já diz, é através dela que o usuário interage com o sistema, com a possibilidade de agendamento e criação de vídeo conferências e execução dos comandos quando em conferência (mudo, por exemplo). Como exemplo, podem ser citados portais *World Wide Web* (WEB) e a interface IVR, a qual através de

uma sequência de instruções de voz gravadas guia o usuário ao adentrar em uma vídeo conferência.

Abaixo da camada de interface do usuário, segue a camada *conference control*. Essa camada é responsável pela alocação dos recursos para uma vídeo conferência e pelo roteamento das chamadas. Existe um grande desafio nessa camada, que se refere ao gerenciamento dos recursos levando em consideração diferentes características de *streams* de cada participante. Há uma grande diversidade de dispositivos, com as mais diversas características, os quais permitem diferentes qualidades de experiência. Uma vez que a alocação de recursos se dá através da definição de número de usuários e duração da vídeo conferência, não é possível prever quantos usuários participarão da vídeo conferência com utilização de codificação e decodificação mais complexas, as quais exigem maior poder computacional e pode não haver recurso suficiente alocado para atender a essa demanda. Uma solução é a definição de uma máxima taxa de transferência de bits (*bit rate*) à qual todos os participantes, independentemente do dispositivo que utilizam, estarão sujeitos.

A próxima camada, *control plane*, é responsável por estabelecer a sinalização com cada dispositivo participante de uma vídeo conferência. Nessa camada são negociados os tipos de mídia e é definida a conexão dos dispositivos com os *mixers* pertencentes à camada mais baixa, *media plane*. Para sinalização, são utilizados dois protocolos, *Session Initiation Protocol*, ou simplesmente SIP, e H.323. Maiores detalhes desses protocolos serão dados em capítulos posteriores.

A camada *control plane* basicamente abre portas H.323 e SIP e aguarda por conexões. Quando um dispositivo se conecta a uma dessas portas, o tipo de mídia a ser trocado é negociado e abre-se um canal lógico entre o dispositivo e o *control plane*, pelo qual são transferidos os dados de mídia. É interessante ressaltar o controle que essa camada faz no caso de falhas de conexão: quando da ocorrência de uma falha, a camada *control plane* avisa as camadas mais baixas de forma que o recurso alocado à conexão que se encerrou seja liberado.

No mais baixo nível do esquemático ilustrado na Figura 4 está a camada *media plane*. É nela que são tratados os dados de mídia que estão sendo enviados e recebidos pelos dispositivos. Como é possível observar na figura, estão presentes nessa camada os *mixers* de áudio e vídeo, aos quais será dada maior ênfase.

Antes que seja abordado um pouco mais da função dos *mixers* na camada *media plane*, é importante o entendimento de duas transformações: *transcoding* e *transrating*.

Por *transcoding* entende-se a transformação de um formato de mídia para outro. Cada dispositivo tem suas características e suporte a formatos de mídia diferentes, e dessa maneira, a mídia proveniente de cada dispositivo deve ser devidamente transformada para que seja adequada aos padrões dos outros dispositivos. Como exemplo, pode ser citada a transformação do formato MPEG-2 para H.264.

Já a operação de *transrating* refere-se ao ajuste da taxa de transferência de bits. Esse processo será melhor entendido através do vídeo *transrater*.

Vídeo *transrater* é um componente chave na estrutura de um sistema de videoconferência que integra dispositivos de diversas redes (móvel, banda larga, LAN). Uma vez que cada dispositivo esteja em uma rede distinta, transfere dados de vídeo em taxas de transferência de bit diferentes, é então necessário que essas taxas sejam ajustadas para que sejam devidamente recebidas pelos outros dispositivos. O papel do vídeo *transrater* se encaixa nesse elo: converter a taxa de transferência de bits do *stream* de banda mais alta para *stream* de banda mais baixa. O caminho contrário não é necessário, uma vez que dispositivos de uma banda mais alta sempre aceitam *streams* de banda mais baixa. A Figura 5 ilustra o papel desempenhado pelo vídeo *transrater*.

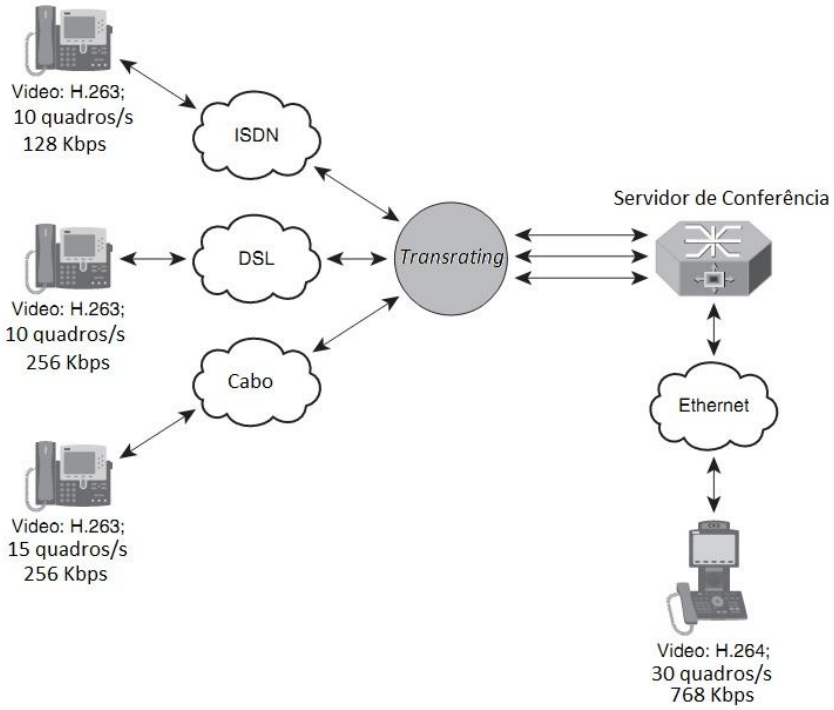

Figura 5 – Esquemático da operação *Transrater*, adaptada de [1].

Além das limitações de banda referente à rede em que cada dispositivo está conectado, outro fator limitante para taxas de transferência de bits mais baixas pode ser o poder de processamento dos dispositivos.

Em um mais baixo nível, a operação de *transrating* tem seu funcionamento da seguinte maneira: os pacotes RTP, *Real-Time Transport Protocol* (pacotes RTP serão melhor abordados nos próximos capítulos), contendo os dados de vídeo são primeiramente armazenados num *buffer*, para que sejam ordenados, uma vez que podem chegar fora de ordem. Esses dados são então decodificados, o que dá origem aos dados no *Raw Picture buffer*. Então, os dados desse último bloco são recodificados a uma menor taxa de transferência de bits e reempacotados em pacotes RTP. O diagrama da Figura 6 exemplifica essa transformação.

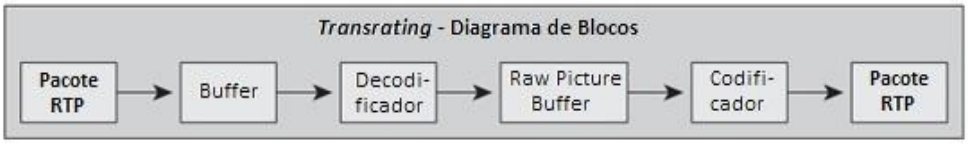

Figura 6 - Diagrama de *Transrating*, adaptada de [1].

De forma semelhante ao vídeo *transrater*, outro componente é o vídeo *transcoder*. Esse componente tem a responsabilidade de conversão de características de vídeo como o formato de codificação, resolução, taxa de quadro e taxa de bits. A transformação dessa última característica, taxa de bits, é a transformação de *transrating* já mencionada. Como pode ser inferido, o vídeo *transcoder* possui a função de *transrating* embutida. A Figura 7 ilustra o diagrama do processo de *transcoding* em mais baixo nível.

Figura 7 - Diagrama de *Transcoding* em mais baixo nível, adaptada de [1].

Na Figura 7, é possível observar a capacidade de execução da função de *transrating* comparando-a com o diagrama da Figura 6. A diferença principal é a presença de um bloco adicional de escala, que possibilita ajustes de resolução, formatos de codificação e outras características.

Mixer de áudio

O *mixer* de áudio é o componente que tem o papel de selecionar *streams* de voz que chegam dos participantes da vídeo conferência e criar na saída um *stream* que contém a soma dos *streams* de entrada. Basicamente, são selecionados os *streams* de entrada com maior energia, três ou quatro no máximo. Esses *streams* são então somados e é então gerado um *stream* de saída que é distribuído aos participantes. O fato de serem selecionados apenas três ou quatro *streams* de entrada para geração de apenas um de saída contendo todos é devido ao fato de o ouvido humano possuir a limitação de não distinguir uma quantidade maior de áudios diferentes ao mesmo tempo. A Figura 8 representa um diagrama de blocos detalhado do *mixer* de áudio.

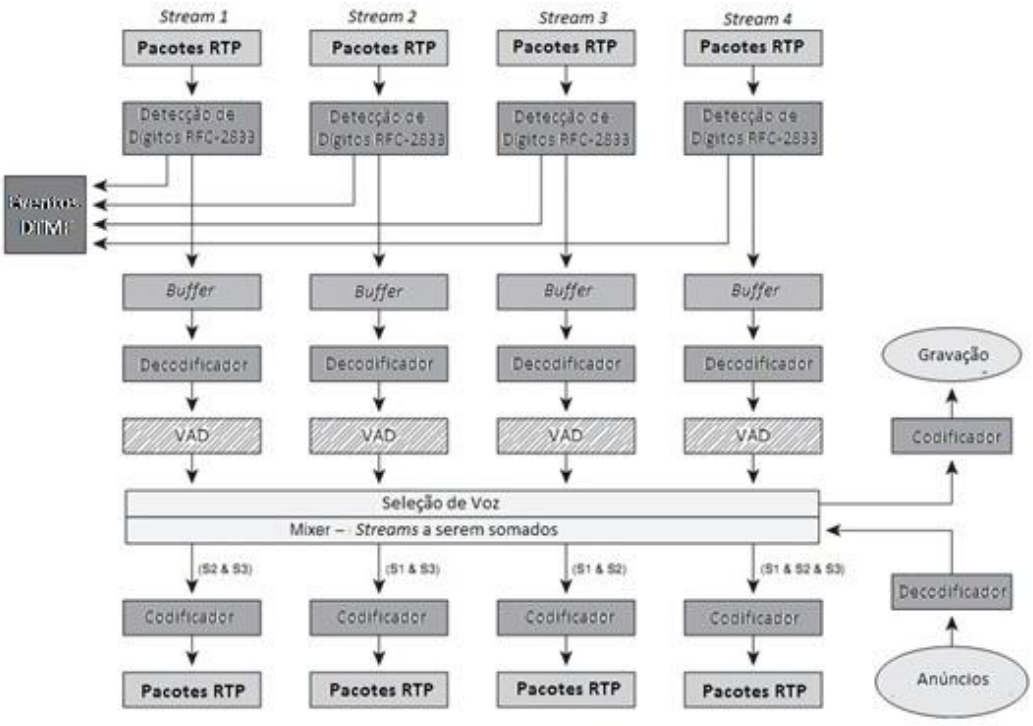

Figura 8 - Diagrama do fluxo da informação em *mixer* de áudio, adaptada de [1].

A seguir será dada uma breve explicação de como as informações percorrem os blocos e o que ocorre em cada um deles.

Assim que os pacotes RTP chegam, se deparam com o bloco RFC-2833 *Digit Detection*, o qual identifica se o pacote RTP contem dados de sinalização *Dual-Tone Multi-Frequency* (DTMF) e, caso positivo, direciona esses dígitos de eventos DTMF para o bloco DTMF *Event handler*, o qual encaminhará as informações para a interface adequada à execução da função. Caso o pacote não contenha dígitos de sinalização DTMF, é diretamente encaminhado ao *buffer*.

O *buffer* tem a função de armazenar os pacotes que chegam e transformá-los em um único *stream* uniforme, tratando problemas como o recebimento de pacotes duplicados, pacotes fora de ordem e pacotes que chegam em intervalos de tempo variados. Esse *buffer* pode ser dinâmico ou estático, com tamanhos variáveis ou fixos, respectivamente. *Buffers* de maior tamanho possuem maior proteção contra estouro do *buffer*, porém apresentam maior latência. O inverso ocorre para *buffers* de menor tamanho: têm uma menor latência, porém menor proteção contra estouro. Assim, no âmbito de melhor desempenho, é interessante que os *buffers* possuam um equilíbrio ideal, de forma a possuírem o menor tamanho possível que permita a proteção contra estouro. É esse equilíbrio que os *buffers* dinâmicos buscam atingir: com a execução de algoritmos responsáveis por uma estimativa contínua dos tempos de chegada e atrasos dos pacotes, os *buffers* são ajustados para suportar da melhor maneira o padrão de fluxo da informação que se tem no momento [4].

Após a informação ser armazenada e tratada no *buffer*, passa pela decodificação e então atinge o bloco *Voice Activity Detection* (VAD). Esse bloco tem uma tarefa interessante para otimização da rede provendo economia de banda utilizada. No VAD, pacotes RTP com voz de baixa energia são substituídos por *silence detection* (SID), que corresponde a um pacote de silêncio. Essa atividade é realizada do lado da informação enviada. Quando se recebe a informação, pode haver também um VAD, que terá o papel de identificar se um pacote é ruído ou pacote de silêncio, e assim tomar as devidas providências como o descarte do pacote.

Após passar pelo VAD, os dados atingem o módulo *Speaker Selection*, o qual tem a responsabilidade de examinar os *streams* que chegam de forma a selecionar os adequados a serem somados para compor o *stream* de saída.

Como mencionado, são selecionados os três *streams* de maior energia para comporem o *stream* de saída. Quando o número de *streams* de entrada é menor que três, são diretamente incluídos no *stream* de saída, caso contrario é executado um algoritmo para que sejam selecionados. É interessante ainda observar que, mesmo que esteja entre os três de maior energia, o *stream* de um usuário nunca fará parte do *stream* de saída que será recebido por ele mesmo, de forma a evitar eco.

Por fim, a informação atinge o bloco de codificação, no qual serão comprimidos segundo um algoritmo negociado e posteriormente empacotados em pacotes RTP para serem enviados.

Arquiteturas

Quando se fala em arquiteturas de sistemas de vídeo conferência, têm-se basicamente dois tipos: centralizadas e distribuídas.

As arquiteturas centralizadas são mais simples de serem implementadas, com a utilização de um único servidor que fornece os serviços e toda a parte de gerenciamento. Já as arquiteturas distribuídas fornecem uma flexibilidade e distribuição da carga de rede, uma vez que as funcionalidades estão distribuídas entre diversos dispositivos. Porém, as arquiteturas distribuídas são bastante complexas, uma vez que deve haver todo um sistema de sinalização entre os dispositivos para que trabalhem de forma transparente como se fossem apenas um único dispositivo.

Capítulo 3 - Princípios de codificação e compressão

Antes de serem empacotados e enviados, os dados sofrem o processo de codificação e compressão. Neste capítulo abordaremos alguns métodos de compressão, formatos suportados e outros aspectos relacionados a esta etapa de todo o fluxo da informação em um sistema de vídeo conferência.

Antes de se estabelecer uma chamada de vídeo conferência, os dispositivos negociam entre si alguns aspectos da mídia que será trocada, e uma dessas características é o formato de vídeo, o qual deve ser suportado pelo codificador/decodificador dos dois dispositivos, que inclui taxa de transferência de bits e de quadros.

Dispositivos mais complexos suportam áudio e vídeo de melhor qualidade, porém requerem um maior poder de processamento. Além disso, quando se trata de altas taxas de transferência de bits, a experiência consome maior largura de banda, e essa taxa de transferência de bits também é uma característica negociada entre os dispositivos antes de se iniciar uma chamada. Uma vez que essa taxa é negociada, durante a chamada as variações de qualidade e taxas de quadro são ajustadas na codificação para que caibam na taxa de transferência de bits negociada anteriormente.

Quando se abrange codificação, está implícita a execução de um pré-processamento, codificação, decodificação e pós-processamento.

O pré-processamento tem como principal função a eliminação de ruído e de informação imperceptível ao olho humano. Para eliminação de ruídos, é comum a utilização de filtros de resposta ao impulso infinita (*Infinite Impulse Response*, IIR). Já com relação às informações não perceptíveis ao olho humano, parte-se do fato de o sistema visual humano ter menor percepção de mudanças em regiões com alto grau de movimento. Sendo assim, é tipicamente usado um filtro passa baixa para “embaçar” a imagem nos pontos com alto grau de movimento. Esse grau de movimento é calculado *pixel* por *pixel* e comparado com um valor limite para se enquadrar ou não nessa definição.

Após o pré-processamento, os dados são então codificados. Há basicamente dois principais tipos de codificação, utilizando-se apenas informação do próprio quadro ou informação de outros quadros. Quando da utilização apenas da informação do próprio quadro para codificação dele mesmo, temos então um quadro independente referenciado como *intraframe*, *I-frame* ou quadro I. A Figura 9 ilustra o fluxo do processo de codificação/decodificação de um quadro I.

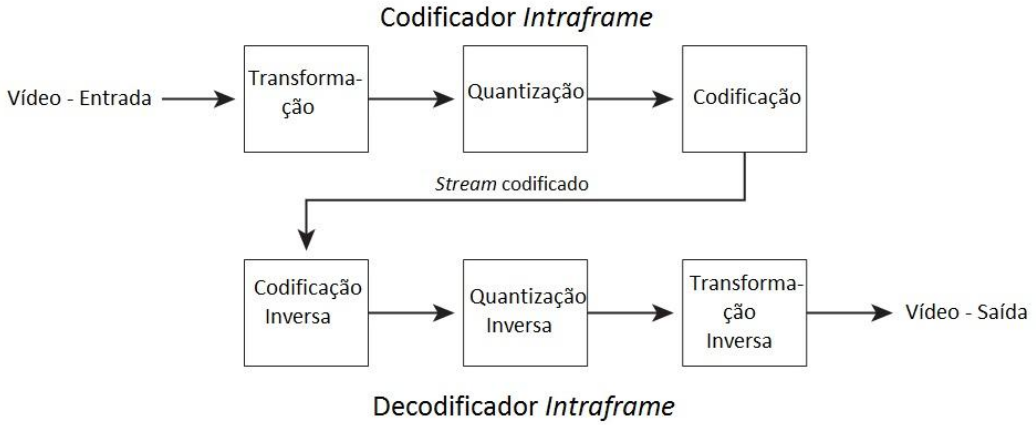

Figura 9 - Codificação e Decodificação de quadro I, adaptada de [1].

Conforme ilustrado na Figura 9, os dados de vídeo passam por uma transformação, seguida de quantização e então codificação. No lado da decodificação, é percorrido o caminho inverso.

Na etapa de transformação, a imagem é dividida em blocos de 8x8 *pixels* ou 4x4 *pixels* e então transformada do domínio espacial para o domínio de frequências. Essa transformação entre domínios é realizada através de Transformada Cosseno Discreta (DCT). A DCT é um tanto complexa, porém apresenta boa precisão. Contrariamente, existem métodos de transformação baseados em matemática mais simples, com complexidade mais baixa, porém menor precisão. A utilização de um tipo ou outro de transformação, bem como de divisão em blocos, varia entre os diferentes dispositivos.

Os coeficientes resultados da aplicação da DCT representam padrões de frequência. Coeficientes próximos do canto superior esquerdo do bloco correspondem a padrões de baixa frequência, ou seja, que variam pouco, também referenciados como coeficientes DC, em analogia à corrente contínua, em inglês *direct current*. Coeficientes na região superior direita correspondem a padrões de alta frequência horizontal, devido

à presença de bordas verticais. No canto inferior esquerdo, estão coeficientes que correspondem a alta frequência vertical, devido a bordas horizontais. E por fim, na região inferior direita estão os coeficientes resultantes de altas frequências verticais e horizontais. A Figura 10 ilustra uma comparação entre domínio de frequências e domínio espacial. Nessa ilustração, os coeficientes estão normalizados de forma que o maior valor esteja em branco.

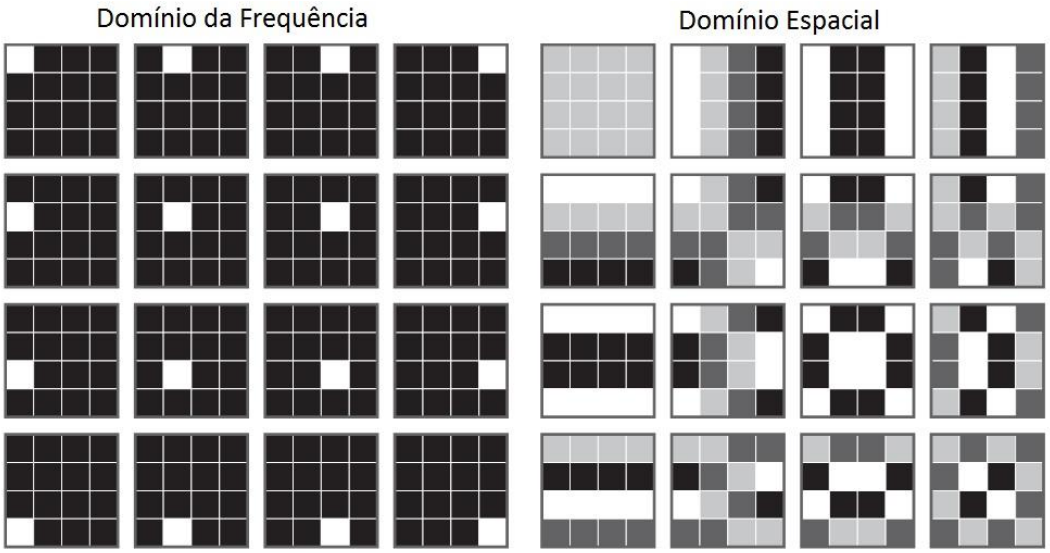

Figura 10 - Domínio de frequências e domínio espacial, adaptada de [1].

A aplicação da etapa de transformação tem como finalidade facilitar a compressão. Há maior vantagem na codificação de imagens no domínio da frequência, uma vez que a maior parte das imagens contem principalmente informações de baixa frequência (região superior esquerda da saída DCT), sendo as informações de alta frequência nulas, e assim culminando na utilização de menor quantidade de bits para codificação. Outro ponto é a limitação do sistema visual humano: uma vez que há maior sensibilidade a informações de baixa frequência, a precisão de informações de alta frequência pode ser diminuída e, assim, diminui-se também a quantidade de informação a ser codificada.

Após a etapa de transformação, a informação passa pela etapa de quantização. Na quantização, os coeficientes de domínio de frequência provenientes da DCT sofrem uma transformação de forma a diminuir o número de bits necessários para codificar tal informação. Os coeficientes são divididos por um valor fixo, ou segundo tabela pré-

definida, e arredondados para um valor inteiro conforme uma função de transferência. Porém, é um processo que acarreta perda de precisão e qualidade, uma vez utiliza aproximação e nem todas as informações poderão ser recuperadas na decodificação. Um exemplo de função de transferência utilizada na quantização é ilustrado na Figura 11.

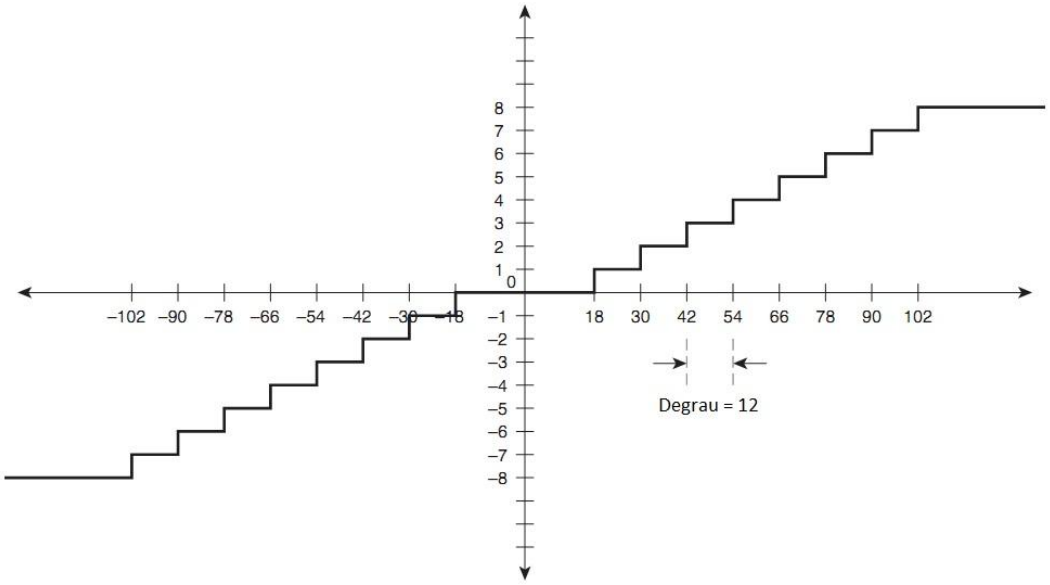

Figura 11 - Quantização - Função de transferência, adaptada de [1].

Geralmente, essas funções são aplicadas com degraus fixos, e quanto maior esse valor, pior a precisão. Porém, existem soluções que envolvem degraus de tamanhos variáveis, e assim é possível utilizar maior precisão para regiões com maior diversidade de valores.

Codificação

Existem diversos algoritmos utilizados para codificação e compressão das informações [5]. Com o aumento de qualidade de imagens, vídeo e áudio, a quantidade de informações a ser transmitida também aumenta, e a compressão tem um papel bastante importante para viabilizar essa transmissão. As informações são comprimidas de forma que as informações caibam nas taxas de transferência de bits disponíveis e ainda possibilitem experiências de qualidade. Idealmente, deve prover a utilização de poucos bits e possibilitar a reconstrução da informação com alta fidelidade.

A codificação e compressão das informações se dão basicamente a partir de informações redundantes. São aplicados algoritmos de forma a eliminar a informação redundante antes da transmissão, de forma que a informação total a ser transferida seja menor, e é comum a utilização de mais de uma técnica de compressão na mesma informação.

Uma técnica de compressão é a *Variable Lenght Coding* (VLC). Eficiente para dados que se repetem em diferentes distribuições probabilísticas, esse algoritmo utiliza cadeias de bit menores para codificar valores que aparecem em maior frequência, favorecendo a diminuição do total de informação codificada. Outra forma compreende a codificação de um valor seguido do número de repetições consecutivas do mesmo valor. Essa técnica tem o nome de *Run Lenght Coding*, e é interessante quando há grande repetição de valores, como silêncio ou zumbido uniforme. Vale ainda ser citada uma outra técnica referenciada como *Arithmetic Coding*, que possibilita a codificação de uma sequência de valores em apenas um valor em ponto flutuante. Uma vez que utiliza ponto flutuante, essa técnica é limitada pela precisão da máquina que se utiliza para tal codificação/decodificação.

Os métodos acima citados são utilizados para codificação da informação a partir dela mesma, ou seja, não utilizam nenhum tipo de predição. Serão então abordadas técnicas utilizadas na codificação de informações com predição e estimativas.

Entende-se por redundância temporal a redundância de informações existente entre quadros adjacentes. Existem então técnicas para tratamento dessa redundância temporal, através da predição de quadros sucessivos e codificação apenas da diferença entre eles.

Há diversos tipos de quadros e é crucial o entendimento dos principais tipos, que são: quadros I, quadros P e quadros B.

Os quadros I, já mencionados anteriormente, são quadros independentes, codificados baseados apenas na informação do próprio quadro, sem referencia a qualquer outro quadro. Os *Predicted Frames*, ou quadros P, são os quadros originados a partir da informação de quadros I ou mesmo P anteriores. Já os *Bidirectional Frames* são codificados a partir de informações de quadro P ou I anterior e posterior.

Utilizando estimativa e compensação de movimento na codificação, os quadros P atingem taxas de compressão maiores que quadros I, porém propagam erros, o que limita o número desses quadros entre quadros I. Os quadros B atingem taxas de

compressão ainda maiores, porém envolvem o processamento de três quadros (P ou I anterior, quadro atual e P ou I posterior) e ainda geram um atraso na codificação/decodificação, tempo esse referente à espera do quadro P ou I posterior. Outra característica interessante, é que os quadros B não propagam erros porque, diferentemente de quadros P, não são referência para codificação de nenhum outro quadro, e assim a perda de um quadro B não gera propagação de erros. A Figura 12 ilustra uma sequência de quadros codificados.

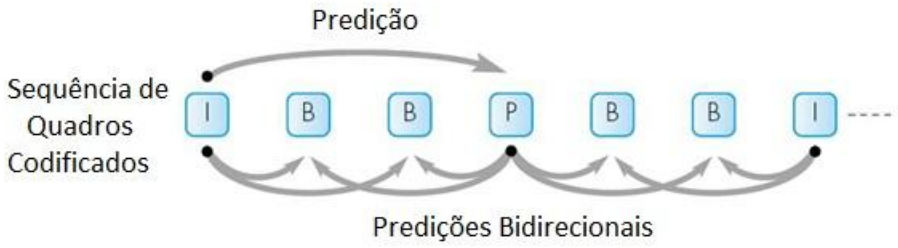

Figura 12 - Sequência de quadros

Os processos de estimativa e compensação de movimento são de grande importância no processo de codificação de quadros com predição, sendo a estimativa o processo que exige maior poder computacional em todo esse tipo de codificação. A ideia principal do processo é encontrar regiões em um quadro que estejam também presentes no quadro seguinte.

Numa visão geral, busca-se no quadro de referência uma região que melhor se assemelhe à região do quadro atual e, então, essa região candidata torna-se previsor do bloco do quadro atual, sendo subtraída do quadro atual para formar um resíduo. Esse resíduo é então codificado juntamente com o *offset* entre o bloco atual e a posição da região candidata.

A utilização de vídeo de alta qualidade tem à frente uma diversidade de barreiras. Limitações de largura de banda da rede e até mesmo de recursos do próprio dispositivo podem afetar a experiência. Com a visibilidade de suprir esses desafios, vale destacar a padronização *Scalable Video Coding* (SVC), que permite a codificação de forma escalável. A ideia principal dessa técnica é oferecer vídeo em varias camadas, sendo uma delas base e outras adicionais que permitirão um refinamento progressivo.

Existem três tipos de escalabilidade: sinal ruído, espacial e temporal. A escalabilidade sinal ruído (SNR) utiliza uma camada base de baixa qualidade, e outras camadas adicionais que correspondem a correções da imagem. Já a escalabilidade espacial fornece como camada base uma imagem de tamanho pequeno, e camadas adicionais que contribuem com aumento de resolução. Por fim, a escalabilidade espacial fornece em camada base uma taxa de quadros baixa e camadas de melhorias que fornecem um aumento dessa taxa. É ainda possível combinar esses tipos de escalabilidade num mesmo *bitstream*. A Figura 13 ilustra um *bitstream* com camadas de aumento de taxa de quadros e de resolução.

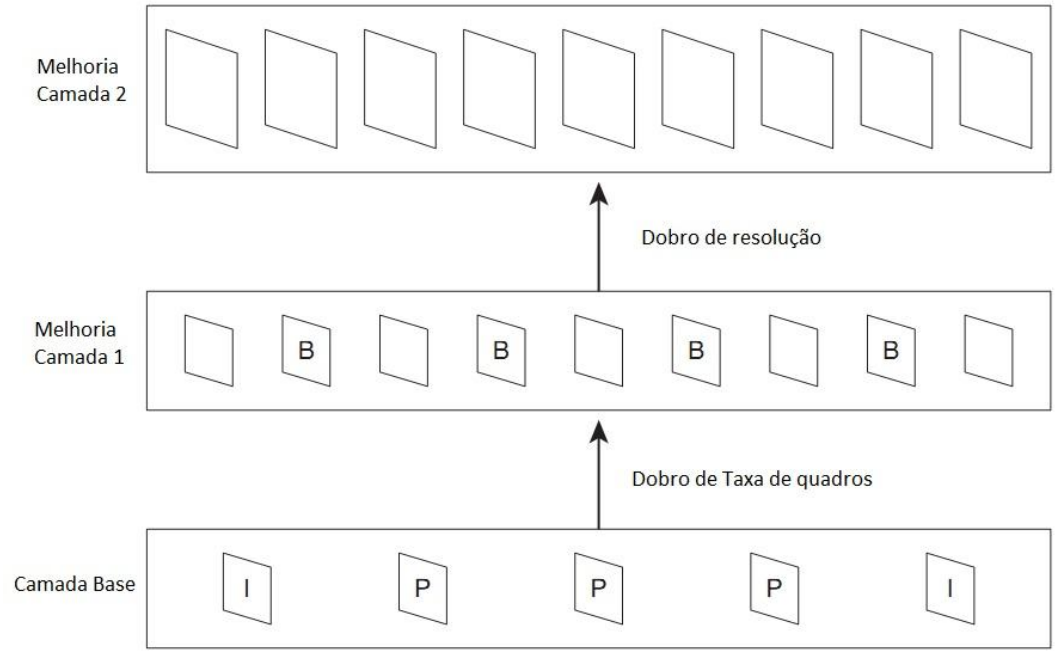

Figura 13 - Codificação escalável, adaptada de [1].

O SVC permite que dispositivos de diferentes capacidades, em uma mesma vídeo conferência, possam ter uma experiência adequada a seus recursos. Um dispositivo mais avançado pode processar uma camada base e todas as outras camadas de refinamento, assim obtendo o máximo da qualidade. Por outro lado, um dispositivo mais limitado pode processar apenas uma camada base com algum refinamento ou não, de acordo com suas limitações.

Capítulo 4 - Transporte de mídia

Em sistemas de vídeo conferência, o transporte da informação é feito através da utilização de pacotes segundo protocolo *Real-time Transport Protocol* (RTP) [6]. Nesse capítulo será abordada em maior detalhe a estrutura dessa padronização.

Todas as informações de voz e vídeo são transportadas em pacotes RTP, transporte esse feito através de *User Datagram Protocol* (UDP). Uma vez que utiliza conexões UDP, o recebimento dos pacotes não é garantido, já que esse tipo de conexão é dita como não confiável, ou seja, não há confirmação de recebimento de pacotes. Por outro lado, a utilização de conexões UDP é interessante tratando-se de *streaming*, como é o caso de vídeo conferências, já que trata-se de uma atividade em tempo real, em que atrasos atrapalham a experiência, e a utilização de conexões confiáveis como *Transmission Control Protocol* (TCP) poderiam gerar atrasos indesejados.

É necessária uma conexão RTP separada em uma porta UDP para cada *stream* de uma vídeo conferência. Os pacotes RTP são formados por um cabeçalho fixo, uma possível extensão do cabeçalho e os dados em si.

O cabeçalho é formado, no primeiro octeto, por campos que indicam versão do RTP utilizado, presença ou não de cabeçalho de extensão e indicação de presença de lista *Contributing Source Identifiers* (CSRC). Esses campos são indicados pelas letras V, X e CC da Figura 14, a qual ilustra o cabeçalho por completo.

Figura 14 - Estrutura de pacote RTP, adaptada de [1].

É possível ainda observar na estrutura do cabeçalho a presença, ainda no primeiro octeto, dos campos P e M. O campo P indica a ocorrência de *padding*, enquanto o campo M (*marker*) é um elemento que indica um pacote como sendo o último pacote de um quadro. Esses são os elementos que compõem o primeiro octeto do cabeçalho.

Em seguida, há o campo *Payload Type* (PT), que especifica o tipo de *codec* e taxa de amostragem dos dados contidos naquele pacote, e o campo Número de Sequência, o qual indica um valor que representa a sequência com que os pacotes foram transmitidos. Aparecem então os campos *Time Stamp*, *Synchronization Source* (SSRC) *Identifier* e CSRC.

Numa vídeo conferência, cada *stream* tem uma identificação única, o que corresponde ao SSRC. Caso ocorra de dispositivo e servidor apresentarem mesmo SSRC, a conexão é fechada e reestabelecida, de forma a atribuir SSRCs únicos aos elementos da vídeo conferência. Como mencionado em capítulos anteriores, o *mixer* de áudio seleciona *streams* a serem somados para formarem um *stream* de saída. O campo CSRC então é responsável por apresentar a lista das *streams* que compõem os dados daquele pacote, e faz isso através de uma lista de SSRCs correspondentes a cada um desses *streams*.

O último campo conforme ilustrado na Figura 15 é o *Payload*, que corresponde aos dados em si, e contem geralmente informações de um quadro ou parte dele, dificilmente apresentando informação de mais de um quadro no mesmo pacote.

Atrelado ao protocolo RTP, existe o protocolo RTP *Control Protocol* (RTCP). O protocolo RTCP é responsável por gerar estatísticas e controle sobre um fluxo de dados RTP, e é importante que a banda utilizada na transmissão de pacotes RTCP não atrapalhe a transmissão de pacotes RTP.

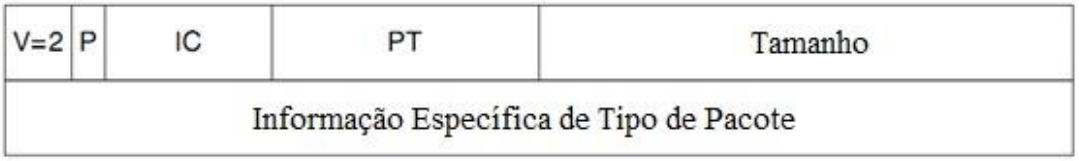

Figura 15 - Pacote RTCP, adaptada de [1].

A Figura 15 exibe a estrutura de um pacote RTCP. Conforme ilustrado, o pacote é formado por um cabeçalho fixo e informação referente a um formato específico do pacote, sendo possíveis cinco tipos de formatos de pacote RTCP: *Sender Report* (SR), *Receiver Report* (RR), *Source Description* (SDES), *Membership Termination* (BYE) e *Application-specific functions* (APP).

De forma análoga ao pacote RTP, o pacote RTCP apresenta também os campos de versão (V) e *padding* (P). Além desses, como é possível observar na Figura 15, apresenta também os campos *Packet Type* e *Lenght*, que indicam respectivamente o tipo do pacote RTCP e o tamanho do pacote.

Cada um dos cinco tipos de pacote RTCP possíveis citados acima tem uma finalidade específica. No caso dos pacotes SR, a finalidade é prover estatísticas da transmissão dos pacotes RTP enviados em um intervalo, e inclui também *timestamps*, importantes na sincronização de vários *streams* (sincronização de áudio e vídeo, por exemplo). Já os pacotes RTCP do tipo RR, são responsáveis por fornecer informação da qualidade do que é recebido, para que assim possam ocorrer ajustes de condições na transmissão de forma a adequar a comunicação.

Outro tipo de pacote RTCP, o tipo SDES, tem o papel de informar dados e outros maiores detalhes do participante, como nome, endereço, entre outros. É através desse tipo de pacote que é enviado o CNAME, que corresponde a um nome que identifica de forma única cada participante da sessão. Como a própria sigla de representação desse tipo de pacote RTCP, o tipo BYE é enviado pelo participante ao deixar a sessão, de forma a indicar a sua saída. Por fim, existe ainda o tipo APP, dedicado ao desenvolvimento de extensões específicas.

Na arquitetura do sistema de vídeo conferência, a informação percorre diversos dispositivos e sofre alterações diferentes em cada um deles. A partir das finalidades de diferentes dispositivos discutidas no capítulo 2, é de se imaginar o que ocorre nessas passagens.

Dois componentes do sistema já discutidos são *transcoders* e *transraters*, que tem a finalidade de transformar formatos de mídia e taxas de bit, respectivamente. Ao passar por esses elementos, os pacotes RTP têm seus cabeçalhos reescritos, porém não há alteração no SSRC, uma vez que o *stream* continua sendo o mesmo, conforme Figura 16.

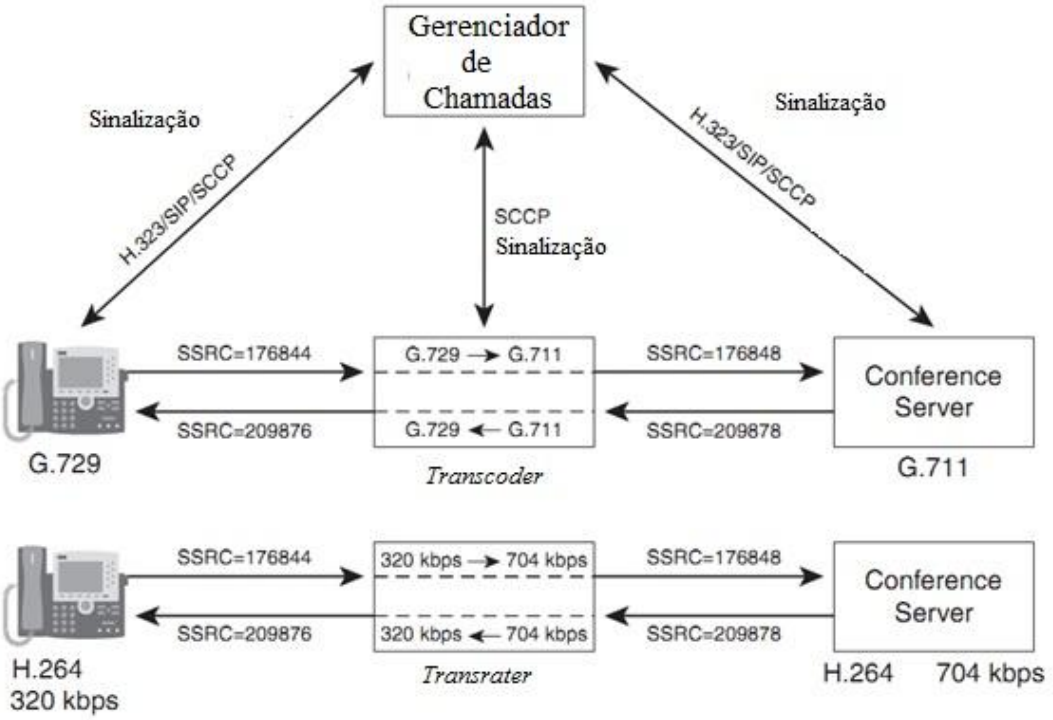

Figura 16 - Alterações de informação em processos de *transrating* e *transcoding*, adaptada de [1].

Não é possível afirmar o mesmo para os *mixers* de áudio e vídeo, uma vez que nesses elementos os *streams* de entrada não são os mesmos *streams* de saída. Na verdade, o *stream* de saída é uma composição dos *streams* de entrada, conforme discutido em capítulos anteriores. Dessa forma, há criação de novos cabeçalhos RTP, uma vez que são criados novos *streams* de saída. Isso implica em criação de novo SSRC de identificação do novo *stream* de saída e inclusão da lista CSRC, composta dos SSRCs dos *streams* de entrada que compõem a saída. A Figura 17 ilustra o processo mencionado para um áudio *mixer*.

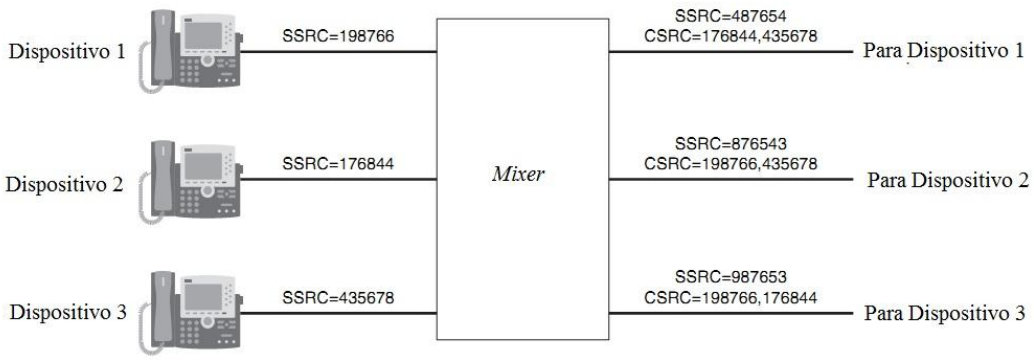

Figura 17 - Alteração da informação em *mixer*, adaptada de [1].

Capítulo 5 - Padrões de sinalização

A sinalização para estabelecimento de chamadas e controle e negociação de mídia em sistemas de vídeo conferência é feita através da utilização de dois principais padrões: *Session Initiation Protocol* (SIP) e H.323. Maiores detalhes dessas duas recomendações serão abordados nas próximas sessões.

Session Initiation Protocol - SIP

O protocolo de sinalização SIP [7] é utilizado no estabelecimento de sessões multimídia, e uma rede SIP é formada de quatro elementos básicos, sendo eles *User Agent*, *Proxy Server*, *Redirect Server* e *Registrar*.

Na classificação de *User Agent* encontram-se os dispositivos de áudio e vídeo bem como os servidores de controle de chamadas. Cada um desses elementos possui uma instância cliente, denominada *User Agent Client* (UAC), e uma instância servidor, denominada *User Agent Server* (UAS), e essas instâncias são responsáveis por iniciar requisições e respostas, respectivamente.

Os elementos *Proxy Servers* são responsáveis pelo encaminhamento de mensagens, bem como implementam algumas funcionalidades como autenticação, segurança e autorização. Esses elementos estão no caminho entre origem e destino, e atuam de forma que ao receberem uma mensagem, determinam qual o destino dessa mensagem, seja ele um *User Agent* ou outro *Proxy Server*, e a encaminham, de forma que ela seja entregue ao elemento adequado. A Figura 18 ilustra a ação de um *Proxy Server* numa rede SIP, e nota-se o papel que tem direcionando as mensagens aos seus destinos.

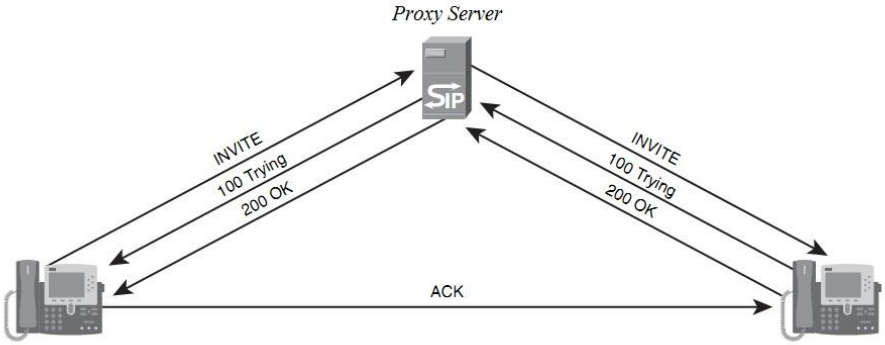

Figura 18 - *Proxy Server*, adaptada de [1].

Outro elemento, o *Redirect Server*, tem o papel de gerar respostas de redirecionamento para requisições feitas a ele, indicando endereços alternativos para o elemento iniciador da requisição.

Por fim, *Registrar* é o elemento responsável por processar as requisições de registro de UAC, e faz isso associando endereço IP desses elementos aos seus *Uniform Resource Identifier* (URI), que corresponde a uma identificação única de cada um dos elementos que compõem uma rede SIP.

É importante serem salientados dois conceitos: transação e diálogo. Transação corresponde a todas as mensagens, requisições e respostas, referentes a uma primeira requisição. Já por diálogo, entende-se o relacionamento entre dois elementos que se comunicam. A Figura 19 expõe esses dois conceitos, e é interessante observar que as mensagens ACK não fazem parte de uma transação.

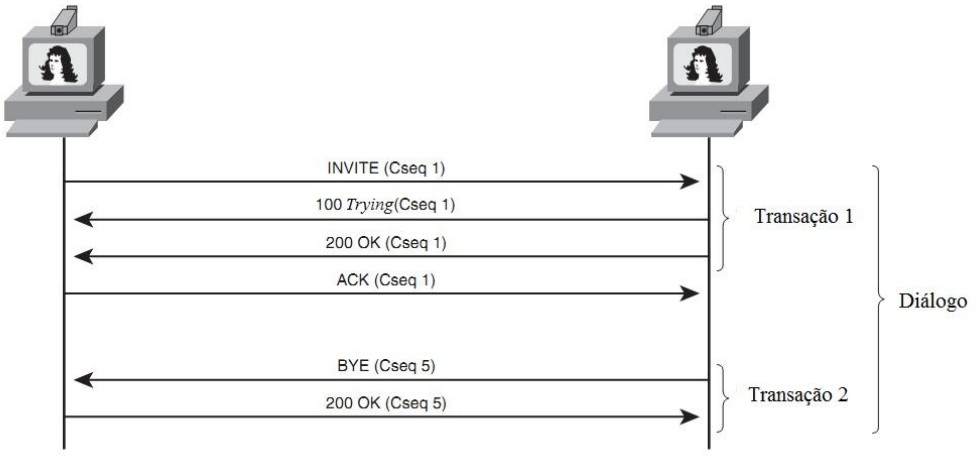

Figura 19 - Transações e diálogo, adaptada de [1].

Ainda na Figura 19, podem ser observadas algumas mensagens da sinalização SIP. Algumas das mensagens de requisições definidas na sinalização SIP estão contempladas na Tabela 1, juntamente com a funcionalidade de cada uma.

Tabela 1 - Mensagens de requisição SIP

Requisição	Funcionalidade
INVITE	Convite à dispositivo para chamada
BYE	Término de diálogo
REGISTER	Solicitação de registro (para elemento Registrar)
INFO	Troca de informações durante chamada
NOTIFY	Notificação de evento

As mensagens de requisição são formadas por linha de requisição, cabeçalho e um corpo de mensagem opcional. Na linha de requisição está especificado o endereço destino da mensagem e o método SIP. Já o cabeçalho, contem diversas informações, dentre elas os endereços de origem e destino da mensagem, identificação única da chamada SIP (*Call*-ID) e *Command Sequence* (Cseq), que corresponde a um identificador que permite relacionar requisições e respostas de uma mesma transação.

De maneira similar, as mensagens de reposta são compostas por uma linha de *status*, a qual contem um código que identifica a situação da mensagem, cabeçalho e um corpo de mensagem opcional. Assim como nas requisições, as mensagens de resposta também contem no cabeçalho o campo Cseq.

A sinalização SIP implementa um método bastante útil para tarifação de chamadas. Quando um dispositivo inicia uma chamada SIP, a mensagem de INVITE percorre o caminho necessário, passando por elementos intermediários (*Proxy Servers*), até atingir o destino. Após isso, todas as mensagens são trocadas diretamente entre os dois dispositivos, e dessa maneira os *Proxy*s não obtêm as informações necessárias sobre a chamada para que possa ser feita a cobrança. A utilização do método *Record Routing* muda esse cenário: com essa funcionalidade, o *Proxy* adiciona um cabeçalho *Record Route* na mensagem com a identificação URI de si mesmo, obrigando que todas as próximas mensagens do diálogo passem por ele.

As mensagens SIP utilizam uma sintaxe para descrição de sessões de mídia denominada *Session Description Protocol* (SDP). As informações SDP são carregadas no corpo das mensagens SIP, e são utilizadas na negociação de parâmetros de mídia de uma sessão.

Conforme mencionado, a negociação de propriedades de mídia entre dispositivos é feita através de SDP, e essa informação pode ser trocada de dois modos: *Early Offer* e *Delayed Offer*. No primeiro, *Early Offer*, a informação SDP é enviada já na mensagem de INVITE do dispositivo que inicia a sessão, sendo então respondida pelo elemento destino. De forma diferente, no modo *Delayed Offer*, o dispositivo que inicia a sessão envia o INVITE sem a informação de mídia SDP, e a oferta de SDP vem como uma resposta do outro dispositivo. O dispositivo iniciador responde então a essa oferta SDP. Esses dois modos são exemplificados na Figura 20.

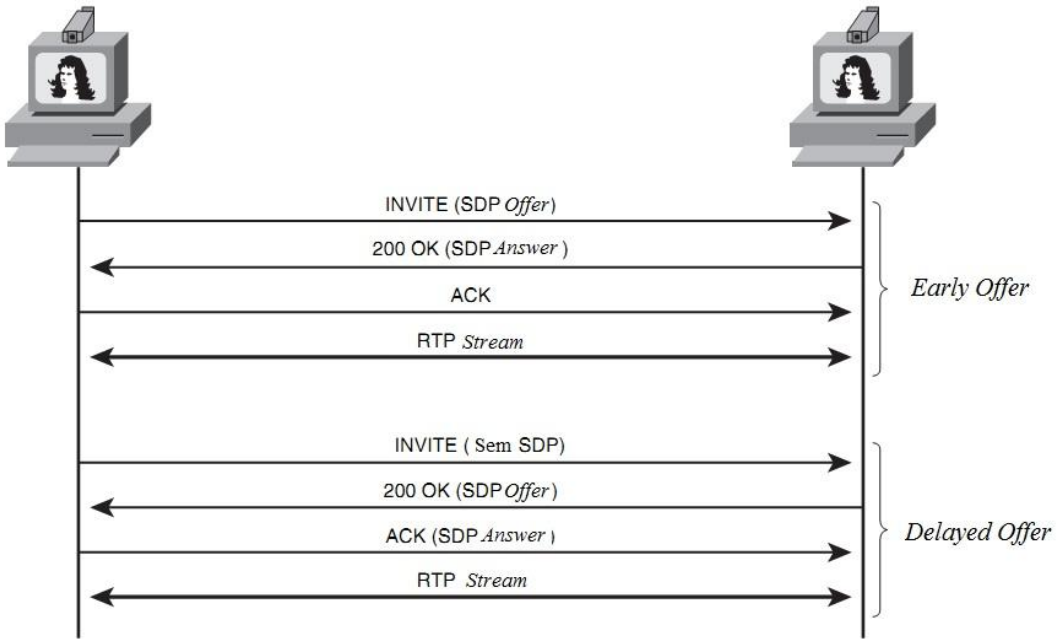

Figura 20 - *Early Offer* e *Delayed Offer*, adaptada de [1].

Uma vez em uma vídeo conferência do tipo *Ad-Hoc*, os dispositivos podem renegociar parâmetros de mídia a ser trocada através de mensagens RE-INVITE. Um dispositivo que inicia uma sessão com apenas o modo de áudio e que, no meio da sessão, decide iniciar também a transmissão de vídeo, envia então uma mensagem de RE-INVITE contendo SDP adequado com parâmetros da nova mídia a ser trocada

(inclusão de vídeo). A esse processo, dá-se o nome de *Escalation*. O processo inverso é denominado *De-Escalation*, e ocorre da mesma maneira. Essa situação é ilustrada na Figura 21.

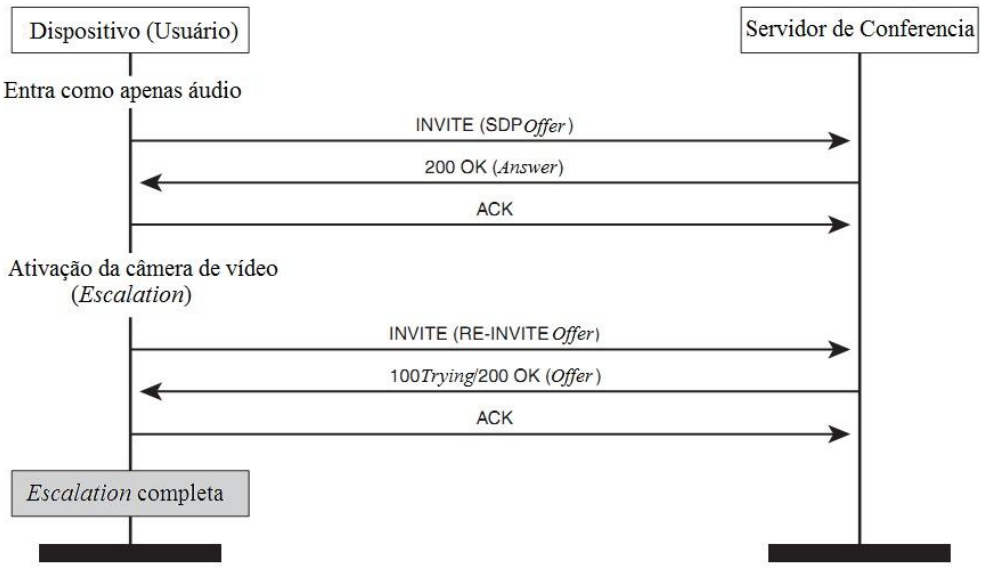

Figura 21 - *Escalation*, adaptada de [1].

O processo de *Escalation* gera um grande desafio aos servidores de vídeo conferência no que se refere à alocação de recursos. Ao iniciar uma sessão *Ad-hoc*, em modo de apenas áudio, o servidor de vídeo conferência não possui a informação de que o dispositivo possui também a capacidade de transmissão de vídeo. Assim, se o dispositivo decide incluir a transmissão de vídeo no meio da sessão, o servidor pode se deparar com uma situação de falta de recursos para suprir essa nova condição. O tratamento a esse tipo de situação pode ser feito através de configurações do sistema de vídeo conferência, de modo a alocar recursos conforme politica definida, ou apenas negar a solicitação caso não haja recurso disponível. Os dispositivos podem deixar uma sessão com o envio de uma mensagem BYE.

A diferença no estabelecimento de chamadas *Ad-Hoc* e *Scheduled* se dá basicamente pela interação com um sistema IVR no último modo. As Figuras 22 e 23 exibem o estabelecimento para cada um dos modos.

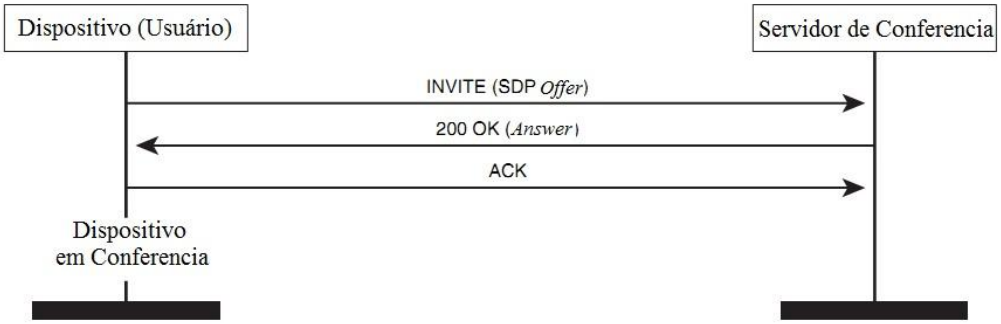

Figura 22 - Estabelecimento de chamada *Ad Hoc - Early Offer*, adaptada de [1].

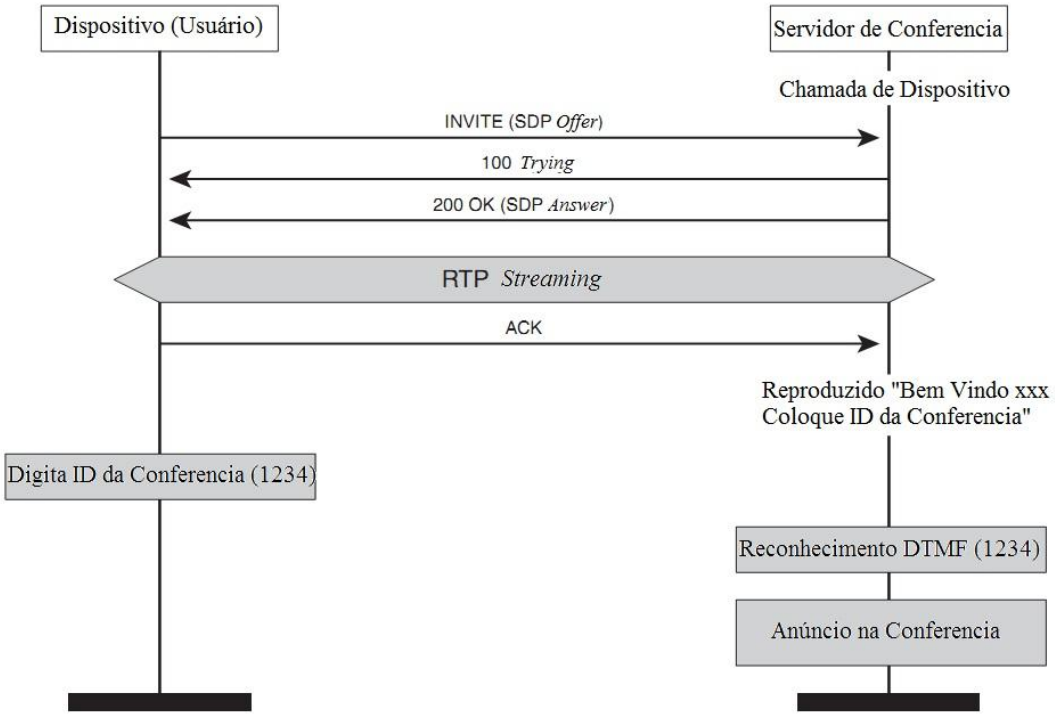

Figura 23 - Estabelecimento de chamada *Scheduled*, adaptada de [1].

No controle de mídia, existem duas requisições importantes, que são *Video Fast Upgrade* (VFU) e *Video Picture Freeze*, ambas geralmente enviadas através de mensagens SIP INFO. Quando um decodificador necessita de um quadro I para continuar a decodificação, envia VFU requisitando esse quadro. O codificador recebe então esse VFU, o interpreta e codifica o próximo quadro como um quadro I e o envia, de forma que o decodificador possa continuar a decodificação. Já a requisição de *Video*

Picture Freeze tem a funcionalidade de paralisar a decodificação. Quando o codificador percebe mudanças que acarretarão em perdas de sincronização, envia a requisição ao decodificador para que paralise o processo de decodificação. O processo de decodificação reinicia quando o codificador envia um sinal de liberação, o qual é gerado quando é enviado o próximo quadro I.

H.323

H.323 é um padrão amplamente empregado na comunicação em vídeo conferências [8]. Assim como no padrão SIP, uma rede H.323 é formada de diversos elementos, dentre eles terminais, *gateways*, *Multipoint control units* (MCU) e *gatekeepers*. Sob a definição de terminais, encontram-se os dispositivos com capacidades de áudio, vídeo e outras funcionalidades, como, por exemplo, os telefones, *desktops* e terminais de vídeo conferência. Os *gateways* são os dispositivos que fazem a interface entre rede IP e rede *Public Switched Telephone Network* (PSTN), tornando transparentes as adaptações necessárias para cada cenário. Já os elementos MCU são responsáveis por permitir vídeo conferências com mais de dois participantes, fornecendo principalmente os serviços de *mixing* já discutidos. Existem ainda os *gatekeepers*, que são elementos responsáveis por fornecer serviços adicionais como controle de acesso e gerenciamento de largura de banda. Uma composição com esses elementos é ilustrada na Figura 24.

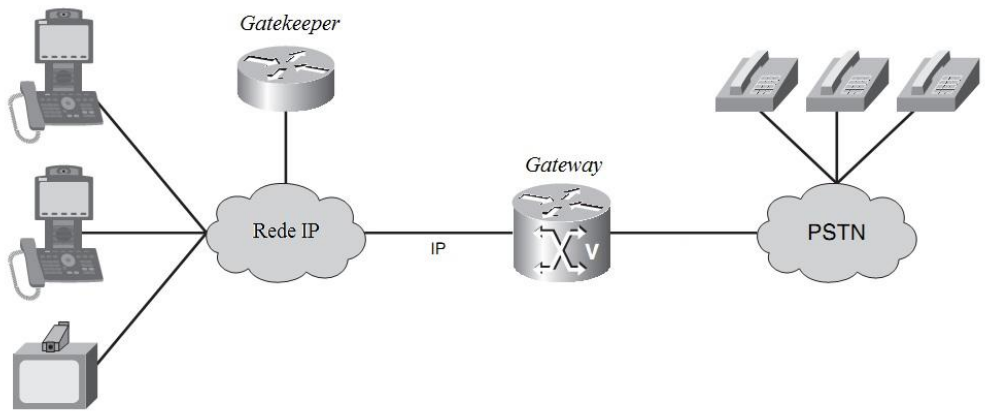

Figura 24 - Elementos de rede - H.323, adaptada de [1].

A especificação H.323 define alguns protocolos a serem utilizados, sendo eles H.225, H.225 RAS e H.245. O primeiro é utilizado na sinalização para o estabelecimento de chamadas [9], enquanto o segundo é utilizado na comunicação entre dispositivos e *gatekeeper*. Por fim, o protocolo H.245 é utilizado no controle e negociação de mídia [10].

A identificação e endereçamento de cada elemento de uma rede H.323 pode ser feita de diversas maneiras. A mais comum é a utilização de dígitos E.164 que correspondem aos números que identificam os telefones na PSTN. Também podem ser utilizados os identificadores H.323 ID e URL ID. O primeiro consiste em basicamente numa *string* e é mais útil localmente, enquanto o segundo consiste numa identificação no formato h323:usuário@nome_do_host e discagem baseada em *web*.

Como mencionado, a sinalização para estabelecimento de chamadas é feita com a utilização do protocolo H.225, o qual utiliza conexão TCP sobre IP para iniciar, estabelecer e terminar chamadas. Além disso, define diversas mensagens, dentre elas *Setup*, *Connect*, *Notify* e *Release Complete*.

As mensagens de *Setup* são utilizadas para iniciar uma chamada e contem um campo indicando se será uma chamada que envolverá apenas áudio ou se também terá vídeo. Se o dispositivo chamado responde à chamada, envia uma mensagem *Connect* que indica que o estabelecimento da conexão esta completo. Uma vez estabelecida a chamada, o intercâmbio de informações durante a chamada é feito através de mensagens *Notify*. Por fim, se um dispositivo deseja terminar a chamada, envia a mensagem *Release Complete*, e a partir de então encerra-se a sinalização H.225 e os recursos alocados podem ser liberados. A Figura 25 exemplifica o estabelecimento de uma chamada com a utilização do protocolo H.225.

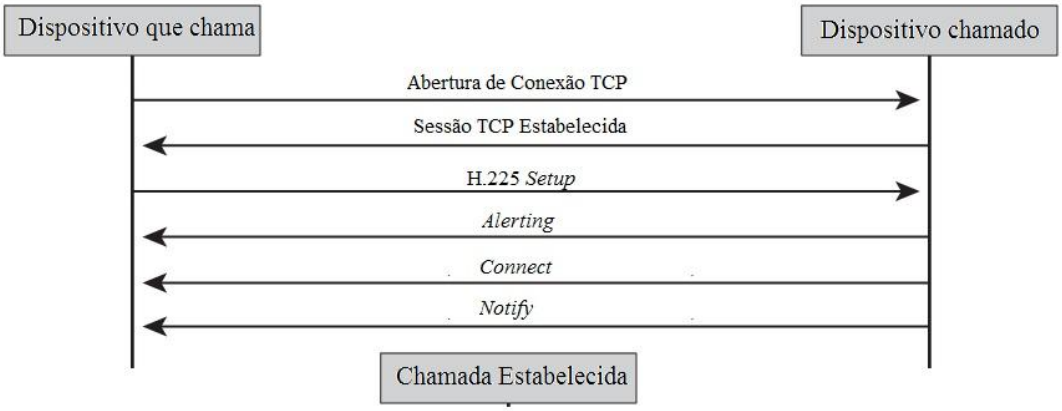

Figura 25 - Estabelecimento de chamada através de H.225, adaptada de [1].

A negociação dos parâmetros de mídia e estabelecimento do canal RTP é feito através do protocolo H.245. As mensagens definidas no padrão H.245 são trocadas através de uma conexão TCP, e dentre elas podem ser citadas *Terminal Capability Set* (TCS), *Master-Slave Determination* (MSD), *Open Logical Channel* (OLC) e *Close Logical Channel* (CLC).

A primeira mensagem trocada é a TCS, na qual estão contidas as informações referentes às capacidades do dispositivo. Cada lado da conexão envia mensagem TCS para que o outro lado possa determinar as propriedades de mídia que utilizará. A mensagem MSD é trocada de forma a estabelecer quais os papéis de cada lado da conexão no gerenciamento do canal lógico e tratamento de conflitos. Para o estabelecimento do canal para transmissão dos dados de mídia, o dispositivo que deseja enviar esses dados utiliza a mensagem de requisição OLC, que contem, dentre outros parâmetros, informação do *codec* a ser utilizado. A mensagem de confirmação referente de OLC, a OLC ACK, contem o endereço IP e porta para os quais os pacotes devem ser transmitidos. Quando deseja fechar o canal, dispositivo envia então mensagem CLC. O fluxo das mensagens na negociação de mídia e estabelecimento de canal para transmissão descrito acima é exibido na Figura 26.

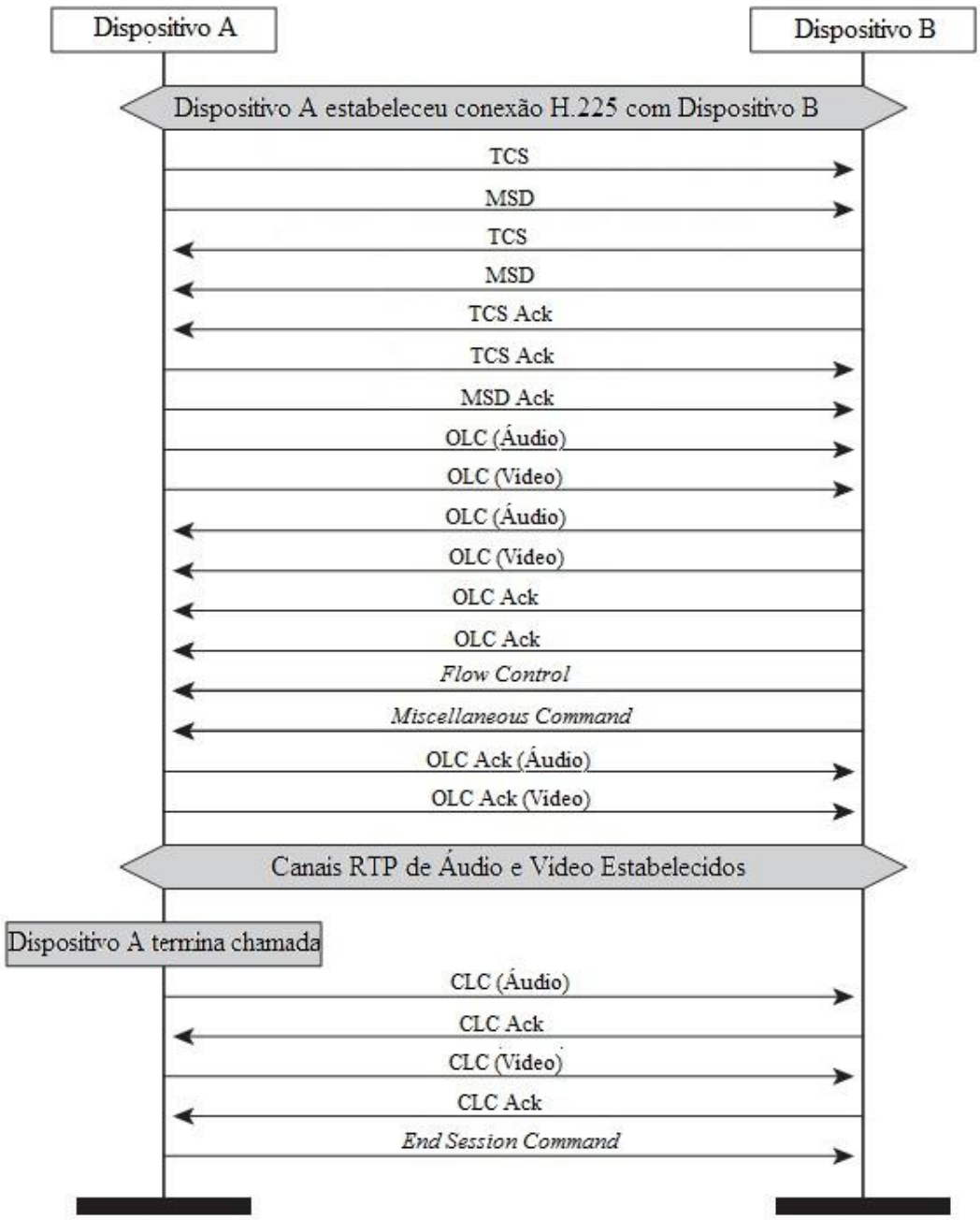

Figura 26 - Negociação de mídia e estabelecimento de canal RTP, adaptada de [1].

Na Figura 26, é possível observar ainda a presença das mensagens *Flow Control* e *Miscellaneous Command*. Essas duas mensagens são específicas para vídeo, e são utilizadas para requisitar ajuste na taxa máxima de transmissão de bits e outras requisições no meio de uma sessão. Ao final, observa-se ainda a presença da mensagem *End Session Command*, que indica o término da sessão e que, então, não serão mais trocadas mensagens H.245.

A comunicação entre *gatekeeper* e outros elementos da rede H.323 é feita através do protocolo RAS, e o canal dessa sinalização é separado dos canais de mídia e estabelecimento de chamada. Essa sinalização pode ser feita de dois modos, denominados *Direct Endpoint Signaling* e *Gatekeeper Routed Call Signaling* (GKRCS). No primeiro modo, a sinalização RAS é feita com o *gatekeeper*, mas as mensagens H.225 e H.245 são trocadas diretamente entre os dispositivos, conforme ilustrado na Figura 27.

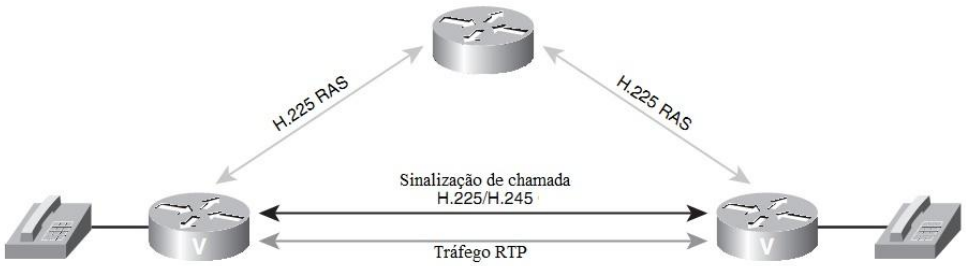

Figura 27 - *Direct Endpoint Signaling*, adaptada de [1].

Já no modo GKRCS, as mensagens H.225 e H.245 são trocadas entre os dispositivos através do *gatekeeper*, conforme Figura 28.

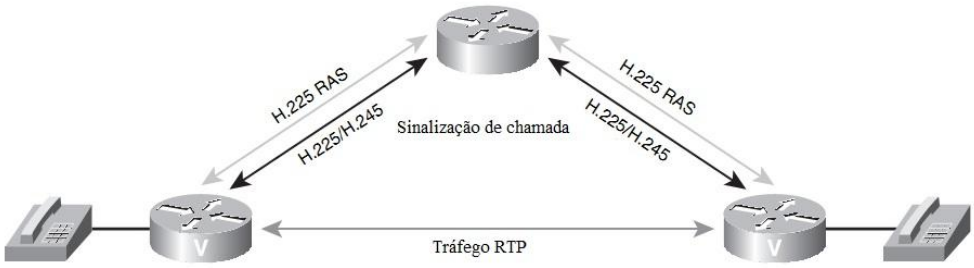

Figura 28 - GKRCS, adaptada de [1].

O protocolo RAS define basicamente três mensagens de requisição: *Registration Request* (RRQ), *Admission Request* (ARQ) e *Disengage Request* (DRQ). A mensagem RRQ é enviada por um dispositivo que deseja se registrar em um *gatekeeper*, enquanto a mensagem ARQ tem a funcionalidade de requisição de quantidade de banda e, por último, a requisição DRQ é enviada para indicar que dispositivo está deixando a sessão.

Essas mensagens de requisição podem ser respondidas com uma confirmação ou rejeição, exceto a DRQ, que é respondida apenas com confirmação.

Além dessas mensagens, é interessante ser mencionada a *Bandwidth Request* (BRQ), que tem o intuito de requisitar ajustes de largura de banda durante uma chamada, podendo também ser respondida com confirmação ou rejeição. A Tabela 2 contem um resumo dessas requisições e suas respectivas rejeições ou confirmações.

Tabela 2 - Mensagens RAS

Requisição	**Confirmação**	**Rejeição**
Registration Request (RRQ)	*Registration Confirm* (RCF)	*Registration Reject* (RRJ)
Admission Request (ARQ)	*Admission Confirm* (ACF)	*Admission Reject* (ARJ)
Bandwidth Request (BRQ)	*Bandwidth Confirm* (BCF)	*Bandwidth Reject* (BRJ)
Disengage Request (DRQ)	*Disengage Confirm* (DCF)	-

Um exemplo do fluxo de mensagens RAS entre dispositivos e *gatekeeper* é ilustrado na Figura 29.

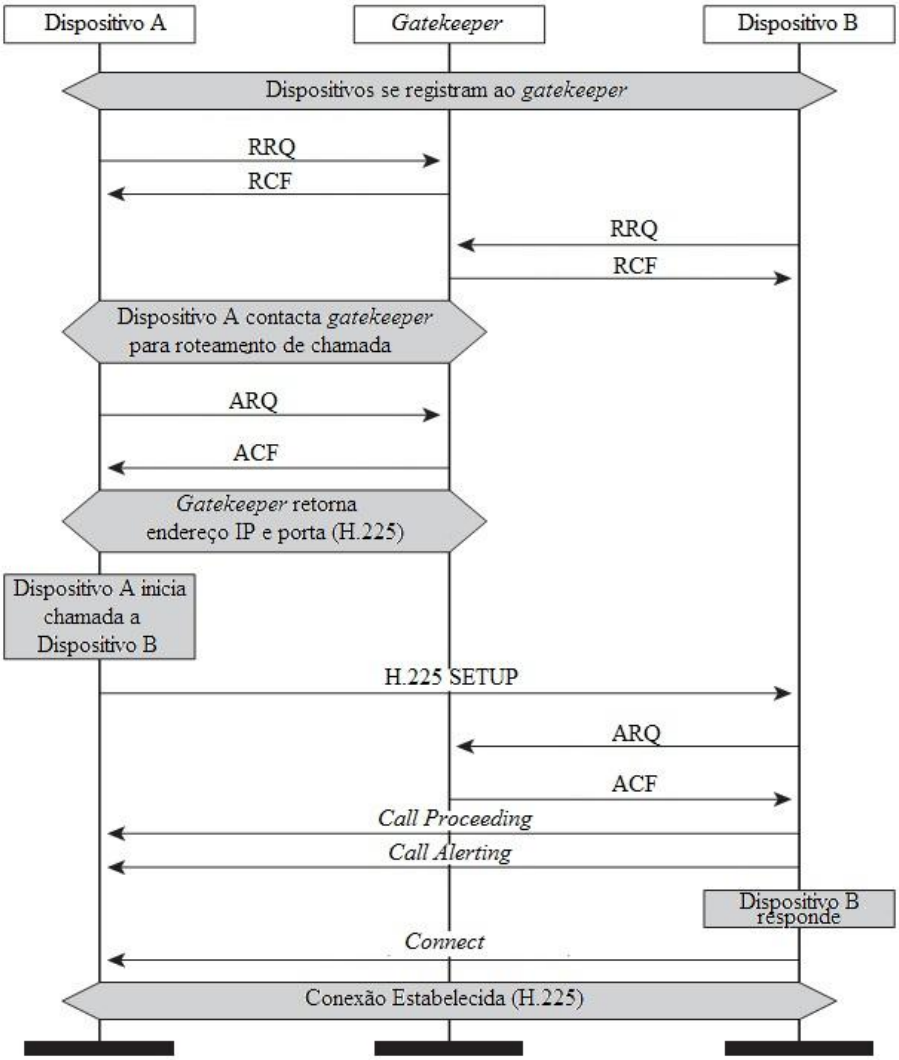

Figura 29 - Comunicação RAS, adaptada de [1].

Capítulo 6 - Sincronização de áudio e vídeo

A sincronização labial é muito importante para experiências de qualidade de vídeo conferências. Esse fundamento é responsável por sincronizar áudio e vídeo decodificados, uma vez que os *streams* desses dois tipos de mídia são separados e precisam ser reproduzidos de forma coerente, de forma que o áudio esteja alinhado com a abertura dos lábios do locutor.

A falta de alinhamento de áudio e vídeo é denominada *skew*, e a realização de operações de sincronização baseia-se muito em limitações de percepção humana. Quanto maior o *skew*, maior o desalinhamento e, consequentemente, maior a percepção humana para essa falta de sincronia. A sensibilidade a *skew* varia de pessoa para pessoa, e de forma geral há maior nível de percepção de *skew* para vídeos de alta resolução e taxa de quadros.

O principal objetivo do mecanismo de sincronização labial é então prover áudio e vídeo com um *skew* próximo de zero, que significa manter na saída o mesmo relacionamento de áudio e vídeo que se tinha na entrada. O *skew* é medido em tempo de apresentação, no dispositivo de saída, e os atrasos contidos em todo o caminho fim a fim, desde a captura até a reprodução, contribuem para esse *skew*. Para alcançar o desafio da sincronização, é necessária a utilização de uma base comum de tempo para os *streams*, e manter as marcas de tempo o mais alinhadas possível na reprodução, adicionando atrasos onde necessário para atingir esse objetivo.

Basicamente, os atrasos se acumulam no decorrer do caminho fim a fim. De uma forma geral, tem-se o atraso no transmissor, o atraso na rede e o atraso no receptor. A Figura 30 expõe o acúmulo de atrasos no caminho fim a fim da informação.

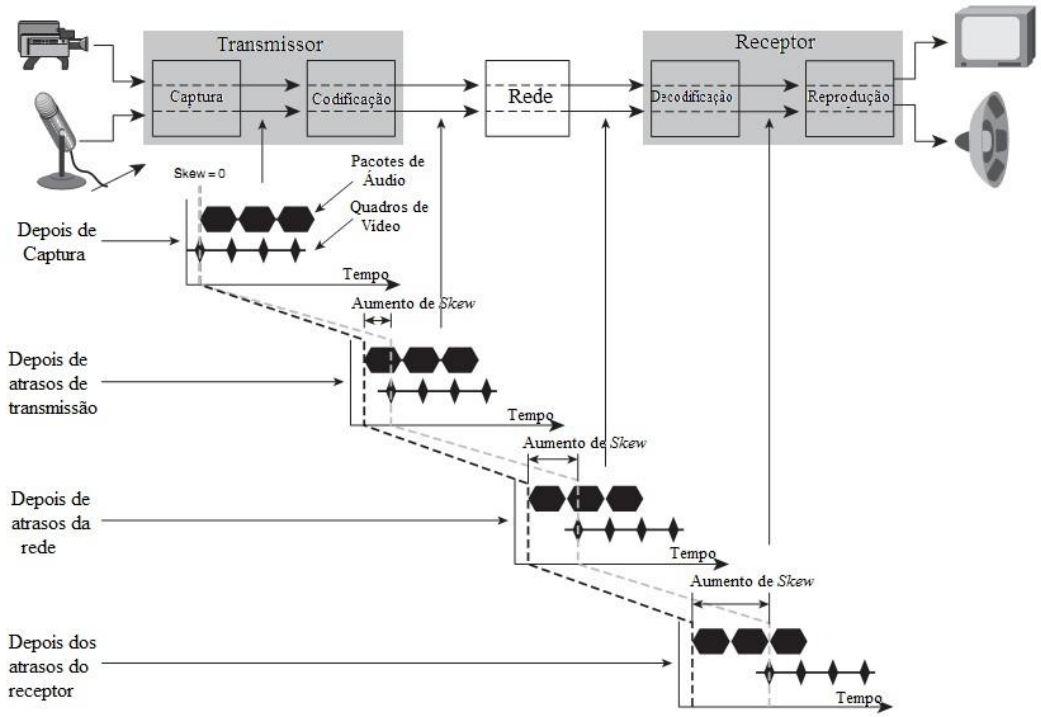

Figura 30 - Cenário com atrasos no caminho da informação, adaptada de [1].

É possível observar que os atrasos nos *streams* de áudio e vídeo são diferentes, e as informações de áudio chegam antes. Isso ocorre geralmente devido ao esforço de processamento que é maior para *streams* de vídeo.

Do lado do transmissor, o processo envolve a captura, digitalização, codificação e empacotamento. Tratando-se de áudio, é comum a presença de codificadores que utilizam quantidades fixas de informações de áudio de cada vez, denominadas *frames* de áudio. A espera pela quantidade de dados de áudio adequada para formar esses *frame*s, bem como o fato de os codificadores processarem um *frame* por vez, acaba por gerar atrasos. O empacotamento RTP também gera atrasos, uma vez que um pacote RTP contem um ou mais *frame*s de áudio, e esse atraso equivale à espera por informação suficiente para que o pacote seja completado adequadamente. No âmbito de captura de vídeo, também há atrasos de codificação. Esses atrasos são variáveis, uma vez que quadros com maior quantidade de movimento exigem maior processamento e acabam por gerar uma maior latência. Assim, ajustes são feitos a todo momento para que se mantenha um limite de atraso suportado, e mecanismos de eliminação de pacotes são adotados para esse objetivo.

Na rede, os atrasos são acumulados devido aos processos de *transrating* e *transcoding*, bem como pelo caminho por diversos elementos até alcançar o receptor.

Já do lado do receptor, as informações são decodificadas e convertidas para sinal analógico e então reproduzidas. Para o caso de *streams* de áudio, existe ainda um *buffer* que acumula os pacotes de forma a reordená-los e superar o obstáculo de pacotes que chegam em intervalos diferentes, obtendo um *stream* uniforme. Uma vez que age como uma espécie de agregador, o *buffer* adiciona atraso no caminho da informação. De forma análoga aos processos de codificação e captura que ocorrem do lado do transmissor, os processos de decodificação e reprodução também geram atrasos, muito devido aos esforços de processamento dos dados. Diferentemente do processo de empacotamento do lado do transmissor, o desempacotamento no lado do receptor não envolve atraso, devido ao fato de não ser necessário aguardar pela chegada de mais informação para iniciar esse processo.

Os componentes de reprodução da informação são divididos em basicamente dois tipos: maleáveis e não-maleáveis. Os maleáveis reproduzem a informação conforme essa é enviada para ele. Já os não-maleáveis, requisitam dados a uma taxa constante para reprodução, como é o caso dos tocadores de áudio.

A abordagem da sincronização labial pelos dispositivos pode ser feita geralmente de duas maneiras, denominadas *Poor Man* e *Common Reference*. Mais simples, a abordagem *Poor Man* baseia-se no momento de chegada dos pacotes para sincronização, uma vez que assume conhecido e constante o atraso gerado em todo o caminho fim a fim da informação. Esse algoritmo assume que os pacotes de áudio e vídeo que chegam ao mesmo tempo estão sincronizados, o que é falho já que os atrasos gerados no tratamento de áudio e vídeo são diferentes, bem como podem haver mecanismos de *Quality of Service* (QoS) que priorizam um ou outro *stream*. Operações de *transcoding* que afetam apenas um dos *streams* também geram diferenças nesses atrasos. Além de tudo, na abordagem *Poor Man* atrasos são inseridos nos *streams* no lado do transmissor e também da rede, o que fere uma característica importante para um mecanismo de sincronização robusto, de que somente o receptor deve adicionar atraso para atingir sincronização labial.

Já a abordagem *Common Reference*, utiliza uma base de tempo comum para atingir sincronização de áudio e vídeo, uma vez que assume variável e desconhecido o atraso acumulado no caminho fim a fim da informação. Os *timestamps* RTP para áudio

e vídeo são diferentes, devido a cada um desses tipos de mídia utilizar um relógio diferente, o que impossibilita a obtenção de sincronização apenas a partir dessas informações. Sendo assim, é necessária a utilização de uma base de tempo comum para os dois, de forma que os *timestamps* RTP possam ser mapeados nessa base comum e seja então possível obter a sincronização. É utilizado então *Network Time Protocol* (NTP) como base comum de tempo, elemento que está presente somente do lado do transmissor. Através então de pacotes RTCP, são enviados os *timestamps* RTP e NTP correspondentes dos *streams*, e do lado do receptor pode então ser feita a sincronização baseada nessas informações.

Capítulo 7 - Segurança dos sistemas de vídeo conferência

Assim como diversos outros cenários de rede, os sistemas de vídeo conferência são alvos de ameaças e precisam estar preparados de maneira adequada para oferecer serviço com segurança. Basicamente, existem alguns fundamentos que são essenciais quando se trata de segurança em redes, sendo eles confidencialidade, autenticação, autorização, identidade, disponibilidade e integridade.

No âmbito de vídeo conferências, ter confidencialidade significa manter as informações de tal maneira que somente transmissor e receptor possam entendê-las, o que pode ser alcançado com utilização de criptografia. Autenticação se relaciona a validação de dados, bem como validação de identidades, enquanto autorização envolve o gerenciamento de permissões para dispositivos ou usuários autenticados. Manter a disponibilidade significa prover os mecanismos adequados para proteção dos recursos e serviços da rede, de forma a evitar problemas de ausência de recursos ou serviços devido a ataques. Por fim, integridade envolve a capacidade analítica de se observar os dados e detectar presença de alterações maliciosas.

Cada um desses fundamentos pode ser afetado por uma diversidade de ataques mal intencionados. Há uma grande gama de ataques à disponibilidade, os denominados ataques *Denial of Service* (DoS). Esses ataques tem o intuito de esgotar os recursos da rede ou de um servidor, e podem ser realizados de diversas maneiras. Um exemplo bastante comum é o *Distributed Denial of Service* (DDoS), no qual diversos dispositivos iniciam grande transmissão de pacotes para um dado nó da rede a ser atacado, tornando o nó então inacessível por excesso de dados na rede. O tratamento a esse tipo de ameaça é feito basicamente inspecionando o tráfego, de forma a identificar e eliminar os pacotes provenientes de ataques, com o uso de equipamentos e *softwares* específicos Ainda tratando-se de DoS, ataques podem ter por objetivo o esgotamento de recursos de um servidor. Nesse tipo de ataque, são enviadas requisições ao servidor, que responde alocando recursos para a conexão, mas nunca obtém uma resposta. Esse exemplo pode ser melhor entendido pela Figura 31.

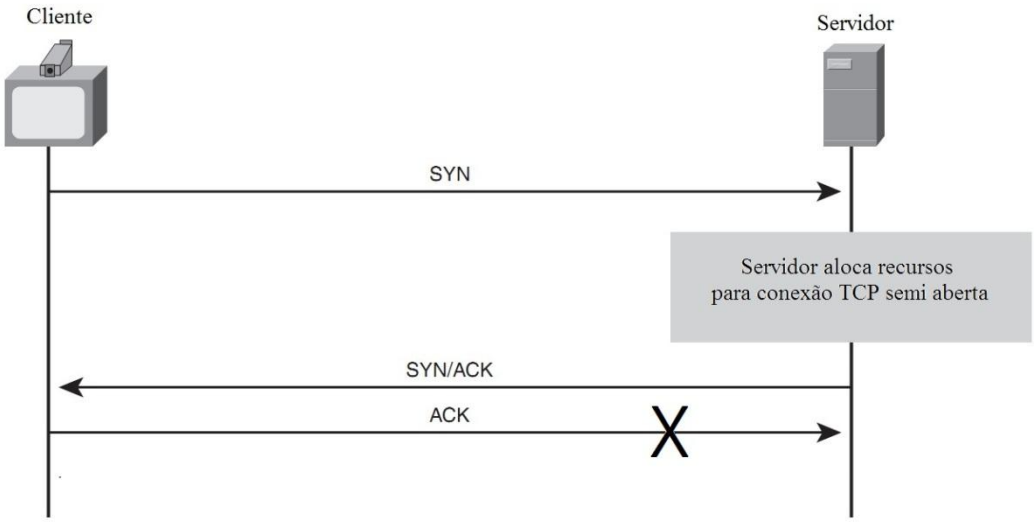

Figura 31 - Ataque DoS, adaptada de [1].

Quando recebe a mensagem de requisição SYN, o servidor aloca recursos e retorna um SYN/ACK e espera por um ACK por um determinado tempo (*timeout*). Acontece que, em casos de ataque, esse ACK nunca é enviado, e os recursos ficam então alocados de maneira inútil ate atingir o *timeout* e, dessa maneira, essas requisições SYN mal intencionadas acabam por esgotar os recursos do servidor. O tratamento a esse tipo de ataque pode ser feito com a inclusão de um *firewall* entre cliente e servidor, conforme ilustrado na Figura 32. Nesse cenário, a troca de mensagens SYN, SYN/ACK e ACK é feita diretamente com o *firewall*, e só então após a recepção do ACK elas são repassadas ao servidor. Desse modo, os recursos do servidor estarão protegidos, de forma que só serão alocados após uma validação de que a conexão não se trata de um ataque.

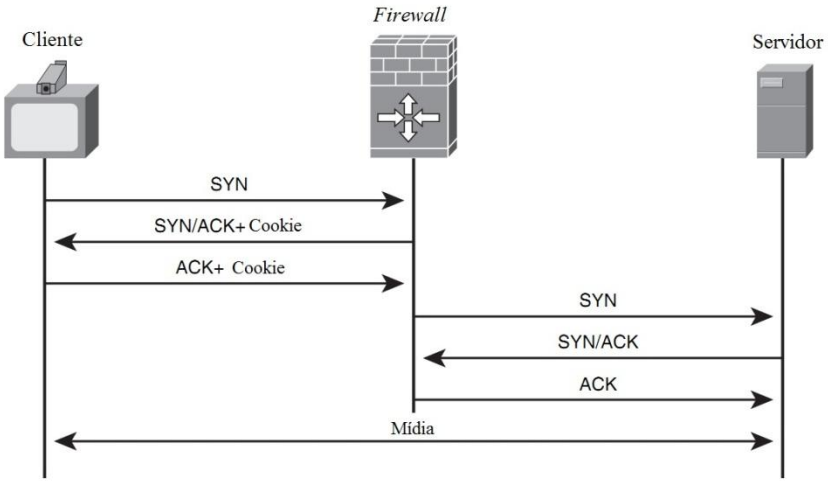

Figura 32 - *Firewall* para evitar DoS, adaptada de [1].

Outro ataque que também envolve o esgotamento de recursos é o *Replay*, no qual uma instância mal intencionada intercepta o tráfego entre dispositivos e guarda os pacotes, então reenvia esses pacotes a um dos dispositivos de forma a esgotar seus recursos de processamento. A utilização de autenticação evita esse tipo de ataque.

Agora tratando-se de autenticação e identidade, uma forma de ataque um tanto conhecida é a *Man-in-the-Middle* (MitM). Nesse tipo de ataque, um dispositivo mal intencionado é inserido no meio do caminho entre dois dispositivos que conversam. Agindo de forma transparente, esse dispositivo finge, para cada dispositivo bem intencionado, ser o outro lado da conversa. Esse tipo de ataque é ilustrado na Figura 33.

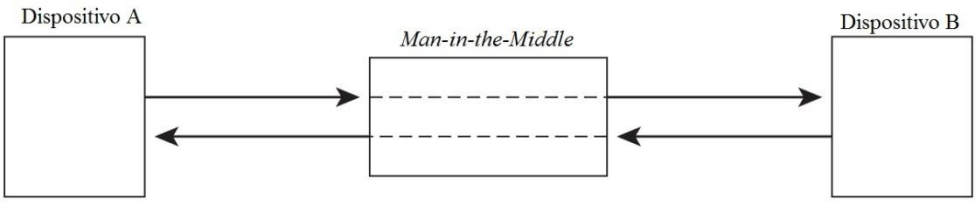

Figura 33 - MitM, adaptada de [1].

Uma vez que age como se fosse o outro lado da conversa, esse dispositivo mal intencionado pode ter acesso a informações que deveriam ser confidenciais a princípio. A utilização de mecanismos de autenticação em cada mensagem protege contra esse tipo de ataque.

É de extrema importância também prover segurança nas camadas de infraestrutura da rede. Dentre os ataques dessas camadas mais baixas, estão os ataques de camada 2, que tem como alvos os principais elementos dessa camada que são os *switch*es. Esses ataques são bastante propagáveis, uma vez que as camadas mais altas são incapazes de identificar o problema, e usualmente é necessário um acesso interno à rede para iniciação desses ataques. Como exemplo, pode ser atacada a tabela de endereçamento de um *switch*, de modo que pacotes que deveriam ser entregues a um único destinatário sejam enviados por meio de *broadcasting* a todas as portas do *switch*, e assim possibilitando acesso a informações que deveriam ser protegidas.

A proteção dos terminais de vídeo conferência também é importante, o que pode ser alcançado com a utilização de escaneadores de vírus e outros *malwares*, utilização de *firmwares* com características de segurança e utilização de *Host-based Intrusion Prevention System* (HIPS). Um cenário com aplicação de diversas formas de segurança é exibido na Figura 34.

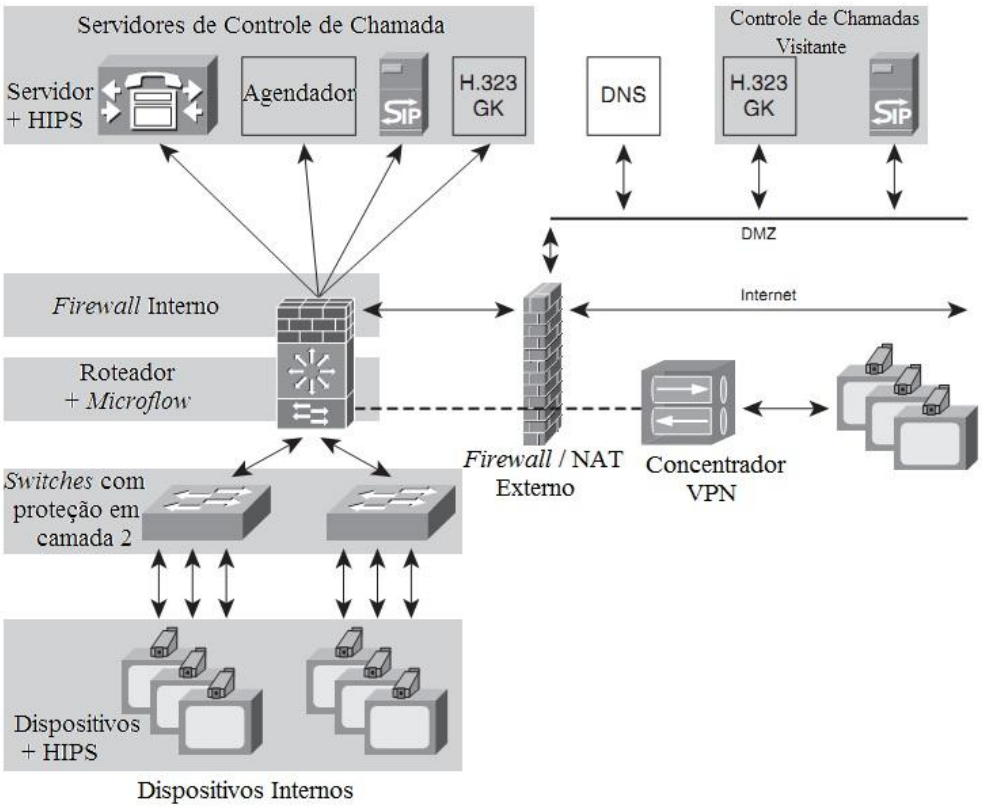

Figura 34 - Cenário com mecanismos de segurança, adaptada de [1].

Nota-se na Figura 34 a presença de servidores e dispositivos com utilização de HIPS, *switch*es com proteção de camada 2, *firewall* interno e externo e concentrador *Virtual Private Network* (VPN). Esse último mecanismo provê túneis seguros para profissionais acessarem uma rede remotamente, bem como na conexão de filiais a suas matrizes, utilizando basicamente fundamentos de autenticação e autorização nessa conexão.

É notável ainda a presença de *Network Address Translation* (NAT) implementado juntamente ao *firewall* externo na Figura 35. Esse é um mecanismo muito interessante que permite esconder a topologia interna da rede, uma vez que provê a tradução de endereços IP e portas privados em endereços IP e portas públicos, alterando essas informações nos pacotes, antes de inserir pacotes provenientes da rede interna na externa. A função NAT [11] pode ser classificada de acordo com suas características de mapeamento e de filtro. Para mapeamento, o mecanismo NAT pode ser classificado como dependente ou independente do dispositivo. No primeiro tipo, faz-se a tradução de diferentes endereços IP e portas para cada destinatário de uma mensagem, enquanto o segundo provê uma mesma tradução para todos os destinatários. Esses dois modos são ilustrados nas Figuras 35 e 36, respectivamente.

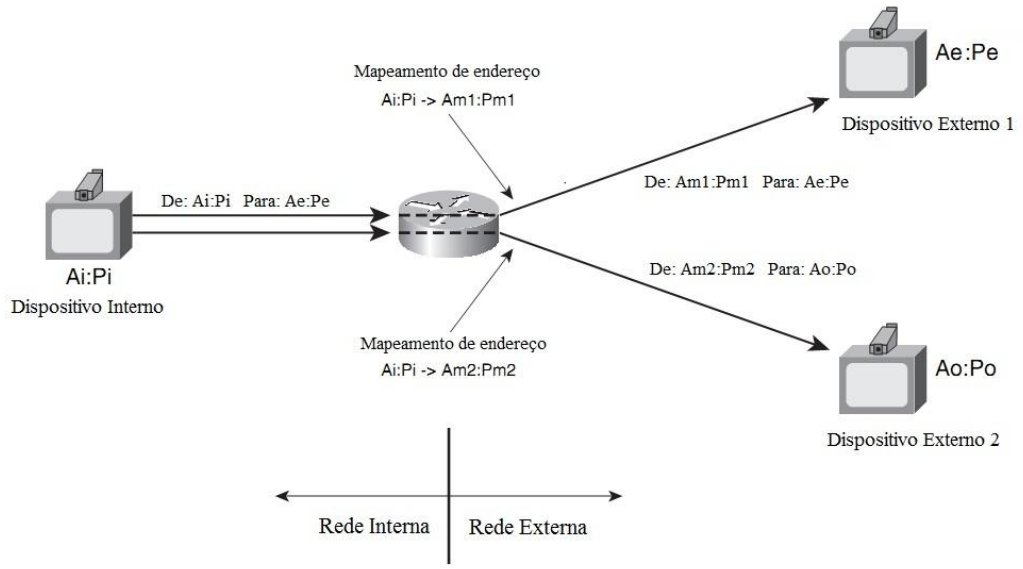

Figura 35 - Mapeamento Dependente de dispositivo, adaptada de [1].

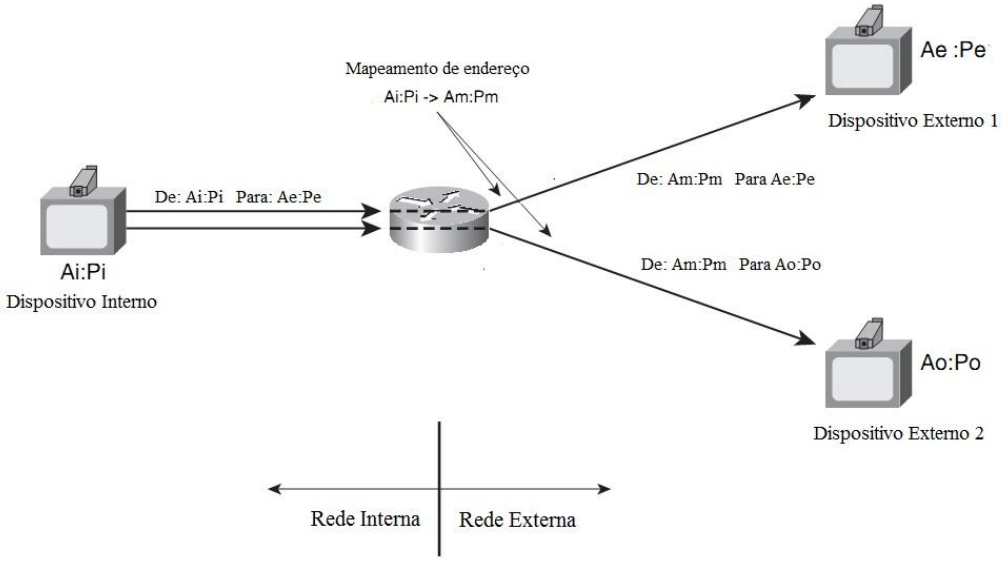

Figura 36 - Mapeamento Independente de dispositivo, adaptada de [1].

Na Figura 35, é possível observar que o endereço IP e porta privados Ai e Pi foram traduzidos em endereço IP e porta públicos Am1 e Pm1 para o dispositivo externo 1, e Am2 e Pm2 para o dispositivo externo 2. Já na Figura 36, Ai e Pi foram traduzidos apenas para Am e Pm, independentemente do dispositivo destino.

Com relação à característica de filtro, o mecanismo NAT pode ser classificado como independente de dispositivo, dependente de endereço ou dependente de endereço e porta. Antes de definir as classificações, é importante o entendimento do conceito de *binding*: uma vez que um dispositivo da rede interna envia pacote para dispositivo da rede externa, diz-se que foi criado um *binding*, através do qual o dispositivo externo então poderá enviar informações ao dispositivo interno. Sem a criação do *binding*, não há como o dispositivo externo saber qual o endereço do dispositivo interno, uma vez que a tradução NAT de endereços expira ao atingir um *timeout*, e uma nova tradução deve ser feita para um elemento interno, tradução essa que só é feita quando esse elemento interno envia uma mensagem para ambiente externo.

No tipo independente de dispositivo, qualquer pacote oriundo de dispositivo externo que seja endereçado ao IP e porta públicos mapeados serão aceitos, conforme Figura 37.

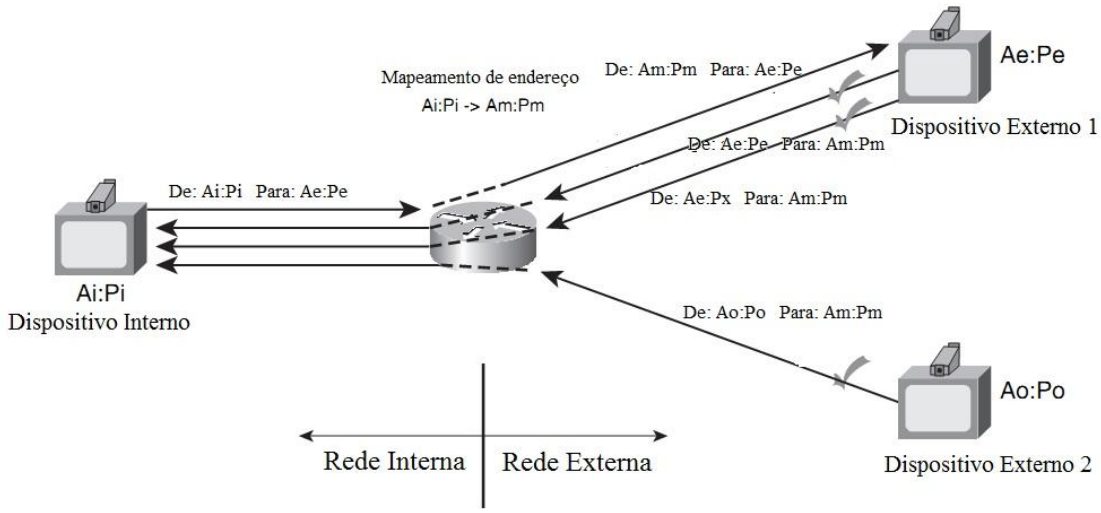

Figura 37 - Filtro Independente de dispositivo, adaptada de [1].

Já no segundo tipo, dependente de endereço, só serão aceitos pacotes de origem externa que tenham como destino o endereço e porta públicos mapeados e que ainda sejam oriundas de um endereço que foi destino de algum pacote proveniente do dispositivo interno. Esse processo pode ser melhor entendido através da Figura 38.

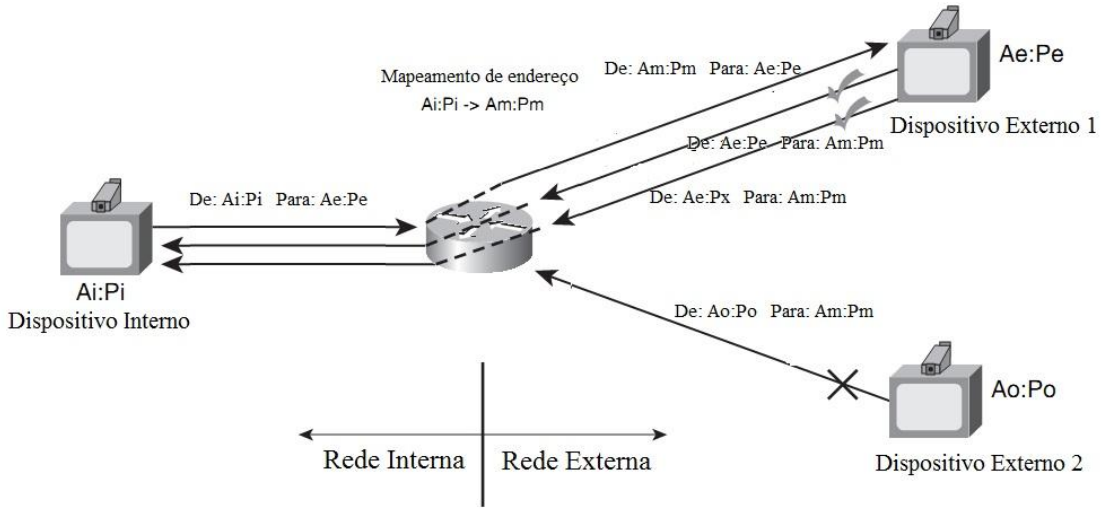

Figura 38 - Dependente de endereço, adaptada de [1].

O tipo dependente de endereço e porta é similar ao último citado, com a diferença de que a condição de porta também deve ser válida, conforme Figura 39.

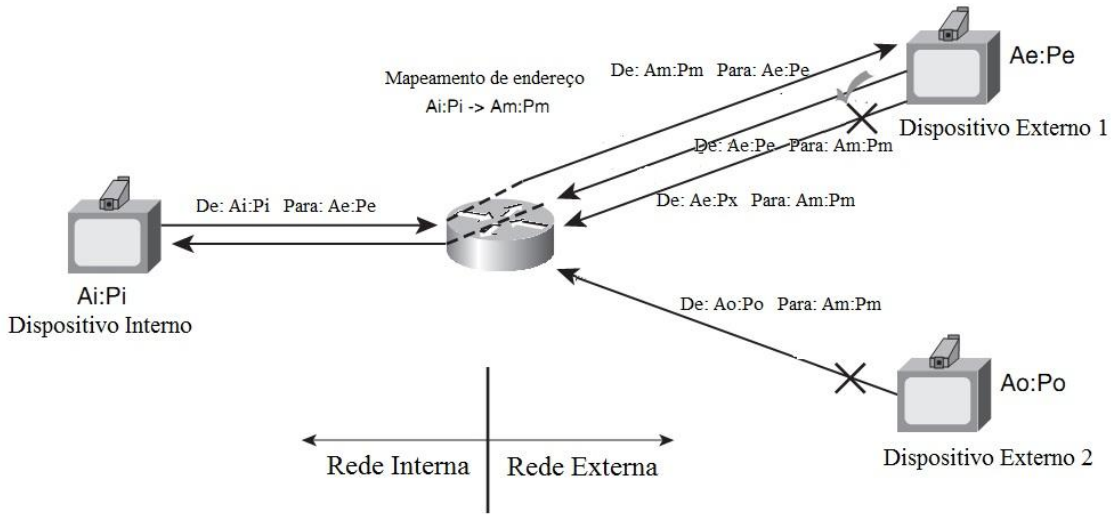

Figura 39 - Dependente de endereço e porta, adaptada de [1].

Em todos esses tipos, é necessário que esteja criado o *binding*, que como dito anteriormente, é através dele que os dispositivos externos conseguem intercambiar informações com dispositivo interno. Além do mais, um *binding* deve ser criado para cada dispositivo interno que precise se comunicar externamente.

O controle de *firewall* sobre o tráfego é feito basicamente através do bloqueio e liberação de portas. Diversas aplicações utilizam portas definidas, e as portas utilizadas pelo padrão H.323 são exibidas na Tabela 3.

Tabela 3 - Portas para sinalização H.323, adaptada de [1].

Função	Porta	Tipo de Transporte
RAS	Porta 1719	UDP
H.225	Porta 1720	TCP
H.245	Portas: 1024 – 65535	UDP
RTP e RTCP	Portas: 1024 – 65535	UDP

Como pode ser observado, a sinalização H.245 bem como os canais RTP e RTCP podem estar definidos em um grande gama de portas, o que traz um desafio ao *firewall*, uma vez que deixar toda essa variedade de portas liberadas fornece um risco grande de ataques. Alguns *firewall*s então são capazes de inspecionar as portas que estão sendo utilizadas para liberá-las e bloquear as outras, permitindo maior controle da informação. Essa capacidade implementada no *firewall* é denominada *Application Layer Gateway* (ALG).

Conclusão

Este trabalho apresentou um estudo sobre os fundamentos dos sistemas de áudio e vídeo conferência. De forma sintética, pode-se dizer que há, basicamente, dois tipos de vídeo conferência, diferenciados pela alocação prévia de recursos. Mais completas, as vídeo conferências com alocação prévia de recursos proporcionam uma característica importante para a experiência, que é a garantia de qualidade do serviço oferecido.

Um sistema de conferências de áudio e vídeo é composto de diversos elementos, *hardwares* e *softwares*, que se organizam em arquiteturas centralizadas ou distribuídas. Essas arquiteturas são compostas por camadas de controle e tratamento de mídia, com destaque para os *mixers*, os quais são responsáveis pelo tratamento de diversos *streams*, selecionando os adequados baseados geralmente numa política de detecção de voz, somando-os e assim obtendo um único *stream* de saída que é distribuído aos participantes. Todo o volume de dados de mídia é transportado através de pacotes RTP, e estabelecimento de chamadas e negociação de mídia com a utilização das recomendações SIP e H.323.

Para a transmissão de dados cada vez mais precisos, e consequentemente em maiores quantidades, são utilizadas diversas técnicas de compressão dos dados, com o objetivo de diminuir o volume de dados na transmissão, com melhor aproveitamento da banda disponível, e recuperar da melhor maneira possível os dados no lado do receptor. A técnica de codificação escalável provê ainda um fundamento bastante interessante no suporte a uma diversidade de dispositivos com capacidades variadas.

Uma vez que áudio e vídeo são definidos em *streams* distintos e a informação passa por diversos tratamentos e redes com latências variadas, há a necessidade de mecanismos para prover sincronização de áudio e vídeo, o que é alcançado com utilização de uma base de tempo comum na sincronização dos diferentes tipos de mídia transmitidos.

Como qualquer outro sistema em rede, os sistemas de conferência estão sujeitos a uma grande diversidade de ataques maliciosos, e a arquitetura do mesmo deve estar

composta de elementos que possam prover proteção, como utilização de *firewalls*, autenticação e NAT.

A realização deste trabalho foi ainda de grande valia na realização do estágio supervisionado, proporcionando conhecimento de diversos conceitos que regem os sistemas de vídeo conferência que puderam então ser aplicados em projetos que envolviam esse tipo de solução.

Referências Bibliográficas

[1] **Firestone, S.; Ramalingam, T.; Fry, S**. *Voice and Video Conferencing Fundamentals*. Cisco Press, 2007.

[2] **Even, R.**; **Ismail, N.**. RFC 4597, *Conferencing Scenarios*. 2006.

[3] **Cisco Systems**. *Cisco TelePresence TX9000 Series*. Disponível em: <http://www.cisco.com/en/US/products/ps12453/index.html>. Acesso em: Outubro de 2012.

[4] **Narbutt, M.; Murphy, L.** *Adaptive playout buffering for audio/video transmission over internet*. Performance Engineering Lab, University College Dublin, Dublin.

[5] **SALOMON, D**. *Data Compression: The Complete Reference*. Springer, 2006.

[6] **Schulzrinne, H.** et al. IETF RFC 3550, RTP: *A Transport Protocol for Real-Time Applications*. 2003.

[7] **Rosenberg, J.** et al. IETF RFC 3261. SIP: *Session Initiation Protocol*. 2002.

[8] **International Telecommunication Union (ITU-T).** H.323: *Packet-based multimedia communications systems*. 2009.

[9] **International Telecommunication Union (ITU-T).** H.225.0: *Call signalling protocols and media stream packetization for packet-based multimedia communication systems*. 2009.

[10] **International Telecommunication Union (ITU-T)**. H.245: *Control protocol for multimedia communications*. 2006.

[11] **Srisuresh, P.**; **Egevang, K.**. IETF RFC 3022, *Traditional IP Network Address Translator (Traditional NAT)*. 2001.