

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

**Adicionando Qualidade de Serviço
para um Ambiente de Colaboração Visual
Baseado em H.323.**

por

ALEXEI KORB

Dissertação submetida à avaliação,
como requisito parcial para a obtenção do grau de Mestre
em Ciência da Computação

Profa. Dra. Liane Margarida Rockenbach Tarouco
Orientadora

Porto Alegre, abril de 2003.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Korb, Alexei

Adicionando Qualidade de Serviço para um Ambiente de Colaboração Visual Baseado em H.323 / por Alexei Korb. – Porto Alegre: PPGC da UFRGS, 2003.

115f:il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2003. Orientadora: Tarouco, Liane Margarida Rockenbach.

1. Colaboração Visual. 2. DiffServ. 3. EPQoSE. 4. H.323. 5. QoS. 6. QoSPE. 7. TRM. I. Tarouco, Liane Margarida Rockenbach. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL.

Reitora: Prof^a Wrana Panizzi

Pró-Reitor de Ensino: Prof. José Carlos Ferraz Hennemann

Pró-Reitora Adjunta de Pós Graduação: Prof^a. Jocélia Grazia

Diretor do Instituto de Informática: Prof. Philippe Oliver Alexandre Navaux

Coordenador do PPGC: Prof. Carlos Alberto Heuser

Bibliotecária – Chefe do Instituto de Informática: Beatriz Regina Bastos Haro.

Agradecimentos

Durante o período que durou este curso de Pós-Graduação em Computação, além dos ensinamentos técnicos em ciências da computação, aprendi na prática diversas outras coisas, entre as quais que sempre estaremos acompanhados de outras pessoas e que o nosso sucesso vai depender da nossa capacidade de ajuda-las e ser ajudado por elas.

Esta dissertação de mestrado é um exemplo deste tipo de experiência, por que não existiria sem a contribuição de diversas pessoas. Com certeza muitas outras pessoas além das mencionadas aqui me apoiaram muito para que finalizasse este texto.

Agradeço aos meus pais João Carlos Korb e Mari M. B. Korb, pelo apoio e carinho que sempre demonstraram.

À Liane Tarouco, minha orientadora, por ter acreditado em mim, ter participado ativamente do desenvolvimento desta pesquisa e principalmente ter tido paciência para esperar o resultado final da pesquisa.

À José Félix diretor de tecnologia da NET S. A., que desde o primeiro dia, nos tempos de Net Sul, demonstrou apoio incondicional a realização deste curso de Mestrado e permitiu que eu pudesse participar das aulas ao mesmo tempo em que realizava minhas tarefas profissionais junto a NET S. A.

Aos professores João Netto e Lisandro Zambenedetti Granville, pelas discussões e elucidações sobre QoS, sistemas de gerenciamento baseado em políticas e SNMP respectivamente.

À equipe da Biblioteca da UFRGS, especialmente a Beatriz Haro e Ida Rossi, pela disponibilidade em ajudar sempre que requisitadas.

Sumário

Lista de Abreviaturas.....	6
Lista de Figuras	9
Resumo	11
Abstract	12
1 Introdução	13
1.1 Interação entre pessoas: cenário e tendências.....	13
1.2 Soluções inovadoras e seus benefícios	13
1.3 A evolução da videoconferência: sistemas de Colaboração Visual	14
1.4 O transporte multimídia em redes de pacotes	17
1.5 A proposta de trabalho.....	19
1.6 Visão geral e organização do texto	20
2 O modelo H.323: entidades e comunicação multimídia.....	22
2.1 A série H.323 e seus componentes	22
2.1.1 Terminais H.323	22
2.1.2 Gateways	25
2.1.3 Gatekeepers	26
2.1.4 MC e MP	28
2.1.5 MCU.....	29
2.2 Procedimentos de sinalização de chamadas	29
2.3 Fase de estabelecimento de chamada	31
2.3.1 Estabelecimento de chamadas simples (sem registro com gatekeeper)	31
2.3.2 Ambos pontos registrados com o mesmo gatekeeper	32
2.3.3 Pontos registrados com diferentes gatekeepers	39
2.4 Fase de comunicação inicial e troca de capacidades.....	41
2.5 Fase de comunicação multimídia	41
2.6 Fase de encerramento da chamada	42
2.7 Procedimentos adicionais para o estabelecimento de chamadas.....	44
2.7.1 Sinalização opcional usando a mensagem LRQ.....	44
2.7.2 Procedimento de conexão rápida.....	45
2.7.3 Pregranted ARQ	46
2.8 Observações práticas e comportamento dos protocolos.....	47
2.8.1 Descrição do ambiente observado	47
2.8.2 Problemas enfrentados	48
3 O H.323 e os requisitos básicos de qualidade de serviço.....	50
3.1 Qualidade de Serviço	51
3.2 Mantendo a qualidade para sinalização e controle de chamadas	52
3.3 A qualidade dos canais de mídia	54
3.4 Adicionando qualidade de serviço em aplicações de Colaboração Visual.....	55
3.4.1 A influência do sistema operacional.....	55
3.4.2 Qualidade de Serviço em nível de aplicação.....	56
3.4.3 Qualidade de serviço na rede.....	58

3.5 Controlando QoS em sistemas H.323.....	66
3.6 Uma nova perspectiva para qualidade de serviço em sistemas H.323	67
3.7 Um sistema de QoS baseado em políticas	70
3.7.1 Funcionamento geral de um sistema de QoS baseado em políticas	72
3.7.2 Características da requisição de QoS por mecanismos provisionados e sinalizados	73
3.8 Negociação de QoS entre terminais e o sistema de políticas	74
4 Mecanismo de requisição de QoS para um ambiente de colaboração visual baseado em H.323	77
4.1 Introdução	77
4.2 Arquitetura do ambiente.....	78
4.2.1 O módulo EPQoSE.....	79
4.2.2 Representação do modelo.....	81
4.3 Funcionamento do modelo	82
4.3.1 Fase de sinalização de QoS	83
4.4 Implementação do ambiente.....	86
4.4.1 O QoSM	87
4.4.2 Necessidades de implementação no QoSM.....	89
4.4.3 A classe H323Terminal	91
4.4.4 A classe QoSDescriptor.....	92
4.4.5 A classe TrafficDescriptor.....	93
4.4.6 A classe QosBroker	93
5 Comentários e considerações finais	97
5.1 Contribuições	97
5.2 Limitações e trabalhos futuros	98
5.3 Tabela Resumo.....	99
Anexo Proposta de Base de Informação de Gerenciamento (MIB) para a interação entre o EPQoSE o QoSM.....	101
Bibliografia.....	112

Lista de Abreviaturas

A/D	Analógico/Digital
ACELP	Algebraic Codebook Excitation Linear Prediction
ACF	Admission Confirmation
AD-PCM	Adaptive Differential Pulse-Code Modulation
API	Application Programming Interface
ARJ	Admission Reject
ARQ	Admission Request
BA	Behavior Aggregate
BB	Bandwith Broker
BCF	Bandwidth Confirmation
B-ISDN	Broadband – Integrated Services Digital Network
BRJ	Bandwidth Reject
BRQ	Bandwidth Request
CBQ	Class Based Queue
CIF	Common Intermediate Format
CLI	Command Line Interface
Codec	Coder/Decoder
COPS	Common Open Policy Services
CSDN	Circuit Switched Data Network
CSN	Circuit Switched Network
D/A	Digital/Analógico
DiffServ	Differentiated Services
DRQ	Disengage Request
DCF	Disengage Confirm
DSCP	Differentiated Services Code Point
DSL	Digital Subscriber Line
EPQoS	End-Point QoS Entity
ETSI	European Telecommunications Standards Institute
FEC	Forward Error Correction
GSTN	General Switched Telephone Network
GCF	Gatekeeper Confirmation
GK	Gatekeeper
GRJ	Gatekeeper Reject
GRQ	Gatekeeper Request
GW	Gateway
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol over Secure Socket Layer
IBM	International Business Machines Corporation ®
ID	Identifier
IETF	Internet Engineering Task Force
IntServ	Integrated Services
IP	Internet Protocol
IRR	Information Request Response
ISDN	Integrated Services Digital Network
ISO	International Standard Organization

ITU-T	International Telecommunication Union – Telecommunication
JAIN	Java Integrated Networks
JTAPI	Java Telephony API
Kbps	Kilobits per second
LAN	Local Area Network
LCF	Location Confirmation
LDAP	Lightweight Directory Access Protocol
L-DCELP	Low-Delay Code Excited Linear Prediction
LRQ	Location Request
MAN	Metropolitan Area Network
Mbps	Megabits per second
MC	Multipoint Controller
MCS	Multipoint Communication Service
MCU	Multipoint Control Unit
Mhz	Megahertz
MIB	Management Information Base
MIPS	Million Instructions Per Second
MP	Multipoint Processor
MPEG-1	Moving Picture Experts Group – layer 1
M-MLQ	MultiPulse-Maximum Likelihood Quantizer
NMS	Network Management System
NTSC	National Television Standards Committee
PCM	Pulse-Code Modulation
PDP	Policy Decision Point
PE	Parte Específica
PEP	Policy Enforcement Point
PG	Parte Genérica
PHB	Per Hop Behavior
PR	Policy Repository
PSDN	Packet Switched Data Network
PSTN	Public Switched Telephone Network
QAME	QoS-Aware Management Environment
QCIF	Quarter Common Intermediate Format
QoS	Quality of Service
QoSM	QoS Manager
QoSPE	QoS Policy Entity
RAS	Registration, Admission and Status
RCF	Registration Confirmation
RED	Random Early Detection
RFC	Request for Comments
RRJ	Registration Reject
RSVP	Resources Reservation Protocol
RTP	Real Time Protocol
RTCP	Real Time Control Protocol
RTCP RR	Real Time Control Protocol Receiver Report
RTCP SR	Real Time Control Protocol Sender Report
SLA	Service Level Agreement
SNMP	Simple Network Management Protocol
SSH	Secure Shell
TIPHON	Telecommunications and Internet Protocol Harmonization Over Networks

TCP	Transport Control Protocol
ToS	Type of Service
TRM	Transport Resource Manager
UCF	Unregister Confirmation
UDP	User Datagram Protocol
UML	Unified Modeling Language
URJ	Unregister Reject
URL	Universal Resource Locator
URQ	Unregister Request
VoIP	Voice over IP
WAN	Wide Area Network
WRED	Weighted Random Early Detection

Lista de Figuras

FIGURA 1.1 – Componentes.....	16
FIGURA 2.1 – Representação de um terminal H.323.....	23
FIGURA 2.2 – Configurações de um gateway.....	25
FIGURA 2.3 – Zona H.323.....	26
FIGURA 2.4 – Localizações possíveis para o MC e MP.....	28
FIGURA 2.5 – Captura da de uma sessão H.323 mostrando a fase de estabelecimento de conexão. (usando o Lan Explorer v.3.6 da Sunrise Telecom).	3130
FIGURA 2.6 – Estabelecimento de chamadas básico. Sem Gatekeepers.....	32
FIGURA 2.7 – Descobrimto automático do gatekeeper.....	33
FIGURA 2.8 – Fase de registro com o gatekeeper.....	33
FIGURA 2.9 – Fase de desregistro com o gatekeeper.....	34
FIGURA 2.10 – Estabelecimento de chamadas roteado pelo gatekeeper.....	35
FIGURA 2.11 – Estabelecimento de chamadas direto.....	36
FIGURA 2.12 – Canal de controle no modo roteado pelo gatekeeper.....	37
FIGURA 2.13 – Dois pontos registrados pelo mesmo gatekeeper no modo de sinalização direto.....	38
FIGURA 2.14 – Dois pontos registrados pelo mesmo gatekeeper, no modo de sinalização roteado.....	39
FIGURA 2.15 – Dois pontos registrados em gatekeepers distintos,.....	40
FIGURA 2.16 – Encerramento de chamada iniciado pelo terminal.....	43
FIGURA 2.17 – Encerramento de chamada iniciado pelo gatekeeper.....	43
FIGURA 2.18 – Sinalização opcional usando a mensagem LRQ.....	45
FIGURA 2.19 – Diagrama do ambiente observado.....	47
FIGURA 2.20 – Log do MCU MeetingPoint mostrando encerramento da sessão.....	49
FIGURA 3.1 – Introdução de atraso para sinais multimídia.....	55
FIGURA 3.2 – Os componentes básicos da arquitetura Intserv.....	60
FIGURA 3.3 – Modelo de reserva de recursos usando RSVP.....	61
FIGURA 3.4 – O modelo DiffServ.....	63
FIGURA 3.5 – Seqüência de mensagens para estabelecer um canal RSVP unicast.....	67
FIGURA 3.6 – Relacionamento entre as entidades funcionais de QoS.....	69
FIGURA 3.7 – Visão esquemática de um sistema de Gerenciamento baseado em Políticas.....	71
FIGURA 3.8 – Encerramento de chamada por falta de recursos de rede.....	75
FIGURA 4.1 – Arquitetura do terminal H.323 proposto.....	78
FIGURA 4.2 – Diagrama de classes do EPQoSE.....	81
FIGURA 4.3 – Fase de sinalização de QoS.....	84
FIGURA 4.4 – Componentes introduzidos no modelo proposto.....	86
FIGURA 4.5 – Arquitetura PBNM padrão.....	87
FIGURA 4.6 – Componentes do PDP.....	88
FIGURA 4.7 – Componentes da parte específica do PDP.....	89
FIGURA 4.8 – Mapeamento entre entidades funcionais do sistema QAME.....	89
FIGURA 4.9 – H323QoS-MIB.....	90

Lista de Tabelas

TABELA 2.1– Tipo de terminal H.323 para determinação mestre/escravo H.245	28
TABELA 3.1 – Nível de serviço para o estabelecimento de chamadas em aplicações de VoIP	53
TABELA 3.2 – Exemplo de classes de serviço usando Diffserv class selector PHB.....	64
TABELA 3.3 – Exemplo de classes de serviço usando Diffserv Assured Forwarding (AFxy) PHB	64
TABELA 4.1 – Variável da classe QoSBroker a ser configurado na tabela qosControlTable.....	94
TABELA 4.2 – Endereços IP e máscara de sub-rede a serem configurados na tabela qosControlTable.....	94
TABELA 4.3 – Valores adicionais a serem configurados na tabela qosControlTable ..	94
TABELA 4.4 – Variável da classe QoSBroker a ser configurado na tabela qosDescriptorTable	94
TABELA 4.5 – Variável da classe QoSDescriptor a ser configurado na tabela qosDescriptorTable	95
TABELA 4.6 – Variável da classe QoSBroker a ser configurado na tabela trafficDescriptorTable.....	95
TABELA 4.7 – Variável da classe TrafficDescriptor a ser configurado na tabela trafficDescriptorTable.....	95
TABELA 4.8 – Valores adicionais a serem configurados na tabela trafficDescriptorTable.....	96
TABELA 5.1 – Tabela resumo de características.....	100

Resumo

A colaboração visual, recurso que permite a troca de informações de forma remota, é construída em cima de uma combinação de diversas ferramentas, na qual estão incluídos: videoconferência, “*streaming de vídeo*”, compartilhamento e transferência de informações e imagens (colaboração em cima de dados) entre outros.

Estas soluções, vêm utilizando cada vez mais, o protocolo IP e a Internet para o transporte dos sinais. Com este objetivo, o ITU-T lançou a recomendação H.323, que definiu um padrão confiável, que permite a troca de sinais multimídia em redes de pacotes sem qualidade de serviço. Entretanto, com o passar dos anos percebeu-se que aplicações que manipulam voz e vídeo, precisam que as redes de pacotes tenham capacidade de prover características semelhantes às oferecidas por redes de comutação por circuito, para o transporte dos sinais multimídia. Neste sentido, redes IP podem utilizar mecanismos de qualidade de serviço como o DiffServ, para prover tratamento adequado dos sinais de áudio e vídeo e assim, aumentar a qualidade percebida pelos usuários. As aplicações de colaboração visual são notáveis candidatas a utilização de mecanismos de QoS da rede. Neste caso é desejável que estas aplicações estejam aptas a especificar o nível de qualidade de serviço desejado e requisitem este nível de serviço para a rede.

Neste contexto, o trabalho apresenta um modelo para o desenvolvimento de um terminal H.323 capaz de requisitar qualidade de serviço para a rede IP, visando aumentar a qualidade percebida pelo usuário em relação aos sinais de mídia. Neste terminal foi incluída uma entidade chamada de EPQoSE, responsável pela sinalização de QoS em benefício do terminal. Além disso, o modelo proposto neste texto apresenta um sistema de gerenciamento baseado em políticas, responsável por controlar as requisições de QoS dos terminais H.323, dentro de um domínio. Como o terminal precisa se comunicar com estas entidades, apresentamos no trabalho, a maneira como ele faz isso e definimos um conjunto de funções que devem ser implementadas pelo QoSM no terminal.

Palavras-chave: Colaboração Visual, DiffServ, EPQoSE, , gerenciamento, H.323, qualidade de serviço (QoS), QoSM, , políticas.

TITLE: “ADDING QUALITY OF SERVICE FOR A H.323 BASED VISUAL COLLABORATION ENVIRONMENT”**Abstract**

Visual collaboration allows remote information interchange combining a set of tools which includes: videoconferencing, video streaming, data sharing and interchange, information and images transfer and so on.

These systems are using more and more the IP protocol and the Internet to transport these signals. For this reason, the ITU-T has launched the H.323 recommendation which defined a standard for multimedia signals interchange in packet networks with no quality of service features. However, it is well known that applications that handle voice and video signals need that packet networks are able to provide a similar behaviour provided by circuit switched networks for multimedia signal transport. In this way, IP networks may use quality of services mechanisms like DiffServ, to provide a suitable treatment for audio and video signals and doing so, improve the quality perceived by the users. Visual collaboration applications are remarkable candidates to use networks QoS mechanisms. In this case, it is desirable that these applications may specify and request for the network the desired level of quality of service.

In this context, this thesis presents a model for the development of an H.323 terminal that is able to signal and request quality of service for the IP network, with the aim to improve the quality of service perceived by the user related to audio and video signals. Into the H.323 terminal was placed a new entity called EPQoSE, responsible to signal QoS needs in behalf of the terminal. Beyond this, the model proposed in this text presents a policy based management system, responsible to control all QoS requests from H.323 terminals, inside a unique domain. We also present in this text how the H.323 terminal will communicate with these entities and so we proposed a set of functions that should be implemented by the QoSM, inside the terminal.

Keywords: Visual collaboration, DiffServ, EPQoSE, , management, H.323, quality of service (QoS), QoSM, policy.

1 Introdução

1.1 Interação entre pessoas: cenário e tendências

Cada aspecto de nossa vida tem mudado de forma muito rápida nos últimos 50 anos. Os valores da sociedade, o modo de se fazer negócios, as habilidades necessárias para o trabalho, a capacidade intelectual humana. Na área da tecnologia, o avanço que aconteceu nos últimos trinta anos, ocorreu de maneira mais rápida do que nos três últimos séculos [CAS 99].

Antes, o suor e a força representavam riqueza e poder. Hoje, não são eles, nem a posse da terra ou do capital que fazem a diferença para o sucesso. Hoje, o diferencial das pessoas de alto potencial, está na posse do conhecimento, está na busca constante do aprimoramento, na capacidade de interagir com um grande número de culturas diferentes assimilando e entendendo as divergências existentes.

O conhecimento, diferente de outros bens, não diminui quando compartilhado, ao contrário, aumenta [SVE 98]. Multiplica-se, além daqueles que participam diretamente da transferência do conhecimento.

A nova “sociedade do conhecimento”, aliada ao espantoso avanço trazido pela revolução da tecnologia da informação das últimas três décadas, está permitindo a disseminação do conhecimento de qualidade além das fronteiras dos países desenvolvidos, abrindo a possibilidade para que os países em desenvolvimento incrementem e melhorem a capacidade intelectual nacional e busquem através da disseminação desta cultura construir um futuro diferente, melhor e mais justo.

Estes novos requerimentos da sociedade estimulam o desenvolvimento de soluções inovadoras, que facilitem e melhorem a qualidade da comunicação entre as pessoas. Estudos mostram que o método mais comumente usado para a transferência de conhecimento – a palestra – é por incrível que pareça, o método menos eficaz. Depois de cinco dias a maioria das pessoas recorda de menos de um décimo do que escutaram na palestra ou aula. Uma combinação auditiva combinada com recursos visuais é preferível e produz uma retenção de aproximadamente 20 por cento do conteúdo [SVE 98].

Concomitantemente, nossa rede de relacionamentos se espalha pelos quatro cantos do globo e a comunicação entre as pessoas passou a ser mais do que mera necessidade social. A globalização, criou uma necessidade ainda maior para que a comunicação entre as pessoas seja mais rápida, eficaz e completa.

Como se isso não bastasse, a velha maneira de interação entre apenas duas pessoas ao mesmo tempo – através do telefone – não está mais suprimindo nossas necessidades. Agora queremos também, a comunicação entre várias pessoas ao mesmo tempo, mesmo que elas estejam muito distantes umas das outras.

1.2 Soluções inovadoras e seus benefícios

O surgimento de redes com capacidade multimídia permitiu a entrega de diferentes tipos de informação através do mesmo meio de transmissão, facilitando o trabalho de construção de ferramentas que combinam diversas formas de conteúdos

entre os quais estão 3D e realidade virtual, textos e imagens tradicionais, sons, dados, vídeo, gráficos, entre outros, que combinados entre si proporcionam diferentes formas de comunicação audiovisual e conseqüentemente estimulam e melhoram a construção do conhecimento.

Uma das mais eficazes ferramentas deste tipo são os sistemas de videoconferência, que permitem a interação bidirecional em tempo real entre os participantes da reunião, aumentando também a compreensão e a qualidade da discussão, ajudando os participantes sentirem-se mais próximos uns dos outros. Mas esta não é a única vantagem que sistemas deste tipo oferecem. Podemos ainda listar outros benefícios, como apontados na obra de Schaphorst [SCH 99]:

- Tomada de Decisões mais rápidas – Videoconferências podem ser estabelecidas com menor dificuldade que encontros “in loco” e pessoas separadas por quilômetros podem compartilhar idéias e informações.
- Decisões mais qualificadas – Como não existem custos adicionais por participante, as conferências podem incluir todas as pessoas responsáveis. Se viagens fossem necessárias, provavelmente isto não ocorreria, ou pelo menos seria muito mais difícil de acontecer.
- Aumento da produtividade - Funcionários altamente capacitados não podem ficar perdendo tempo nos aeroportos. Com a videoconferência, se a reunião têm a duração de duas horas, os envolvidos irão gastar apenas duas horas do seu tempo.
- Redução dos custos de viagens – Muitos custos estão associados com as viagens, pois elas envolvem passagens, hospedagem, alimentação, estacionamento e assim por diante. A videoconferência ajuda a eliminar estes gastos.
- Redução do cansaço – Viagens de negócios são geralmente cansativas, podendo resultar em fadiga e performance reduzida dos colaboradores. Este também é um grande benefício da videoconferência, que ajuda eliminar a fadiga resultante de diversas viagens.
- Ajuda na construção de uma equipe – Através de encontros freqüentes, juntando nas reuniões todos os envolvidos, é possível atingir decisões por consenso e não decisões verticais, o que resulta em maior grau de motivação e energia para todos os membros da organização.

Outra vantagem bastante importante é o resguardo da integridade física das pessoas. Sempre que as pessoas necessitam viajar para participar de um encontro ou evento, estão correndo risco. A videoconferência diminui a necessidade de exposição destas pessoas a estes riscos, ajudando a aumentar o nível de segurança das pessoas.

1.3 A evolução da videoconferência: sistemas de Colaboração Visual

Os primeiros sistemas de videoconferência utilizavam exclusivamente transmissão via satélite, que embora ofereça alto grau de qualidade para os sinais de áudio e vídeo, apresenta também elevado custo de manutenção.

Em situações onde se deseja transmitir sinais de áudio e vídeo a partir de um único local para todas as outras localidades envolvidas na transmissão, as soluções via satélite ainda são vistas como uma boa opção, por que conseguem atingir praticamente qualquer localidade do globo terrestre. Entretanto, tornam-se especialmente caras para obtenção de interatividade bidirecional entre todas as localidades participantes, pois neste caso uma enorme infra-estrutura precisa ser disponibilizada.

Na busca por soluções eficientes e mais baratas o ITU-T definiu a recomendação H.320, publicada em dezembro de 1990. Esta recomendação especifica como sinais multimídia, devem ser processados e transmitidos em redes ISDN. A solução de videoconferência H.320 reduziu o custo, quando comparado à manutenção dos canais de satélite. Entretanto, este novo sistema apresenta um problema diferente: a tecnologia ISDN ainda não se encontra disponível em todas partes de um país, especialmente em regiões interioranas, tornando quase impossível a composição de uma solução completa baseada exclusivamente nesta tecnologia.

Com a Internet tornando-se onipresente, e o reconhecimento de que o futuro da comunicação está baseado no protocolo da Internet – o IP – os fabricantes começaram a investir no desenvolvimento de produtos voltados para a realização de videoconferência através da Internet. Os primeiros produtos, embora bastante eficientes, foram desenvolvidos com tecnologia proprietária, o que impedia a integração dos sistemas. Entretanto, a eficiência destes sistemas perde em importância para a interoperabilidade, suprimida pela inexistência de padrões.

Em 1996, o ITU-T lançou a recomendação H.323, que alavancou o trabalho para a definição de um padrão confiável, que permitisse a troca de sinais multimídia em redes de pacotes sem qualidade de serviço. Com liberação da segunda versão do H.323, em fevereiro de 1998 e sua adoção pelo fórum de voz sobre IP, o H.323 passou a chamar a atenção de todos os fabricantes e tornou-se o primeiro padrão de fato para a comunicação multimídia em redes IP.

O principal motivo que levou os diversos fabricantes a abandonarem seus protocolos proprietários e adotar o H.323 para comunicação multimídia em tempo real através da Internet foi sem dúvida a possibilidade de interoperação com produtos de diferentes fabricantes.

Outra importante característica do H.323 é que ele mantém a compatibilidade com sistemas mais antigos e que já possuíam grande base instalada, tal como sistemas H.320. Adicionalmente o padrão prevê a comunicação com diferentes tecnologias de transporte como: PSTN, ATM e outras.

O H.323 prevê também, a integração com recursos de colaboração de dados, através da utilização do T.120 [ITU 97b] – outra recomendação desenvolvida pelo ITU-T – proporcionando experiências de comunicação mais enriquecedoras aos usuários envolvidos.

Combinando as facilidades de áudio, vídeo e dados, o H.323 estendeu o conceito de comunicação visual – apenas áudio e vídeo – para a colaboração visual, chamada também de colaboração integrada [DAV 99] e telefonia IP [KUM 2001], que engloba bem mais que reuniões virtuais.

A colaboração visual ou telefonia IP permite a construção de relacionamento e troca de informações, eliminando quase a zero, o ruído que afetava a comunicação remota. A colaboração visual é construída em cima de diversas ferramentas, na qual estão incluídos: videoconferência, “*streaming de vídeo*”, compartilhamento e

transferência de informações e imagens (colaboração em cima de dados), web-based call centers, e muito mais conforme descrito em [HAS 2000].

Nestes sistemas, diferentes níveis de interação entre os participantes podem ser obtidos, permitindo o estabelecimento de um vínculo de relacionamento bem maior que o obtido em qualquer outro tipo de ferramenta anterior. A construção de relacionamento é conseguida em função da quantidade e qualidade das ferramentas disponíveis, como ilustrado abaixo:



FIGURA 1.1 - componentes utilizados para prover a colaboração visual

Quanto mais poderosos os recursos de colaboração e quanto maior a qualidade dos recursos de comunicação, melhor será o resultado final, tanto no sentido do número de possibilidades de informações que podem ser compartilhadas quanto permitindo que a construção de relacionamento seja estabelecida mais rapidamente entre os participantes.

Agregando o poder dos computadores à tecnologia para a transferência de vídeo, voz e dados em cima da Internet, um sistema de colaboração visual oferece todas as ferramentas necessárias, na busca por respostas rápidas às necessidades de comunicação da “era do conhecimento”, mantendo um grau de interação que só podia ser obtida anteriormente em reuniões presenciais.

A colaboração visual como ferramenta de comunicação e troca de informações, facilita a criação de vínculos de confiança entre as partes que se comunicam e portanto sua utilização agrega valor em diversas atividades, como:

- Educação: treinamentos, cursos de línguas estrangeiras, participações em seminários ou congressos, entre outras atividades que visam o desenvolvimento intelectual das pessoas.
- Atendimento a clientes: Neste caso, com a popularização dos serviços através da Internet, é possível montar uma central de atendimento, onde os atendentes passam a resolver os problemas dos clientes como se estivessem ao lado do mesmo, evitando o deslocamento até o local do cliente e economizando tempo necessário para isto, reduzindo o custo associado.
- Desenvolvimento de projetos: é possível reunir pessoas dispersas em diversos lugares do mundo para o desenvolvimento de um projeto, sem a necessidade de reuniões presenciais. Este tipo de ferramenta facilita a realização de “projetos relâmpagos” – onde parcerias são montadas entre as empresas, apenas para o desenvolvimento de um projeto e logo depois de encerrado o projeto, a parceria é terminada.
- Telemedicina: Será possível a realização de diagnósticos complicados muito mais rapidamente, através da discussão conjunta entre grandes especialistas sobre um assunto da medicina, economizando tempo, reduzindo custos, salvando mais vidas e aumentando a qualidade no atendimento.
- Telefonia: O modelo de telefonia baseado no protocolo IP, permite que os usuários possam colaborar usando telefones IP, o que não é possível nos sistemas atuais. Estes telefones, podem se conectar aos telefones tradicionais da rede pública de telefonia e até mesmo a telefones móveis celulares.

1.4 O transporte multimídia em redes de pacotes

Tradicionalmente a Internet foi desenvolvida para transportar dados e a rede de telefonia, foi projetada para carregar sinais de voz. Estes sinais possuem características diferentes e por isso os conceitos sobre os quais estas duas redes foram construídas, são também diferentes. Redes de dados trabalham com tecnologia de comutação de pacotes enquanto que as redes para transmissão de voz trabalham com tecnologia de comutação por circuitos.

A tecnologia de comutação por circuitos, permite a transmissão contínua de informações através de um canal dedicado previamente estabelecido entre a origem e destino. Para poder enviar e receber informações, a origem deve primeiramente estabelecer uma conexão com o destino. No momento em que o destino aceita a conexão, recursos da rede são reservados exclusivamente para os participantes da chamada. Este modelo garante que não haverá interrupções no sinal trocado entre os participantes e as informações chegarão ao destino na mesma ordem em que foram enviadas pela origem. Depois do estabelecimento da conexão as informações podem transitar de um lado para o outro, sempre através do canal estabelecido. Quando não existir mais a necessidade de comunicação, uma das partes encerra a conexão, o canal é liberado e os recursos da rede podem ser alocados para outra chamada.

Este tipo de serviço oferece a vantagem de manter uma taxa constante na entrega das informações, reduzindo o atraso fim-a-fim. Entretanto, a mesma solução limita o número de usuários que podem utilizar simultaneamente a rede de forma eficiente, uma vez que os recursos de rede são alocados exclusivamente para cada chamada. Em situações onde existem muitas conexões para serem estabelecidas através do mesmo canal físico de comunicação, o custo para manter o nível de qualidade aceitável é bastante alto, em razão da necessidade de manutenção de canais de maior capacidade.

As redes de comutação de pacotes, como a Internet, foram projetadas para otimizar o uso da largura de banda e minimizar os custos associados à manutenção dos canais físicos, mas não garantem que a informação irá realmente chegar ao seu destino em um limite de tempo máximo e a uma taxa constante, como na situação anterior.

Neste modelo as informações podem também seguir por caminhos diferentes, já que não existe um canal dedicado e por conseqüência, os pacotes que carregam as informações podem chegar fora da ordem original em que foram enviadas. Como a banda disponível é compartilhada por todos usuários, no caso onde vários usuários enviam muitas informações simultaneamente, situações de congestionamento podem surgir e por isso alguns pacotes podem ser retidos, causando atraso e variação no tempo de entrega dos pacotes (jitter). No pior caso, os dados podem ser descartados, necessitando de reenvio por parte da origem.

Aplicações emergentes como as aplicações e serviços avançados multimídia, precisam que as redes de comutação de pacotes tenham capacidade de prover características semelhantes às oferecidas por redes de comutação por circuito, para o transporte dos sinais. Este é um requisito essencial para prover a entrega contínua dos dados, minimizar as ocorrências de atraso, variação no atraso e perda de pacotes e assim manter a qualidade da comunicação [LI 2000], [GLI 2000].

Adicionalmente, as necessidades de largura de banda e outros parâmetros de qualidade são diferentes para cada um dos serviços e aplicações que compõe uma solução de colaboração visual. Quando possível, mecanismos que garantam níveis de serviços diferenciados são desejados e devem ser usados.

Em sistemas empregados para Educação à Distância ou Telefonia, onde a mensagem falada precisa ser compreendida sem perda de conteúdo, os sinais de voz são muito mais importantes do que quaisquer outros tipos de recursos visuais. Nestes sistemas, é necessário minimizar a ocorrência de jitter, manter o atraso unidirecional menor que 150 ms e garantir a largura de banda na faixa de 8 Kbps e 64 Kbps.

Em aplicações para Telemedicina, diagnósticos on-line e tele-monitoramento, a precisão, qualidade e prontidão dos sinais de vídeo são mais importantes e devem portanto, ser priorizados e entregues com a menor degradação possível, de forma a manter a integridade e inteligibilidade da imagem.

Aplicações para a troca e compartilhamento de dados, diferente dos casos anteriores não necessitam de tanto esforço por parte da rede em relação ao atraso, variação no atraso, largura de banda e dessequencialização de pacotes. Entretanto, é extremamente sensível à perda de pacotes e pacotes entregues com erros.

Prover estes diferentes tipos de requisitos sobre a Internet, requer empregar ou desenvolver soluções e tecnologias que garantam que o serviço prestado seja confiável, eficiente e seguro, mantendo a qualidade esperada.

A telefonia IP, embora já tenha se tornado uma indústria de milhões de dólares, está longe de ter alcançado a maturidade [HER 2000], e pelo menos algumas questões importantes precisam ainda ser resolvidas para que tenhamos produtos e serviços que possam ser empregados comercialmente com níveis de qualidade aceitáveis [LAW 99]:

Qualidade de serviço – Oferecer serviços de colaboração visual ou telefonia através da Internet, significa usar uma rede com aproximadamente 600 milhões de usuários em todo o mundo, que é controlada por diversas organizações governamentais e privadas, grandes, médias ou pequenas, localizados em países desenvolvidos ou não. Agregar qualidade de serviço nestas condições não é uma tarefa fácil. Além disso, quando o usuário deseja usar serviços com qualidade garantida, devem existir acordos de níveis de serviço, entre os diferentes provedores de serviços, onde a classe de serviço

usada por um prestador de serviço é mapeada para uma classe de serviço equivalente de outro prestador de serviço e assim por diante, para que o usuário seja cobrado adequadamente e os seus requisitos cumpridos apropriadamente.

Segurança e privacidade – A Internet é uma rede pública e por isso está exposta a diversos riscos. Mesmo em ambientes controlados, o IP possui diversas vulnerabilidades conhecidas que precisam ser tratadas adequadamente pelos donos da rede, de forma a aumentar o nível de segurança do ambiente. Para que aplicações como Telefonia IP sejam empregadas comercialmente, mecanismos que garantam a segurança e a privacidade das informações que estão trafegando na rede, devem necessariamente existir, uma vez que o sigilo telefônico é garantido por lei na maioria dos países. Os mecanismos de proteção a serem utilizados devem englobar autenticação, autorização, confidencialidade, integridade e não-repúdio.

Disponibilidade – A disponibilidade é um fator crítico para os serviços de Telefonia. Em situações onde acordos de níveis de serviço precisam ser cumpridos ela torna-se fator tão importante quanto os dois mencionados anteriormente. Nos países onde existem órgãos regulamentadores, que fiscalizam a qualidade do serviço prestado (ex.: ANATEL), as prestadoras são constantemente verificadas e a disponibilidade é rigorosamente acompanhada. A disponibilidade pode ser alcançada através de diferentes formas como: redundância de hardware e software, bem como redundância de canais de comunicação e preparação para situações de emergência, desastre ou outros acontecimentos.

Integração entre a Internet e a rede de Telefonia – Eventualmente soluções de telefonia através da PSTN e da Internet deverão ser integradas. Esta integração deve prever no mínimo a manutenção dos atuais padrões de níveis de serviço. Entretanto a Internet e a PSTN possuem grandes diferenças de infra-estrutura:

- Diferenças significativas entre as versões nacionais e internacionais do mesmo protocolo.
- Diferenças conceituais e procedurais entre comutação por circuitos e comutação por pacotes, como visto anteriormente.
- Diferenças de numeração entre a PSTN e a Internet
- Diferenças operacionais entre as estruturas de controle (a PSTN é controlada de forma centralizada e a Internet de forma distribuída).

Embora, muitas destas questões foram previstas para serem realizadas por entidades como gateways e gatekeepers – no modelo H.323 – as soluções são muito dependentes da interpretação e entendimento que os fabricantes de equipamentos têm do padrão. Eventualmente estas diferenças tendem a diminuir, mas o fato é que elas ainda existem.

Resolver todas estas questões é uma tarefa extremamente complexa, além de levar períodos de tempo mais longos, pois os órgãos de padronização precisam entrar em acordo até que os padrões sejam definidos, aceitos pelos fabricantes e realmente implementados.

1.5 A proposta de trabalho

O objetivo deste trabalho é estudar o protocolo H.323 de maneira profunda e abrangente para identificar os fluxos de comunicação empregados pelo H.323, seus

comportamentos, suas características, bem como identificar quais destes fluxos de comunicação necessitam de tratamento com qualidade de serviço e por que eles requerem este tipo de tratamento. Neste sentido, o trabalho está concentrado em apenas uma das quatro questões mencionadas na sessão anterior: o provimento de Qualidade de Serviço para aplicações de colaboração visual baseadas na recomendação H.323.

Para identificarmos os canais H.323 que requerem tratamento com QoS, devemos estudar a recomendação H.323 e monitorar sessões de comunicação H.323 sob diversas situações de tráfego na rede. Através destes monitoramentos pretendemos investigar os motivos das falhas e interrupções dos serviços em decorrência de sobrecarga na rede, perda de pacotes, atrasos e outras situações que uma rede de pacotes está sujeita. Comparando estas observações com a recomendação e outras referências bibliográficas, poderemos apontar fatores que contribuem para a degradação da comunicação em certas circunstâncias, bem como fatores que podem ser empregados para aumentar a qualidade percebida pelo usuário final.

Nosso plano de trabalho visa definir um modelo para a construção de um ambiente de colaboração visual baseado no H.323 capaz de solicitar para entidades de rede as necessidades de qualidade de serviço que o terminal deseja ter em uma determinada sessão multimídia. Assim, o resultado deste estudo deve gerar a construção de um cliente (terminal H.323) para o trabalho colaborativo utilizando uma pilha H.323 distribuída gratuitamente. Este terminal deve implementar um mecanismo para que aplicações H.323 possam requisitar para a rede um tratamento adequado às necessidades dos usuários, aumentando o nível de qualidade percebido.

A idéia é deixar que os mecanismos de QoS existentes nos dispositivos de rede sejam utilizados para proporcionar a qualidade esperada. Assim, uma vez que a rede tenha aceitado a requisição de QoS do terminal H.323, a própria rede IP se encarregará de tratar adequadamente os pacotes pertencentes a estes fluxos de dados.

Acreditamos que a implementação adequada de soluções de colaboração visual em redes IP, está associada a uma combinação de fatores que vão desde a utilização de mecanismos de QoS da rede, até o correto dimensionamento configuração e parametrização dos dispositivos que irão compor a solução. Por isso, a utilização de terminais H.323 que tenham capacidade para requisitar QoS da rede, em conjunto com um modelo coerente de controle e gerenciamento de QoS, oferecem uma abordagem interessante e eficaz para agregar qualidade às soluções de colaboração visual baseadas no H.323.

O resultado do trabalho deverá proporcionar um aumento na percepção de qualidade dos sinais percebida pelos usuários, redução do atraso unidirecional fim-a-fim, reserva de largura de banda para fluxos de comunicação multimídia, bem como redução da taxa de perdas de pacotes para os fluxos de mídias gerados por sistemas H.323.

1.6 Visão geral e organização do texto

Este texto está dividido em cinco capítulos, cada um com um objetivo diferente conforme mostrado a seguir:

O texto é iniciado com uma introdução geral ao assunto, no capítulo 1. Nesta parte, o assunto é introduzido e os conceitos de colaboração visual são apresentados em relação ao contexto social atual. Nesta parte também são explicadas as vantagens do uso de sistemas de colaboração visual. Além disso, um breve histórico da evolução destes sistemas é provido. Em seguida, é também oferecida uma rápida explicação sobre as

diferenças existentes entre o transporte multimídia em redes de pacotes e redes de circuitos. Por fim, é introduzida a proposta a ser desenvolvida neste trabalho.

O capítulo 2, o modelo H.323, têm o objetivo de explicar como a recomendação H.323 funciona. Este capítulo inicia descrevendo os componentes da série H.323, (terminais, gateways, gatekeepers, MC, MP e MCU) suas características e funções. Logo em seguida, são explicadas cada uma das fases necessárias para o estabelecimento da conferência. Estas fases são: sinalização e estabelecimento da chamada, registro com o gatekeeper, estabelecimento do canal H.245 e troca de capacidades, estabelecimento dos canais multimídia e encerramento da chamada.

No capítulo 3, são apresentados os requisitos básicos de qualidade de serviço para sistemas H.323 e os parâmetros de QoS que devem ser observados, para cada um dos canais de comunicação do sistema H.323: RAS, sinalização, controle de chamadas e canais de mídia. Neste capítulo são apresentados para cada um destes canais os parâmetros de qualidade que precisam ser tratados e mantidos para que o usuário final tenha uma boa percepção da qualidade do serviço. Neste mesmo capítulo são ainda mostrados como pode-se agregar QoS em sistemas H.323 e quais as diferenças entre o provimento de QoS em nível de aplicação ou em nível de rede. São apresentados também, dois modelos diferentes para o provisionamento de QoS em redes IP o IntServ/RSVP e o DiffServ. Para finalizar o capítulo, é mostrado uma nova perspectiva para o gerenciamento de QoS em redes IP, através de um sistema de gerenciamento através de políticas. Nesta parte o conceito e os requisitos para o funcionamento de um sistema de QoS baseado em políticas são também explicados.

O capítulo 4 apresenta nossa proposta de solução para resolver alguns dos problemas apresentados no capítulo 2 e algumas necessidades apresentadas no capítulo 3. Neste capítulo apresentamos a arquitetura de um terminal H.323 que possui a capacidade de se comunicar com um sistema de QoS baseado em políticas. Neste capítulo são descritos os principais módulos do terminal H.323 e como ele deverá funcionar quando estiver interagindo com as entidades de rede. Ainda no capítulo quatro são descritas as necessidades de implementação nas entidades de rede, para que o modelo possa funcionar de acordo como explicado no capítulo 3.

Finalmente no capítulo 5, são apresentadas as conclusões do trabalho. Neste momento apresentamos o que foi feito, a contribuição do trabalho, os benefícios e os resultados obtidos. Além disso, exploramos as limitações que o trabalho possui e possíveis melhoramentos. Também apresentamos, nesta parte do texto, comentários que visam orientar trabalhos futuros no melhoramento do modelo e otimização do protótipo desenvolvido.

2 O modelo H.323: entidades e comunicação multimídia

A recomendação H.323 descreve terminais e outras entidades que provêm serviços de comunicação multimídia sobre redes de pacotes, sem garantia de qualidade de serviços. Estas entidades oferecem serviços de comunicação em tempo real. Nestes serviços estão incluídos a transmissão de áudio, vídeo e dados.

Na verdade a recomendação é um guarda-chuva que determina os padrões a serem utilizados para sinalização, estabelecimento de sessões, controle de chamadas, gerenciamento de largura de banda, controle de admissão, codecs para transferência de áudio e vídeo e protocolos de transferência de dados.

No H.323, o suporte para transmissão e manipulação de sinais de áudio é obrigatório para todas as entidades, enquanto que o suporte para transmissão de vídeo e dados é opcional, mas quando suportado, todas as entidades devem ser capazes de encontrar um modo de operação comum para a troca destes sinais.

O que o H.323 realmente introduziu de novidade no mundo da comunicação visual foram os procedimentos de sinalização para estabelecimentos de chamadas, maneiras de escolher codecs comuns de áudio e vídeo e a capacidade de interoperação com terminais de outras tecnologias, como, por exemplo, terminais H.320 em redes ISDN e terminais H.310 em redes B-ISDN, uma vez que o conjunto de protocolos RTP/RTCP continua sendo utilizado como base para o transporte de dados isócronos em redes IP.

Embora o desenvolvimento inicial do H.323 estivesse focado exclusivamente no estabelecimento de conferências em uma rede local, a sua ambição aumentou no momento em que a Cisco e a Vocaltech se uniram para formar o Fórum de voz sobre IP (Voice over IP Forum), e escolheram o H.323 como protocolo padrão para a telefonia IP em ambiente WAN [HER 2000].

2.1 A série H.323 e seus componentes

Os elementos H.323 que a recomendação descreve são: terminais, gateways, gatekeepers, MPs, MCs e MCUs, que se comunicam através de fluxos de informações. Estes fluxos podem ser fluxos de áudio, vídeo, dados, controle de comunicação e controle de chamadas.

Os sinais de áudio devem ser digitalizados e codificados adequadamente, para redução da taxa de tráfego na rede. Uma vez codificados, os sinais são transportados através do protocolo RTP. O controle deste fluxo é realizado através do canal RTCP referente a fluxo. Da mesma forma, os sinais de vídeo devem ser digitalizados e codificados.

2.1.1 Terminais H.323

Terminais são entidades básicas definidas na recomendação H.323. Através dos terminais, usuários podem se comunicar verbal e visualmente além da possibilidade de comunicação através da transferência e compartilhamento de dados. Os terminais podem ser equipamentos de hardware ou um aplicativo em software, ou pode também ser um sistema completo para a troca de sinais de áudio, vídeo e dados. Pode ainda, ser um simples telefone com capacidade para transmissão de sinais de áudio. A Figura 2.1,

abaixo, ilustra os elementos presentes em um terminal, bem como quais destes elementos fazem parte do escopo da recomendação H.323.

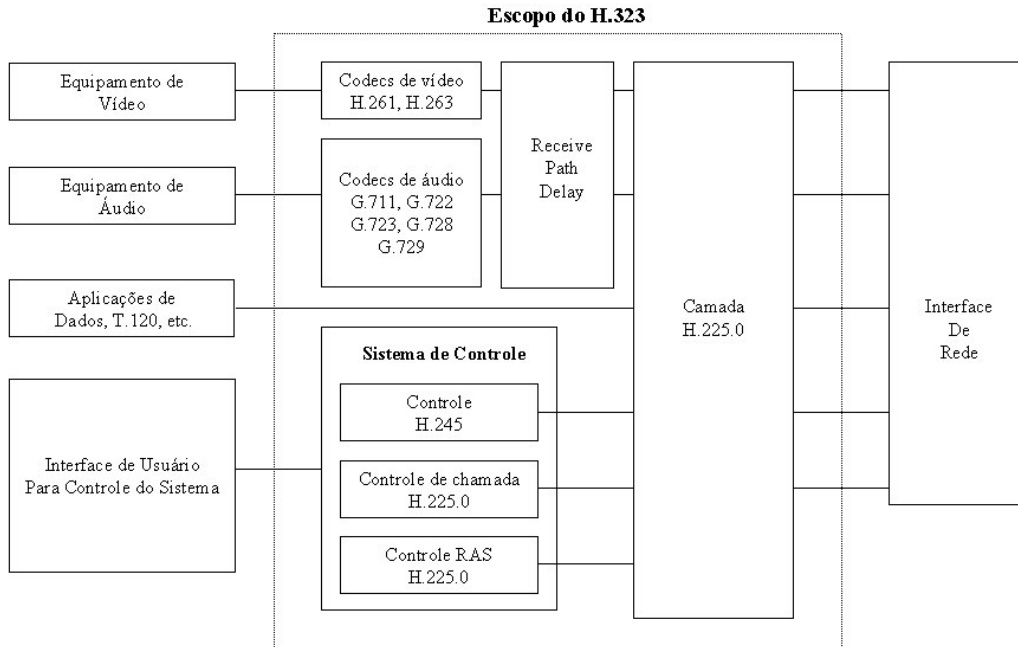


FIGURA 2.1 - Representação de um terminal H.323

O H.323 especifica os codecs de áudio (G.711, G.722, G.723, G.728, etc.), codecs de vídeo (H.261, H.263), canais para troca de dados (T.120, etc.), a unidade de controle do sistema e a camada H.225.0, que compõe um terminal H.323. Entretanto, a recomendação não define os dispositivos associados, isto é, dispositivos de áudio, vídeo, rede, aplicações e interfaces para o compartilhamento de dados.

Os terminais H.323 devem necessariamente oferecer suporte para áudio e por isto, devem implementar pelo menos um codec de áudio, o G.711 nos modos A-law e μ -law, entretanto o G.711 não pode ser usado para canais de banda estreita (< 56 Kbps). Para estes casos os terminais devem usar o codec G.723.1 ou o codec G.729. Opcionalmente, os terminais poderão codificar os sinais de áudio, segundo as recomendações G.722, G.728 e MPEG-1. A escolha de qual codec será utilizado durante a sessão, é feita logo após o estabelecimento da conexão através de mensagens de troca de capacidade. Estas mensagens estão definidas na recomendação H.245 e o procedimento é descrito posteriormente neste texto.

Um terminal H.323 deve ser capaz de operar de forma assimétrica, isto é, enviar áudio no formato G.711 e receber áudio no formato G.729, por exemplo. Os terminais poderão também enviar mais de dois canais de áudio ao mesmo tempo.

Os codecs de vídeo são opcionais, mas se suportados devem pelo menos codificar os sinais de vídeo segundo a recomendação H.261 no modo QCIF. Além disso, os terminais poderão codificar e decodificar os sinais em outras formas de operação do H.261 ou do H.263. Outros codecs de vídeo e formatos de imagem também

poderão ser utilizados, mas apenas mediante negociação através do H.245 na fase de troca de capacidades. Para sinais de vídeo é permitido o modo de operação assimétrico em relação à taxa de bit de vídeo, taxa de quadros e resoluções de imagem, permitindo que um terminal possa enviar imagens QCIF enquanto recebe imagens no formato CIF.

A troca e compartilhamento de dados, também são opcionais para os terminais H.323. A recomendação T.120 é utilizada para interoperabilidade com outros terminais H.323 e terminais H.324, H.320 ou H.310, embora aplicações ou protocolos não padronizados possam também ser utilizados, através de negociação na fase de troca de capacidades.

As funções de sinalização de chamada (H.225.0-Q.931), controle de chamada (H.245) e funções controle de registro, admissão e estado junto ao gatekeeper (RAS), também devem estar presentes em um terminal.

Os canais de controle de chamada H.245, são usados para carregar mensagens de controle fim-a-fim, isto é, de um terminal até outro terminal. Isto inclui os procedimentos e mensagens necessárias para a troca de capacidade, abertura e encerramento dos canais lógicos para troca de áudio, vídeo ou dados, mensagens de controle de fluxo entre várias outras mensagens adicionais. Este canal pode ser estabelecido entre dois terminais, entre um terminal e um MC, ou entre um terminal e um gatekeeper. Como as entidades H.323 (terminal, gateway, MCU ou gatekeeper) podem suportar várias conexões simultâneas, um canal de controle para cada sessão deve ser aberto.

Os canais de sinalização de chamadas utilizam as mensagens definidas na recomendação H.225.0, para estabelecer as conexões entre duas entidades H.323. Entidades podem ser terminais, gateways, gatekeepers e MCUs. Este canal é independente dos canais de controle H.245 e do canal RAS. O canal de sinalização H.225.0 é aberto antes do canal H.245, para realizar as operações de sinalização, estabelecimento e encerramento de sessões H.323.

O canal de sinalização RAS, também usa o formato de mensagens H.225.0. Entretanto, a funcionalidade destas mensagens é diferente da funcionalidade do canal de estabelecimento de chamadas. O canal RAS executa funções de registro (capacidade de saber que alguém está presente na rede ou pode ser contatado através de um certo terminal), admissão (chequear os direitos de acesso à recursos da rede) e manutenção do estado atual dos recursos da zona (monitoramento da disponibilidade de recursos de rede). Em redes onde a figura do gatekeeper não está presente, o canal de sinalização RAS não é utilizado, uma vez que as mensagens RAS são trocadas apenas entre terminais e o gatekeeper da zona.

A entidade mencionada como “Receive Path Delay” está presente com a finalidade de adicionar atraso aos fluxos de mídia, considerando o jitter na chegada dos pacotes. Os fluxos podem ser propositalmente atrasados nesta entidade, para manter a sincronização com outros fluxos de mídia presentes na comunicação. O atraso introduzido deve ser aplicado apenas para fluxos que estejam chegando (serão reproduzidos no terminal) e nunca para os sinais que estão partindo do terminal. Os terminais receptores podem usar esta facilidade para obter sincronia de lábio entre sinais de áudio e vídeo.

A camada H.225.0 está presente para garantir a conformidade das mensagens enviadas e recebidas dos canais de sinalização e do canal RAS com a recomendação H.225.0. Assim, esta camada é a responsável por formatar, no terminal, as mensagens H.225.0.

2.1.2 Gateways

Gateways são componentes opcionais em um sistema de conferência H.323. Eles provêm muitos serviços, sendo que o mais comum é a tradução de formatos de sinais entre terminais H.323 e terminais de outros tipos ou tecnologias. Isto é, ele deve prover a tradução apropriada entre modos de transmissão de tecnologias diferentes, por exemplo, formatos de transmissão H.225.0 de/para H.221, e procedimentos de comunicação H.245 de/para H.242.

O gateway também é o responsável pelo estabelecimento de chamadas em ambos os lados, o lado da rede IP e o lado da CSN (rede de comunicação por circuitos). A tarefa de tradução entre formatos de sinais de mídia (áudio, vídeo e dados), também pode ser realizada pelo gateway.

Terminais H.323 podem conversar entre si em uma mesma rede, sem o envolvimento de um gateway. Entretanto quando um terminal H.323 desejar conversar com uma pessoa que possui apenas um telefone comum, conectado na rede pública de telefonia (PSTN) ou um terminal H.320 em linhas ISDN, a comunicação, deverá necessariamente, passar por um gateway, que irá executar todas as traduções necessárias para que a comunicação aconteça. A sua função é refletir as características de um terminal na rede IP para um terminal na CSN e vice-versa.

Como ilustrado na figura 2.2, o Gateway pode executar funções de terminal H.323 ou MCU H.323. Nestes casos ele irá parecer para todos os terminais ou MCUs H.323 presentes na rede IP, como se fosse apenas mais um terminal ou MCU H.323 e a sua comunicação com estes dispositivos, ocorre através dos procedimentos H.323 definidos na recomendação.

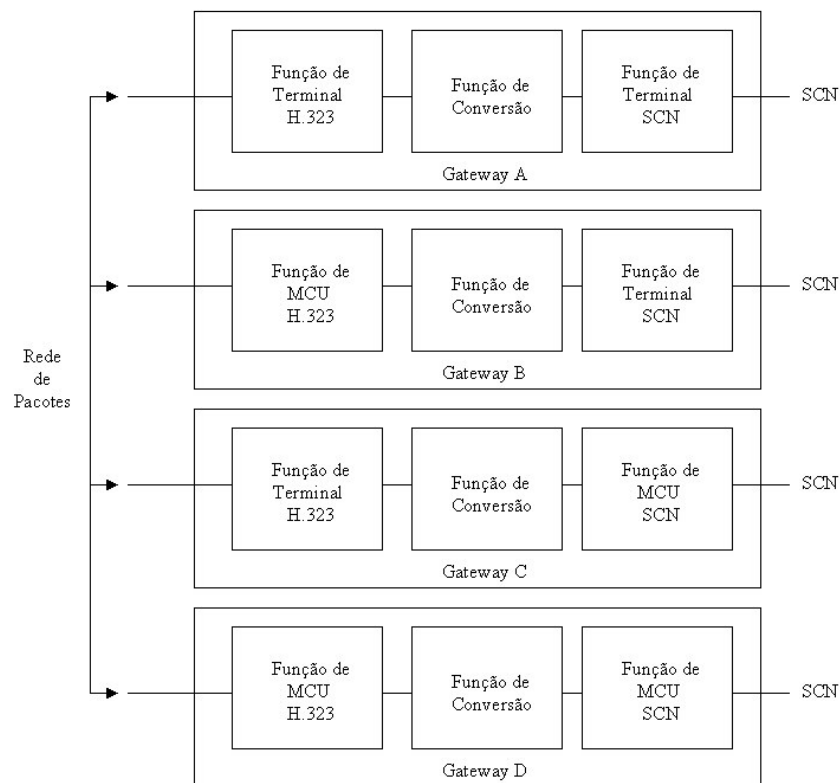


FIGURA 2.2– Configurações de um gateway

A função de conversão realiza todas as traduções necessárias para a interoperação dos terminais H.323 localizados no lado da rede IP com os terminais localizados na CSN. Os tipos de terminais presentes na CSN podem ser: H.310, H.320, H.321, H.322, H.324, V.70, GSTN ou terminais ISDN.

A função de terminal ou MCU no lado da rede CSN é vista pelos outros terminais desta rede como se ele fosse apenas um outro terminal do mesmo tipo (padrão). Este tipo deve, necessariamente, ser o mesmo do terminal que está participando da sessão. Os tipos de terminais são descritos nas seguintes recomendações: H.310, H.320, H.321, H.322, H.324, V.70, GSTN ou terminais ISDN.

O número de conexões simultâneas, conferências simultâneas e de terminais que podem se conectar ao gateway simultaneamente, são deixados a critério de cada fabricante, mas se o dispositivo que implementa o gateway, implementa também funções de MCU ele deve exercer rigorosamente as funções de gateway separadamente das funções de MCU. Da mesma forma, se ele implementar as funções de MCU para a CSN deve seguir as respectivas recomendações.

Embora atualmente os gateways ainda realizam todas estas funções de forma centralizada, a tendência é que gradativamente, as funções dos gateways sejam decompostas em entidades funcionais mais especializadas de forma a prover maior escalabilidade a rede de VoIP, conforme apontado por [LIU 2000].

2.1.3 Gatekeepers

O gatekeeper é o mais importante e complexo componente de um sistema H.323, embora sua presença não seja obrigatória para a realização conferências ou sessões H.323.

O gatekeeper introduz a noção de zona. Uma zona é a coleção de todos os terminais, gateways e MCUs, que estão sob o gerenciamento de um único gatekeeper. Uma zona inclui pelo menos um terminal e pode ou não incluir gateways e MCUs, mas contém um e apenas um gatekeeper. A zona não está amarrada a nenhuma tecnologia de rede e pode ser composta de múltiplos segmentos de rede interconectados por roteadores, switches ou outros dispositivos.

A figura 2.3, ilustra uma zona com todos os componentes: gatekeeper (GK), MCU, Gateway (GW), terminais (T1, T2, ..., T5). Como vimos estes dispositivos encontram-se separados em segmentos de redes distintos conectados por dois roteadores (R).

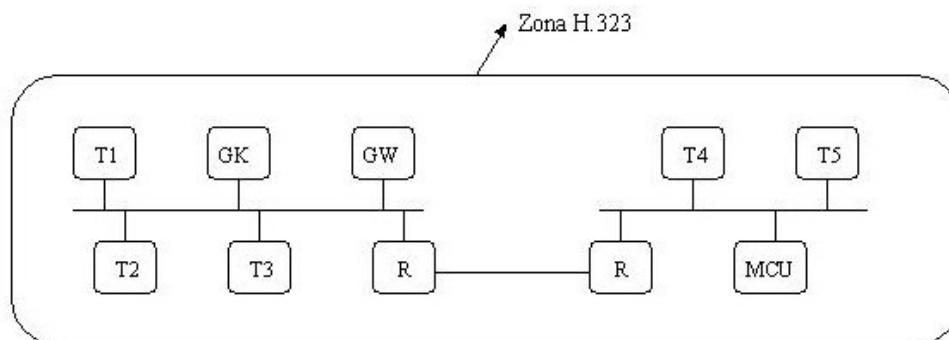


FIGURA 2.3 - Zona H.323

O gatekeeper provê serviços de controle de chamadas para os terminais H.323 presentes na zona. Ele é um componente logicamente separado dos terminais, gateways ou MCUs, embora a implementação possa coexistir com estes dispositivos, no mesmo equipamento.

O gatekeeper deve obrigatoriamente oferecer os seguintes serviços para os dispositivos H.323 presentes na sua zona:

- Tradução de endereços – Tradução de alias, para endereços de rede, isto é, quando você deseja chamar uma pessoa pelo seu endereço de e-mail, o terminal precisa saber qual é o endereço IP ou número de telefone na PSTN, correspondente ao terminal (ou telefone) desta pessoa. Isto deve ser realizado através de uma tabela que é atualizada através das mensagens de registro (RAS), embora outros métodos para a atualização desta tabela também sejam permitidos, como a utilização de um servidor de diretório, por exemplo.
- Controle de admissão – O gatekeeper é o responsável por autorizar (ou não) cada terminal a utilizar os recursos da rede. Se a rede estiver bastante carregada, pode ser necessário impedir que a conexão aconteça, por que não será possível garantir os recursos mínimos para o estabelecimento da sessão. Os parâmetros relacionados com a autorização podem estar baseados na quantidade de largura de banda, ou outros critérios como nível de QoS desejado para a sessão. Outra opção é permitir que todas as chamadas sejam aceitas, sem levar em consideração a qualidade de serviço destas chamadas. Atualmente, a maioria dos gatekeepers implementa o controle de admissão limitando apenas o número máximo de usuários que podem se conectar a zona e não utilizam qualquer tipo de mecanismo baseado em QoS.
- Controle de largura de banda – O gerenciamento de largura de banda verifica ou não se existe banda suficiente para um terminal utilizar na sessão de videoconferência, levando em consideração os codecs de áudio e vídeo escolhidos para a comunicação multimídia. A requisição pode ser aceita ou negada.
- Gerenciamento de zona – Todos os terminais que estiverem registrados com o gatekeeper, devem ser servidos por ele dentro da sua zona. O controle da zona é realizado pelo gatekeeper.

Além destes serviços o gatekeeper poderá oferecer as seguintes funções adicionais:

- Sinalização de controle de chamada – Em conferências ponto-a-ponto, o gatekeeper pode escolher processar ele mesmo os sinais de controle de chamadas H.225.0-Q.931 entre os terminais, ou alternativamente ele poderá avisar aos terminais para que troquem estas mensagens diretamente entre eles. Estas funcionalidades estão descritas com maiores detalhes na sessão 2.3.2.
- Autorização de chamadas – O gatekeeper poderá rejeitar uma chamada de um terminal, em razão de falhas de conexão, acesso restrito a certos gateways ou terminais e acesso restrito durante alguns intervalos de tempo, mas outros fatores e critérios de decisão podem também ser utilizados.
- Gerenciamento de Largura de Banda – Controle do número de terminais H.323 que podem ter acesso simultâneo à rede, se ele determinar que a largura de banda necessária não está disponível. Esta função também está presente durante uma chamada ativa, no caso do terminal requisitar mais banda.
- Gerenciamento de chamadas – O gatekeeper poderá manter uma lista das chamadas em andamento. Esta informação pode ser necessária para indicar que

um certo terminal está ocupado e para prover informações que serão utilizadas no controle de largura de banda.

Para oferecer a capacidade de prover conferência multiponto a partir de conferências iniciadas como chamadas ponto-a-ponto, o gatekeeper pode escolher receber os controles de chamadas H.245, de ambos os terminais. Quando a conferência mudar de ponto-a-ponto para multiponto, o gatekeeper poderá redirecionar, o canal de controle para um MC. É importante observar que o gatekeeper não precisa realizar nenhum tipo de processamento nas mensagens H.245, apenas roteia-las de um ponto ao outro.

Em redes que possuem gateways, o gatekeeper deverá estar presente para realizar a tradução de endereços E.164 para endereços de rede (p.ex.: endereços IP).

2.1.4 MC e MP

O controlador multiponto (MC) é responsável por funções de controle em uma conferência multiponto. O MC realiza a troca de capacidades com cada terminal que deseja participar da conferência. Isto significa dizer que conferências multiponto só irão acontecer na presença de pelo menos um MC.

Desta forma, o MC determina os modos de operação selecionados para a conferência, que pode ser comum para cada participante da conferência ou alternativamente diferente para alguns terminais (tipos de codecs de áudio e vídeo, transferências de dados ou outros recursos).

O MC pode estar localizado dentro de um gatekeeper, gateway, MCU ou terminal, como ilustrado na figura 2.4, entretanto, vale ressaltar que um MCU sempre possuirá um MC, que irá processar todos os canais de controle de todos os terminais envolvidos na sessão.

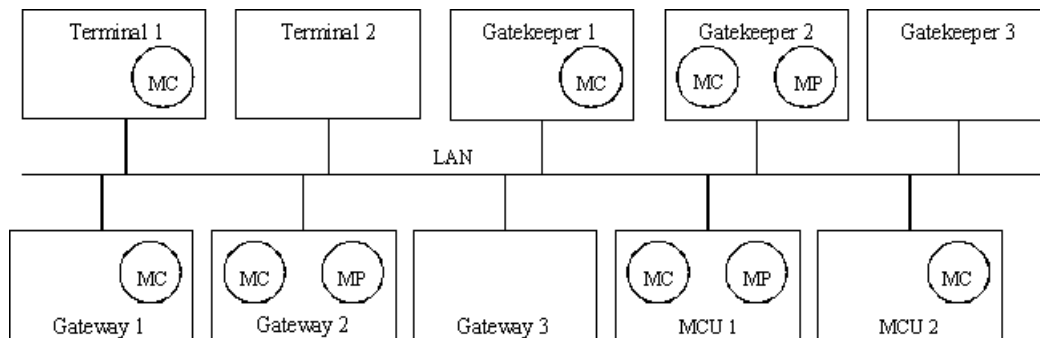


FIGURA 2.4 - Localizações possíveis para o MC e MP

Em uma conferência, quando existir mais de um terminal com capacidade de MC (como ilustrado na figura acima), estes terminais deverão utilizar a determinação “*master/slave*” (mestre/escravo), para escolher, qual será o MC ativo na sessão – que será o responsável por controlar a conferência. A escolha é sempre feita conforme os valores indicados pela tabela 2.1. O MC ativo deverá sempre utilizar o valor 240.

TABELA 2-1– Tipo de terminal H.323 para determinação mestre/escravo H.245

Tabela de valores do tipo de Terminal	Entidade H.323			
	Terminal	Gateway	Gatekeeper	MCU
Entidade sem MC	50	60	NA	NA
Entidade contém um MC mas nenhum MP	70	80	120	160
Entidade contém MC com MP de dados	NA	90	130	170
Entidade contém MC com MP de dados e áudio	NA	100	140	180
Entidade contém MC com MP de dados, áudio e vídeo.	NA	110	150	190
MC Ativo	240	240	240	240

O processador multiponto (MP) por sua vez recebe sinais de áudio, vídeo e dados, dos terminais participantes de uma conferência centralizada ou híbrida e realiza a agregação (“*mixing*”) ou comutação (“*switching*”) dos sinais, devolvendo o sinal processado para os terminais.

No processo de agregação o MP poderá atenuar ou eliminar alguns inputs de sinal, para reduzir qualquer ruído introduzido e assim devolver um sinal mais “limpo” para os pontos participantes da conferência. Para os sinais de dados, um MP deverá ser capaz de atuar como um MCS não folha ou como um MCS principal.

O serviço de comunicação multiponto (MCS) está definido nas recomendações T.122 e T.125 da série T.120 e não será discutido neste texto. O MCS é um serviço genérico, projetado para suportar a comunicação multiponto full-duplex entre um número arbitrário de entidades de aplicação T.120, sobre uma variedade de tecnologias de rede. Resumindo, é ele que provê a capacidade para troca e compartilhamento de dados, em uma conferência multiponto.

2.1.5 MCU

O MCU é um componente H.323 que provê a capacidade para suportar conferência multiponto. O MCU deve necessariamente consistir de um MC e zero ou mais MPs.

Um MCU que suporta conferências centralizadas consiste de um MC e um MP de áudio, vídeo e dados para a agregação de sinais de áudio, vídeo e dados. MCUs que suportam conferências descentralizadas consistem de um MC e um MP que implemente a recomendação T.120, usando capacidades de áudio e vídeo descentralizadas.

Um gateway pode implementar um MCU no lado interno da rede. Um gatekeeper poderá também conter um MCU. Em ambos os casos, as funções devem estar separadas e não podem interferir uma na operação da outra. Isto significa que, embora as funções de gatekeeper, gateway e MCU possam coexistir em um mesmo equipamento, elas devem ser implementadas separadamente e a ativação de um serviço, não implica na ativação do outro e vice-versa.

2.2 Procedimentos de sinalização de chamadas

O H.323 define também os procedimentos para o estabelecimento e controle de chamadas, mas utiliza o formato de mensagens definidas nas recomendações H.225.0 e H.245 do ITU-T respectivamente. Mensagens H.225.0 são usadas no processo de

estabelecimento e encerramento de chamadas e na comunicação RAS (mensagens entre dispositivos e o gatekeeper). Mensagens H.245 são usadas para carregar as informações de controle de chamadas.

O H.323 não impõe nenhuma restrição em termos de sincronização, nos procedimentos de estabelecimento de chamadas. Isto significa que um ponto A pode iniciar a comunicação de estabelecimento de chamadas para um ponto B no mesmo instante em que o ponto B inicia o mesmo procedimento com o ponto A. Em uma situação como esta, a aplicação deve saber como agir, como por exemplo, enviar um sinal de ocupado sempre que já existir uma mensagem de estabelecimento de chamadas em curso.

Um terminal H.323 deve suportar as seguintes mensagens para o estabelecimento de chamadas:

- *Setup*
- *Alerting*
- *Connect*
- *Release Complete*
- *Status Facility*

Outras mensagens definidas como: *Call proceeding*, *Status*, *Status Enquiry*, são opcionais.

A figura 2.5, mostra a captura de uma sessão, na fase de estabelecimento de chamada entre dois terminais H.323. Primeiramente, vimos duas mensagens que nada tem a ver com a sessão H.323. Logo em seguida inicia a comunicação entre os dois terminais: o primeiro possui endereço IP igual a 143.54.1.121, enquanto que o segundo conhecido como *tigresa.inf.ufrgs.br* possui endereço IP igual a 143.54.41.165.

Destination	Source	Protocol	Summary	Size	Tick (msec.)
tigresa.pgjie.ufrgs.br	143.54.1.121	UDP	Src Port = 49595 Dest Port = 49599	70	0
143.54.1.121	tigresa.pgjie.ufrgs.br	ICMP	Destination Unreachable	70	5
tigresa.pgjie.ufrgs.br	143.54.1.121	H.323 Call Signalling	TCP (SYN)	62	2615
143.54.1.121	tigresa.pgjie.ufrgs.br	H.323 Call Signalling	TCP (ACK,SYN)	62	0
tigresa.pgjie.ufrgs.br	143.54.1.121	H.323 Call Signalling	TCP (ACK)	60	0
tigresa.pgjie.ufrgs.br	143.54.1.121	H.323 Call Signalling	TCP (ACK,PSH)	60	0
143.54.1.121	tigresa.pgjie.ufrgs.br	H.323 Call Signalling	TCP (ACK)	60	180
tigresa.pgjie.ufrgs.br	143.54.1.121	H.323 Call Signalling, Setup	TCP (ACK,PSH)	280	0
143.54.1.121	tigresa.pgjie.ufrgs.br	H.323 Call Signalling	TCP (ACK,PSH)	60	90
tigresa.pgjie.ufrgs.br	143.54.1.121	H.323 Call Signalling	TCP (ACK)	60	190
143.54.1.121	tigresa.pgjie.ufrgs.br	H.323 Call Signalling, Alerting	TCP (ACK,PSH)	93	0
tigresa.pgjie.ufrgs.br	143.54.1.121	H.323 Call Signalling	TCP (ACK)	60	200
tigresa.pgjie.ufrgs.br	143.54.1.121	UDP	Src Port = 49595 Dest Port = 49599	78	2579
143.54.1.121	tigresa.pgjie.ufrgs.br	ICMP	Destination Unreachable	70	0
143.54.1.121	tigresa.pgjie.ufrgs.br	H.323 Call Signalling	TCP (ACK,PSH)	60	130
tigresa.pgjie.ufrgs.br	143.54.1.121	H.323 Call Signalling	TCP (ACK)	60	190
143.54.1.121	tigresa.pgjie.ufrgs.br	H.323 Call Signalling, Connect	TCP (ACK,PSH)	172	0
tigresa.pgjie.ufrgs.br	143.54.1.121	H.245	TCP (SYN)	62	5
143.54.1.121	tigresa.pgjie.ufrgs.br	H.245	TCP (ACK,SYN)	62	0
tigresa.pgjie.ufrgs.br	143.54.1.121	H.245	TCP (ACK)	60	0
tigresa.pgjie.ufrgs.br	143.54.1.121	H.245	TCP (ACK,PSH)	60	0
143.54.1.121	tigresa.pgjie.ufrgs.br	H.245	TCP (ACK,PSH)	60	0
tigresa.pgjie.ufrgs.br	143.54.1.121	H.245, Terminal Capability Set (request)	TCP (ACK,PSH)	583	0
143.54.1.121	tigresa.pgjie.ufrgs.br	H.245, Terminal Capability Set (request)	TCP (ACK,PSH)	583	5
tigresa.pgjie.ufrgs.br	143.54.1.121	H.245	TCP (ACK,PSH)	60	0
143.54.1.121	tigresa.pgjie.ufrgs.br	H.245	TCP (ACK,PSH)	60	10
tigresa.pgjie.ufrgs.br	143.54.1.121	H.245, Terminal Capability Set Ack (response)	TCP (ACK,PSH)	63	5
143.54.1.121	tigresa.pgjie.ufrgs.br	H.245, Terminal Capability Set Ack (response)	TCP (ACK,PSH)	60	0
tigresa.pgjie.ufrgs.br	143.54.1.121	H.245	TCP (ACK)	60	175
tigresa.pgjie.ufrgs.br	143.54.1.121	H.323 Call Signalling	TCP (ACK)	60	0
143.54.1.121	tigresa.pgjie.ufrgs.br	H.245, Master Slave Determination Ack (response)	TCP (ACK,PSH)	129	0
tigresa.pgjie.ufrgs.br	143.54.1.121	H.245	TCP (ACK,PSH)	60	20
tigresa.pgjie.ufrgs.br	143.54.1.121	T.120	TCP (SYN)	62	20
143.54.1.121	tigresa.pgjie.ufrgs.br	T.120	TCP (ACK,SYN)	62	0

Annotations in the image:

- Red box: Estabelecimento da conexão TCP, para mensagens H.225.0-Q931
- Blue box: Mensagens de estabelecimento de chamadas H.225.0-Q931: - Setup - Alerting - Connect
- Green box: Estabelecimento da conexão TCP para as mensagens de controle H.245.
- Green box: Mensagens H.245 para a troca de capacidade entre os terminais e posterior determinação mestre/escravo (quem será o MC ativo).

FIGURA 2.5 – Captura da de uma sessão H.323 mostrando a fase de estabelecimento de conexão.
(usando o Lan Explorer v.3.6 da Sunrise Telecom).

As primeiras mensagens mostram as três fases de estabelecimento de conexão TCP (three way handshake). Logo em seguida, é iniciada a comunicação H.323, conforme descrita na sessão 2.3. Conseguimos ver nesta captura as mensagens *Setup*, *Alerting* e *Connect*. Depois que a mensagem *Connect* chega até o terminal chamador (terminal que iniciou a chamada), é estabelecido um canal H.245 responsável pela troca de capacidade entre os terminais, bem como pela determinação mestre/escravo. Em seguida os canais lógicos de mídia devem ser estabelecidos e a comunicação de áudio, vídeo e dados poderão ser iniciados.

2.3 Fase de estabelecimento de chamada

Esta fase utiliza as mensagens de controle definidas na recomendação Q.931 do ITU-T. Além disso, a recomendação H.225.0 define que mensagens Q.931 são mandatórias na fase de sinalização de chamadas do H.323, por isso estas mensagens são chamadas de H.225.0-Q931. Entretanto, antes desta fase ser iniciada, toda a comunicação de registro e admissão com o gatekeeper – quando existir um gatekeeper presente na rede – bem como reserva de largura de banda devem ter sido realizadas previamente. Este procedimento será explicado na sessão 2.3.2.

O cenário mais simples da fase de estabelecimento de chamada envolve a comunicação entre apenas dois terminais H.323, sem a presença de qualquer gateway, gatekeeper ou MCU. Os terminais devem ser capazes de enviar mensagens *Alerting*. Se o terminal que está recebendo a chamada puder responder com qualquer uma das mensagens *Connect*, *Call Proceeding* ou *Release Complete* dentro de 4 segundos, a mensagem *Alerting* não é necessária, caso contrário ela deve ser enviada ao terminal chamador. A seguir, diferentes situações para o estabelecimento de chamadas são explicadas com a finalidade de oferecer uma visão completa dos procedimentos de comunicação H.323.

2.3.1 Estabelecimento de chamadas simples (sem registro com gatekeeper)

O cenário mais simples de uma comunicação H.323, acontece quando dois terminais se comunicam diretamente entre si, sem o envolvimento de qualquer outra entidade H.323.

Neste cenário, os dois terminais se comunicam diretamente como mostrado na figura 2.6. O terminal 1 envia uma mensagem *Setup* para o terminal 2. Se este conseguir processar e responder dentro de 4 segundos então a mensagem *Alerting* não é necessária.

A mensagem *Setup* é direcionada para a porta 1720 do terminal 2, como definida no apêndice D da recomendação H.225.0. O terminal 2 poderá responder com *Alerting*, *Call Proceeding* e *Connect*. A mensagem *Connect* deve possuir em um de seus campos o endereço de transporte (porta), que deverá ser utilizado posteriormente para o estabelecimento do canal de controle H.245, que se seguirá após a mensagem *Connect* ser processada pelo terminal 1.

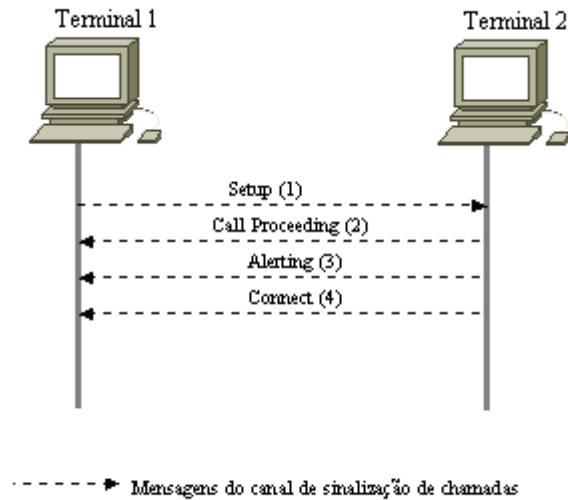


FIGURA 2.6 – Estabelecimento de chamadas básico. Sem Gatekeepers

2.3.2 Ambos pontos registrados com o mesmo gatekeeper

O próximo cenário ocorre quando um gatekeeper estiver envolvido na comunicação. Neste caso, o gatekeeper executa as funções de tradução de endereço, gerenciamento de admissão e gerenciamento de largura de banda. Diferentes situações podem ocorrer quando um gatekeeper estiver envolvido na comunicação: os dois terminais podem se registrar com o mesmo gatekeeper, apenas o terminal chamador se registra com o gatekeeper, apenas o terminal chamado se registra com o gatekeeper ou ainda os dois terminais se registram com diferentes gatekeepers. Em qualquer destes casos, o(s) gatekeeper(s) poderá(ão) ainda escolher entre proceder com o estabelecimento da conexão no modo roteado ou direto, conforme será explicado.

Para entendermos melhor os procedimentos de sinalização vamos primeiramente verificar como ocorre o registro de um terminal com um gatekeeper, através do canal RAS e como o gatekeeper conduzirá a fase de estabelecimento de chamada e controle, após os terminais terem se registrado com ele.

Para se conectar a um gatekeeper, primeiramente é necessário encontrá-lo na rede. Existem duas abordagens: a primeira é a configuração manual do gatekeeper. Este procedimento está fora do escopo da recomendação H.323. A segunda abordagem visa, encontrar o gatekeeper automaticamente na rede e então efetuar o registro com ele.

Para localizar o gatekeeper, o terminal poderá enviar mensagens multicast para o grupo 224.0.1.41, porta 1718 como definido na recomendação H.225.0. Outras formas também podem ser utilizadas, como broadcast e endereçamento manual. A primeira mensagem, *gatekeeper Request (GRQ)*, basicamente diz o seguinte: "Quem é o meu gatekeeper?". Após o envio desta mensagem, um gatekeeper presente na rede, possivelmente irá responder a chamada com a mensagem *gatekeeper Confirmation (GCF)*, indicando "Eu posso ser seu gatekeeper!". Esta mensagem deverá indicar o endereço de transporte (porta), onde deverão ser realizados os procedimentos de registro e admissão de chamadas no gatekeeper. Se o gatekeeper não puder aceitar a chamada responderá para o terminal com a mensagem *gatekeeper Reject (GRJ)*. Como mostrado na figura 2.7.

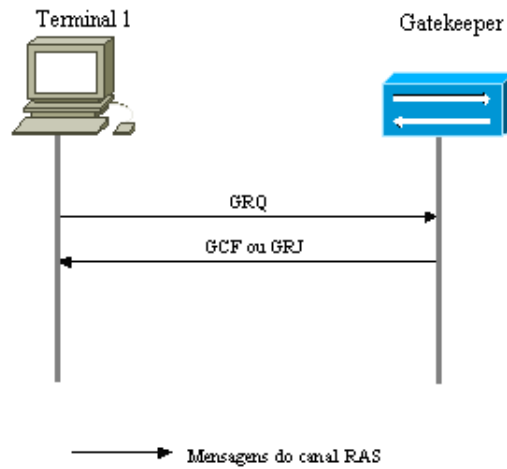


FIGURA 2.7 – Descobrimto automático do gatekeeper

Se mais de um gatekeeper estiver presente e responder ao terminal, o terminal poderá escolher àquele que melhor lhe convier, através de procedimentos não descritos na recomendação, como por exemplo, se registrar com o gatekeeper que respondeu primeiro à sua requisição. Entretanto, outras abordagens também podem ser utilizadas.

Após o descobrimento do gatekeeper, os terminais devem se registrar ao gatekeeper para se juntar à zona. O processo de registro com o gatekeeper deverá acontecer antes que as mensagens de estabelecimento e controle de chamada sejam iniciadas.

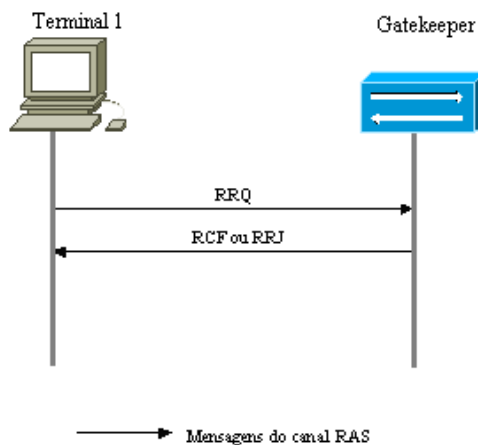


FIGURA 2.8 – Fase de registro com o gatekeeper

Um terminal deverá enviar a mensagem *Registration Request* (RRQ), requisitando autorização para se juntar à zona, como ilustrado na figura 2.8. Esta mensagem é enviada para o endereço e porta enviados anteriormente pelo gatekeeper no processo de descobrimento do gatekeeper. O gatekeeper poderá então responder com a confirmação do registro através da mensagem *Registration Confirmation* (RCF), ou rejeitar o registro através da mensagem *Registration Reject* (RRJ).

O gatekeeper poderá também, cancelar o registro do terminal a qualquer tempo, enviando uma mensagem *Unregister Request* (URQ), para o terminal. Neste caso o terminal deverá responder com a mensagem *Unregister Confirmation* (UCF). Este terminal deverá, procurar outro gatekeeper para se registrar antes de poder participar de uma nova conferência, exceto para chamadas ponto-a-ponto.

Quando o registro com o terminal é cancelado, ele passa a ser chamado de ponto não registrado (unregistered endpoint). Nesta situação, ele não faz parte da zona e não participará das funções realizadas pelo gatekeeper. Em outras palavras ele não poderá participar de nenhuma conferência através deste gatekeeper. Todo este processo está ilustrado na figura 2.9.

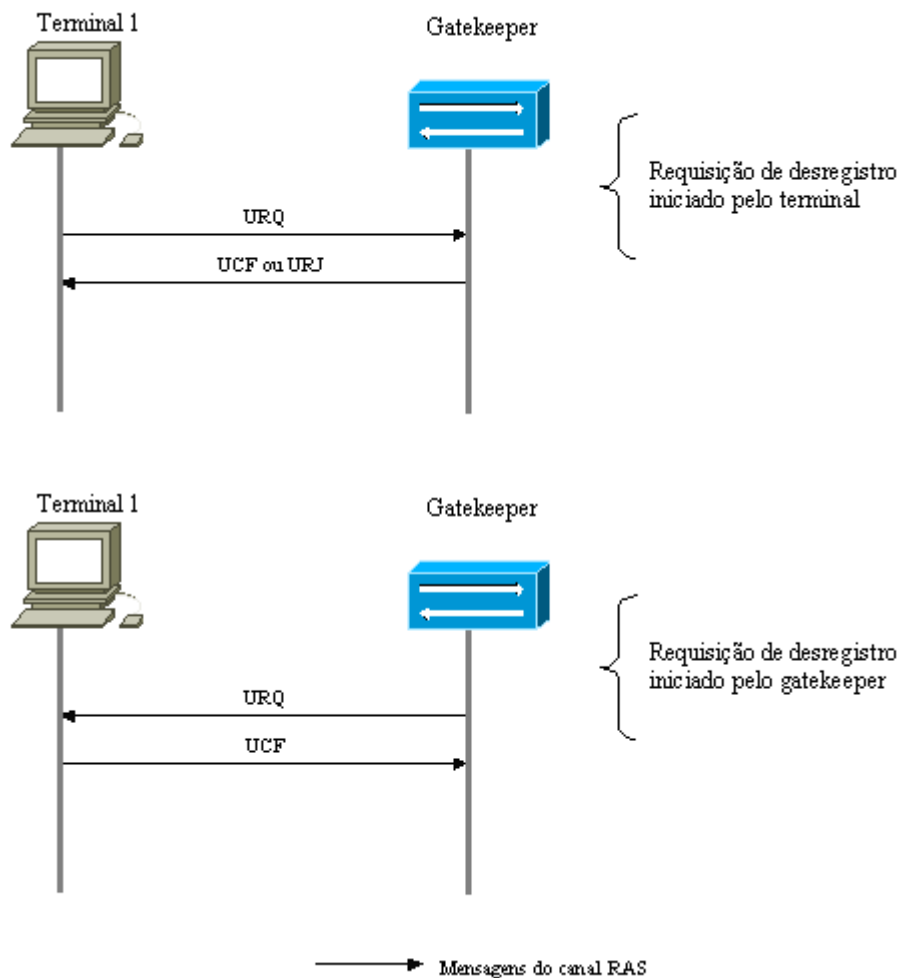


FIGURA 2.9 – Fase de desregistro com o gatekeeper

O canal RAS é utilizado também para troca de outras mensagens como: admissões, mudança na quantidade de largura de banda necessária, monitoramento do estado da chamada e desligamento com o gatekeeper.

Durante a fase de registro com o gatekeeper, ele (o gatekeeper) irá escolher como o processo de estabelecimento de chamadas irá ocorrer. O gatekeeper pode definir para os terminais que as mensagens de sinalização podem ser direcionadas de duas formas. A primeira possibilidade é o estabelecimento de chamadas direto, onde um terminal deverá enviar as mensagens H.225.0-Q.931 (*Setup*) diretamente para o outro

terminal no momento de efetuar os procedimentos de estabelecimento de chamada. A outra possibilidade é o estabelecimento de chamada roteado, onde as mensagens de estabelecimento de chamadas serão roteadas através do gatekeeper. Ambas formas utilizam as mesmas conexões, as mesmas mensagens com os mesmos objetivos, entretanto seus fluxos de comunicação são diferentes, conforme ilustrado nas figuras 2.10 e 2.11.

A figura 2.10 ilustra o processo de estabelecimento de chamadas no modo roteado. Neste caso o terminal 1 realiza a admissão junto ao gatekeeper (1) que irá aceitar ou rejeitar a admissão do terminal. No caso do gatekeeper rejeitar a admissão o terminal não poderá realizar a chamada (2 - ARJ). Se o gatekeeper aceitar a admissão do terminal 1 (2 - ACF), este irá enviar a mensagem *Setup* diretamente ao gatekeeper (3) que deverá roteá-la ao terminal 2 (4). Ao receber a mensagem *Setup*, o terminal 2 deve pedir permissão ao seu gatekeeper para poder alocar os recursos da rede e aceitar a chamada (5). No caso do gatekeeper rejeitar a admissão o terminal 2 não poderá aceitar a chamada (6 - ARJ). Se o gatekeeper aceitar a admissão do terminal 2 (6 - ACF), este irá responder ao terminal 1 com a mensagem *Connect* (7). O gatekeeper deve rotear a mensagem *Connect* até o terminal 1 (8).

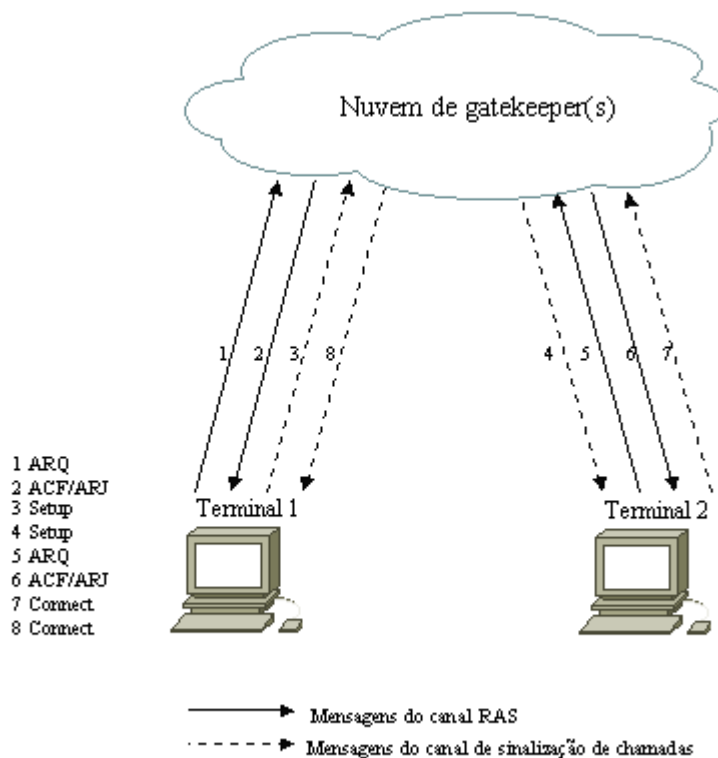


FIGURA 2.10– Estabelecimento de chamadas roteado pelo gatekeeper

A figura 2.11 ilustra o processo de estabelecimento de chamadas no modo direto. Neste caso o terminal 1 realiza a admissão junto ao gatekeeper (1) que irá aceitar ou rejeitar a admissão do terminal. No caso do gatekeeper rejeitar a admissão o terminal não poderá realizar a chamada (2 - ARJ). Se o gatekeeper aceitar a admissão do terminal 1 (2 - ACF), o terminal 1 irá enviar a mensagem *Setup* diretamente ao terminal 2, diferentemente do modelo roteado (3). Ao receber a mensagem *Setup*, o terminal 2 deve pedir permissão ao seu gatekeeper para poder alocar os recursos da rede, da mesma forma que faria no modo roteado. No caso do gatekeeper rejeitar a admissão o terminal

2 não poderá aceitar a chamada (5 - ARJ). Se o gatekeeper aceitar a admissão do terminal 2 (5 - ACF), este irá responder diretamente ao terminal 1 com a mensagem *Connect* (6).

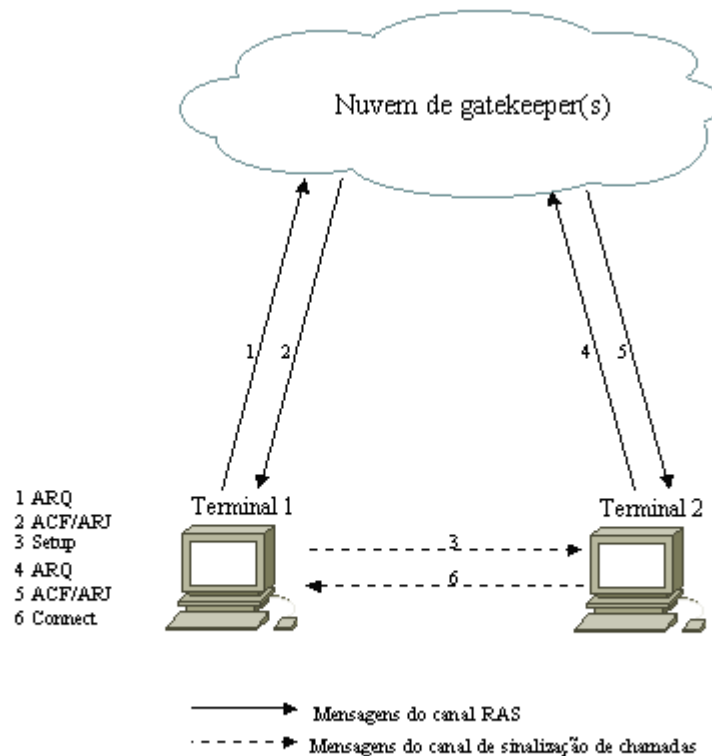


FIGURA 2.11 – Estabelecimento de chamadas direto

Quando é utilizado o modelo de estabelecimento de chamadas roteado, como ilustrado na figura 2.10, após o estabelecimento da chamada (canal H.225.0-Q.931) podem ainda, existir duas maneiras para o estabelecimento do canal de controle H.245: Este canal pode permanecer sendo roteado pelo gatekeeper ou poderá ser estabelecido diretamente entre os terminais participantes da conferência.

A versão 2 do H.323 definiu apenas o primeiro caso para o estabelecimento do canal de controle H.245 – o modo roteado – pois este modo de operação permite ao gatekeeper, o redirecionamento do canal de controle H.245 para um MC, quando uma conferência, rapidamente passar do modo ponto-a-ponto, para o modo multiponto. As fases para o estabelecimento do canal H.245 neste modo estão ilustrado a seguir:

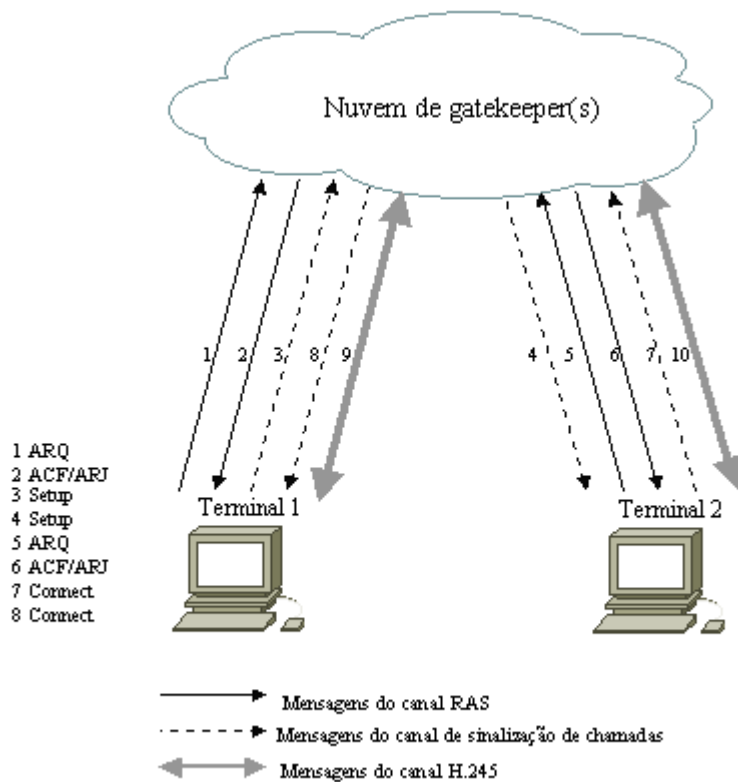


FIGURA 2.12 – Canal de controle no modo roteado pelo gatekeeper

Neste caso o procedimento é semelhante ao explicado no processo de estabelecimento de chamadas no modo roteado, entretanto, depois que a chamada foi estabelecida, o terminal 1 ou o terminal 2 podem iniciar o procedimento de estabelecimento do canal de controle H.245. Este procedimento não segue uma ordem pré-definida e portanto ambos terminais poderão iniciar este processo simultaneamente.

Agora, depois que entendemos como os procedimentos de registro com o gatekeeper acontecem e a seqüência em que são emitidas as mensagens entre os pontos participantes da conferência, vamos retornar ao processo de estabelecimento de chamadas.

Para explicar como ocorre este processo com a presença de um gatekeeper, vamos usar o caso onde os dois terminais se registram com o gatekeeper, no modo de chamada roteado, assumindo que as mensagens de descoberta do gatekeeper (GRQ) e registro com o gatekeeper (RRQ) já foram trocadas entre o terminal e o gatekeeper, conforme explicado acima. Em outras palavras, o terminal já sabe quem é o seu gatekeeper e já está registrado na zona.

Neste caso, o terminal 1 inicia a fase de admissão enviando a mensagem *Admission Request* (ARQ) (1) ao gatekeeper, pedindo permissão para iniciar uma chamada. O gatekeeper responderá com uma das mensagens *Admission Confirmation* (ACF) (2) ou *Admission Reject* (ARJ) (2). Se o gatekeeper resolver aceitar a chamada, sua mensagem ACF deverá conter o seu próprio endereço de transporte (porta), indicando que este é o local onde o terminal deverá direcionar as mensagens de estabelecimento de chamada (*Setup*). No modo direto o gatekeeper deve indicar o endereço e porta do terminal chamado, para que a troca de mensagens de sinalização possa ser realizada diretamente entre os dois pontos, sem a interferência (roteamento) do gatekeeper, ilustrado na figura 2.13.

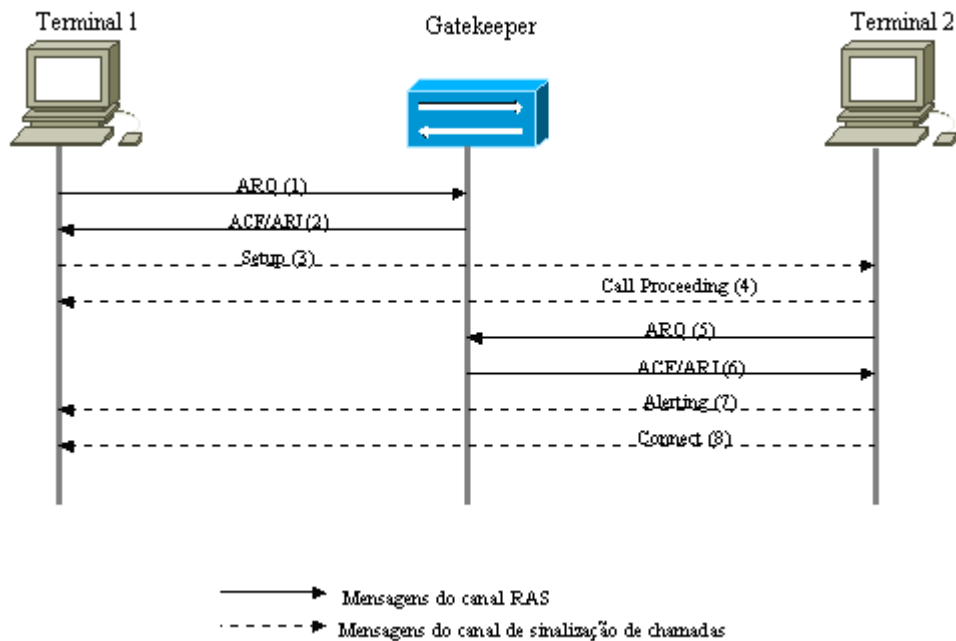


FIGURA 2.13– Dois pontos registrados pelo mesmo gatekeeper no modo de sinalização direto

O que notamos é que no modo roteado as mensagens de sinalização devem necessariamente passar pelo gatekeeper e no modo direto elas são trocadas diretamente entre os pontos finais.

Uma vez que o terminal 1 recebeu o endereço para onde deverá ser direcionado as mensagens de estabelecimento de chamada, ele inicia o processo enviando a mensagem *Setup* (3) para este endereço, neste caso o terminal 2. Se o terminal 2 resolver aceitar a chamada ele deverá se registrar com o gatekeeper através da troca de mensagens ARQ (5) e ACF/ARI (6).

Obtendo a resposta positiva do gatekeeper (ACF) o terminal 2 procede com a fase de estabelecimento de chamadas enviando ao terminal 1 a mensagem *Alerting* (7 – opcional) e a mensagem *Connect* (8).

No modo roteado – figura 2.14 – a primeira fase de conexão é igual ao modo direto. Entretanto quando o terminal 1 recebe como resposta o endereço para onde deverá ser direcionado as mensagens de estabelecimento de chamada, ele inicia o processo enviando a mensagem *Setup* (3) para o próprio gatekeeper, responsável por rotear esta mensagem para o terminal 2. O gatekeeper reenvia a mensagem para o ponto 2 (4). Se o terminal 2 resolver aceitar a chamada ele deverá se registrar com o gatekeeper através da troca de mensagens ARQ (6) e ACF/ARI (7).

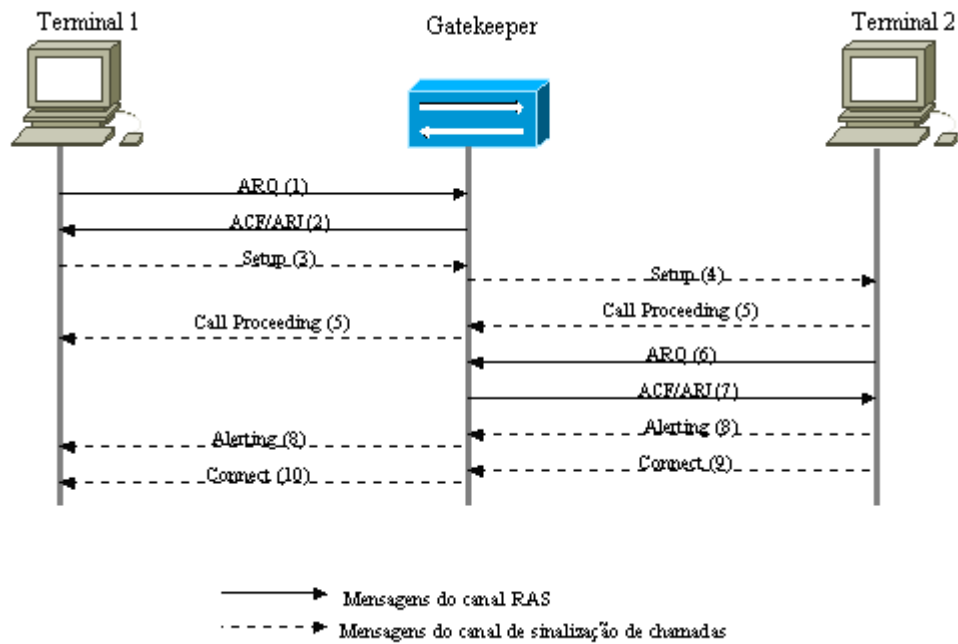


FIGURA 2.14– Dois pontos registrados pelo mesmo gatekeeper, no modo de sinalização roteado.

O terminal 2 responde então com a mensagem *Connect* (9), para o gatekeeper, que repassa a mensagem para o terminal chamador (10). Lembramos ainda, que apenas um dos terminais pode ter se registrado com o gatekeeper. Ou o terminal 1 ou o terminal 2.

2.3.3 Pontos registrados com diferentes gatekeepers

Existem ainda outras situações onde o registro com o gatekeeper ocorre de forma diferente. Esta situação é comum quando uma empresa irá se comunicar com outra empresa e ambas possuem sua própria zona H.323. A situação ilustra também, a possibilidade de uma empresa que possui uma zona H.323 utilizar serviços de outra empresa. Estes serviços podem ser a conexão com a rede pública de telefonia (PSTN), através de um gateway terceirizado com outra companhia.

Para explicar este cenário também usaremos o caso mais complexo onde os terminais estão registrados com diferentes gatekeepers e ambos escolhem rotear o canal de estabelecimento de conexão.

Neste cenário – ilustrado na figura 2.15 – o terminal 1, inicia os procedimentos de registro com o gatekeeper 1 ARQ(1) / ACF(2). Na mensagem ACF (2), o gatekeeper deve indicar o endereço onde deverá ser estabelecido o canal para sinalização de chamadas, neste caso o seu próprio endereço. Lembramos que poderia ser o endereço, do gatekeeper 2 ou ainda diretamente o endereço do terminal 2.

Como neste caso, o gatekeeper irá rotear o canal de estabelecimento de chamadas, o terminal 1 envia a mensagem *Setup* (3) para o gatekeeper 1, que redirecionará esta chamada para o endereço do terminal 2 (4), uma vez que este ainda não foi admitido pelo seu gatekeeper.

Se o terminal 2 deseja aceitar a chamada, ele inicia os procedimentos de registro com o seu gatekeeper (6). Se o gatekeeper aceitar o registro ele deverá retornar na mensagem ACF (7), o endereço de transporte no qual deverá ser estabelecido o canal para sinalização. Neste caso o seu próprio endereço. Neste momento o terminal 2 deve

avisar ao gatekeeper 1 que o seu gatekeeper deseja rotear a chamada, pois como ele não sabia disso anteriormente ele enviou a mensagem *Setup* diretamente ao terminal 2 (4). Ele faz isso enviando a mensagem *Facility* (8), que contém o endereço de transporte do gatekeeper 2.

O gatekeeper 1 envia então, a mensagem *Release Complete* (9), e reinicia o processo de estabelecimento de chamadas, através do gatekeeper 2, enviando a mensagem *Setup* (10), para ele. O gatekeeper 2 redireciona a mensagem *Setup* (11) para o terminal 2. A mensagem *ReleaseComplete* (9) é necessária, por que os gatekeepers decidiram rotear as mensagens de sinalização e agora a comunicação existente deverá ser encerrada e restabelecida no modo roteado, através dos dois gatekeepers.

O terminal 2 se registra novamente com o gatekeeper (12,13) e envia a mensagem *Connect* (15), para o seu gatekeeper, que redireciona a mensagem para o gatekeeper do terminal 1 (16), que finalmente recebe a aviso que sua chamada foi completada (17).

Devemos notar que a mensagem *Connect* (15), enviada pelo terminal 2 contém o seu endereço de transporte para o estabelecimento do canal de controle H.245, que é explicado na seção 2.4. A mensagem enviada pelo gatekeeper 2 (16), pode possuir ou o endereço do terminal 2, ou o endereço do gatekeeper 2, conforme sua escolha em estabelecer este canal no modo direto ou no modo roteado. A mensagem *Connect* (17) enviada pelo gatekeeper 1 ao terminal 1 pode possuir o endereço de transporte enviado pelo gatekeeper 2 (endereço do terminal 2 ou do gatekeeper 2) ou o seu próprio endereço conforme sua escolha em estabelecer o canal de controle H.245 no modo direto ou no modo roteado.

A troca de mensagens mostrando o processo de estabelecimento de chamada, está ilustrada na figura a seguir:

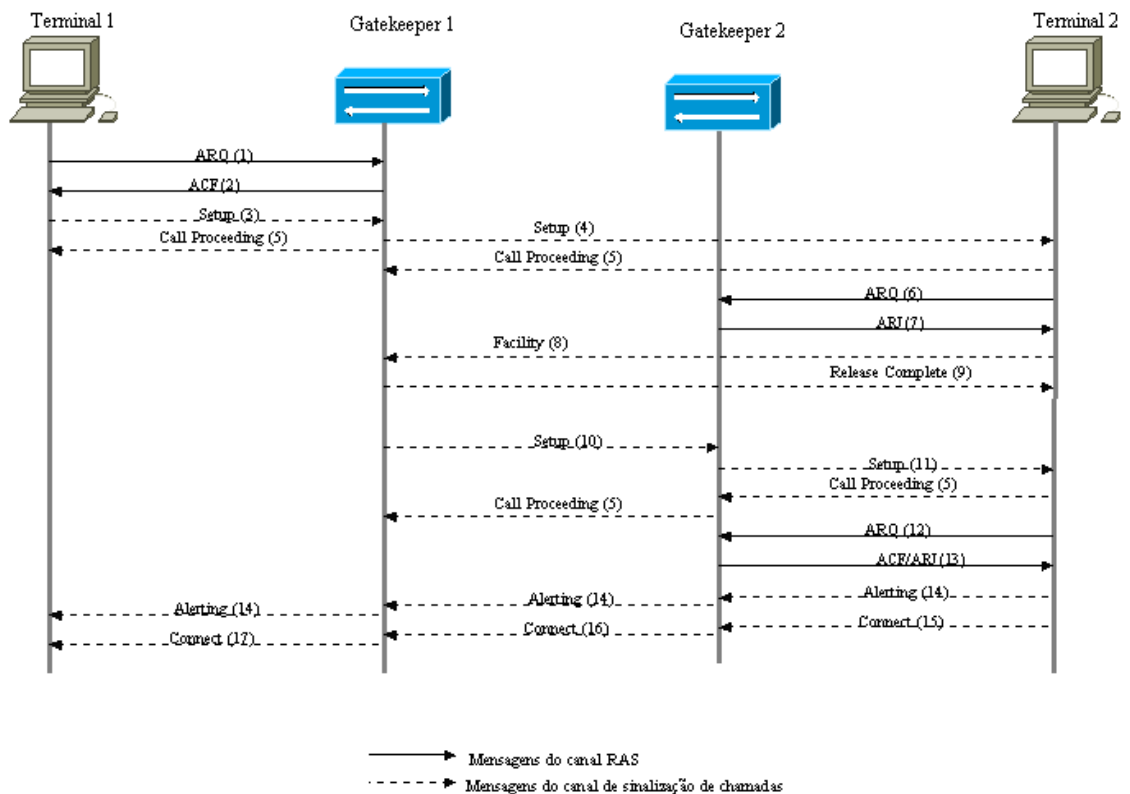


FIGURA 2.15– Dois pontos registrados em gatekeepers distintos, no modo de sinalização Direto/Roteado

2.4 Fase de comunicação inicial e troca de capacidades

Após a conexão ter sido estabelecida, isto é, após a mensagem Connect ser entregue ao terminal que iniciou a chamada, dá-se início a segunda fase de comunicação, antes que os terminais possam trocar sinais de áudio, vídeo e dados. Esta fase visa estabelecer o canal de controle H.245, que é utilizado para a troca de capacidade, determinação *master/slave* e abertura dos canais de mídia/dados.

A troca de capacidade é a primeira coisa a ser feita após a abertura do canal de controle. A primeira mensagem a ser enviada deverá ser necessariamente a mensagem *terminalCapabilitySet*, que indica quais são os codecs que o terminal possui para a troca de sinais de mídia. Após a escolha do modo de operação ter sido realizada, segue-se a determinação de qual dispositivo envolvido na conferência será o mestre e quem será o escravo. Em situações onde ambos os terminais possuem capacidade de MC, a determinação mestre escravo (*master/slave*) visa determinar qual MC será o MC ativo nesta conferência, conforme explicado na seção 2.1.4.

No caso destes dois procedimentos falharem, eles devem ser tentados novamente, duas vezes, antes dos terminais abandonarem esta fase e iniciarem a fase de encerramento da chamada como descrito posteriormente na seção 2.6.

2.5 Fase de comunicação multimídia

Depois da determinação das capacidades de cada terminal e da determinação de mestre/escravo, os procedimentos da recomendação H.245 devem ser seguidos para a abertura dos canais lógicos de transmissão de mídia (áudio, vídeo e dados).

Os canais de áudio e vídeo devem ser estabelecidos em endereços de transporte (portas) dinamicamente alocados e são identificados na mensagem *OpenLogicalChannelAck*. O emissor deverá usar o canal identificado nesta mensagem para enviar os sinais de mídia, que devem ser transportados através de canais não confiáveis, como o protocolo UDP.

Os dados transportados nestes canais devem trafegar por canais distintos e são unidirecionais exceto no caso do T.120.

Assim, quando o terminal 1 desejar abrir um canal de mídia com o terminal 2 ele deverá enviar a mensagem *OpenLogicalChannel*, para o terminal 2 que conterà o número daquele canal, os parâmetros para troca de mídia como, por exemplo, Áudio G.711. No caso de áudio e vídeo esta mensagem também deve conter o endereço de transporte (porta), onde o terminal 2 deve enviar os relatórios (RTCP RR) do protocolo RTCP. Logo que receber esta mensagem o terminal 2 deverá enviar ao terminal 1 a mensagem *OpenLogicaChannelAck*, que deve possuir o endereço (porta) na qual o terminal 1 deve enviar os dados, bem como o endereço (porta) na qual o terminal 2 espera receber os relatórios RTCP (RTCP SR).

Após os canais de mídia terem sido estabelecidos, a troca destes sinais pode ser iniciada pelos terminais envolvidos e a conferência é enfim estabelecida. Os sinais de áudio ou vídeo são enviados em pacotes RTP.

Os relatórios enviados nos pacotes RTCP permitem que o terminal 2 sincronize os diversos fluxos de mídia, bem como possa medir a distância que o terminal 1 está. Os relatórios RR enviados pelo terminal 2 permitem que o terminal 1 possa medir a qualidade de serviço da rede, podendo reduzir a taxa de transmissão, se necessário.

O H.323 estabelece que apenas um par de endereços de transporte (portas) RTP/RTCP, seja usado por sessão (sessão de áudio deve usar o mesmo endereço de

transporte RTCP tanto para fluxos do terminal 1 para o terminal 2 quanto do terminal 2 para o terminal 1).

2.6 Fase de encerramento da chamada

A chamada pode ser encerrada por qualquer um dos terminais envolvidos na conferência, entretanto, o procedimento para encerramento de chamadas é diferente, para conferências onde existe a presença de um gatekeeper e quando a figura do gatekeeper é inexistente (chamadas ponto-a-ponto).

Quando a chamada não envolve um gatekeeper, os seguintes procedimentos devem ser executados:

1. O terminal que deseja encerrar a chamada deve encerrar a transmissão de vídeo e então encerrar todos os canais lógicos para vídeo.
2. Deve descontinuar a transmissão de dados e então encerrar com todos os canais lógicos para transmissão de dados.
3. Deve descontinuar a transmissão de áudio e então fechar todos os canais lógicos para a transmissão de áudio.
4. Ele deve transmitir a mensagem H.245 *endSessionCommand*, no canal de controle H.245, indicando que ele deseja se desconectar desta chamada e encerrar o canal de controle H.245.
5. Deve esperar até receber a mensagem *endSessionCommand*, do outro terminal e então fechar o canal de controle H.245.
6. Se o canal de controle de chamadas estiver aberto, a mensagem *Release Complete* deve ser enviada e o canal deve ser fechado.
7. A chamada é encerrada.

Devemos notar que estes procedimentos são utilizados para encerrar uma chamada e não a conferência, que deve ser encerrada usando uma mensagem H.245 *dropConference*. Neste caso os terminais devem esperar até que o MC ativo termine com as chamadas como descrito acima.

Em redes que contém um gatekeeper, este deve estar ciente do encerramento da conferência para poder efetuar a liberação da banda, e conseqüentemente realocar esta banda em outras chamadas e conferências.

Neste caso, após executar os passos de 1 até 6 descritos acima, cada terminal deve enviar a mensagem H.225.0 Disengage Request (DRQ) (3), para o gatekeeper. O gatekeeper deverá responder com uma mensagem Disengage Confirm (DCF) (4). Depois de enviar a mensagem DRQ, os terminais não devem enviar mais nenhuma mensagem IRR não solicitadas pelo gatekeeper. O procedimento está ilustrado nas figuras 2.16 e 2.17.

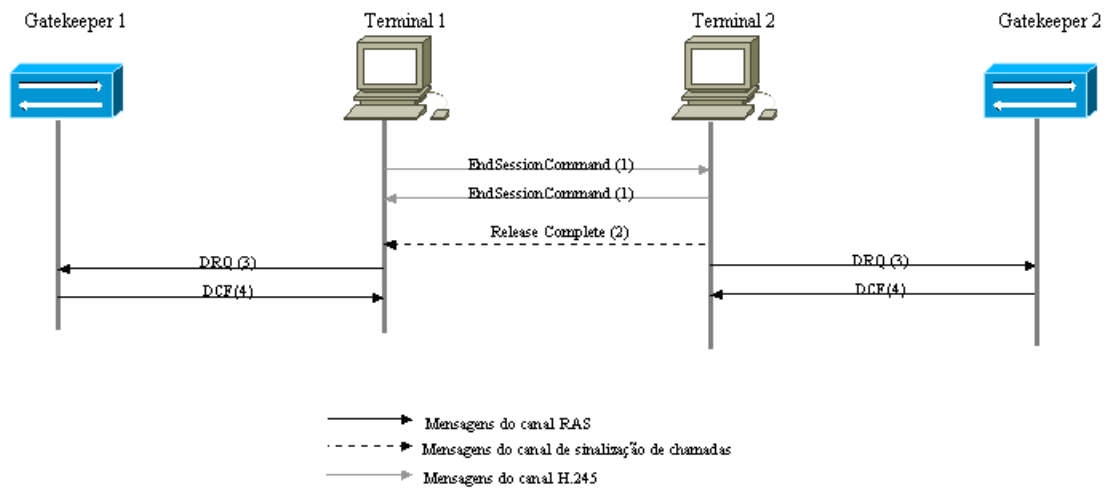


FIGURA 2.16 – Encerramento de chamada iniciado pelo terminal.

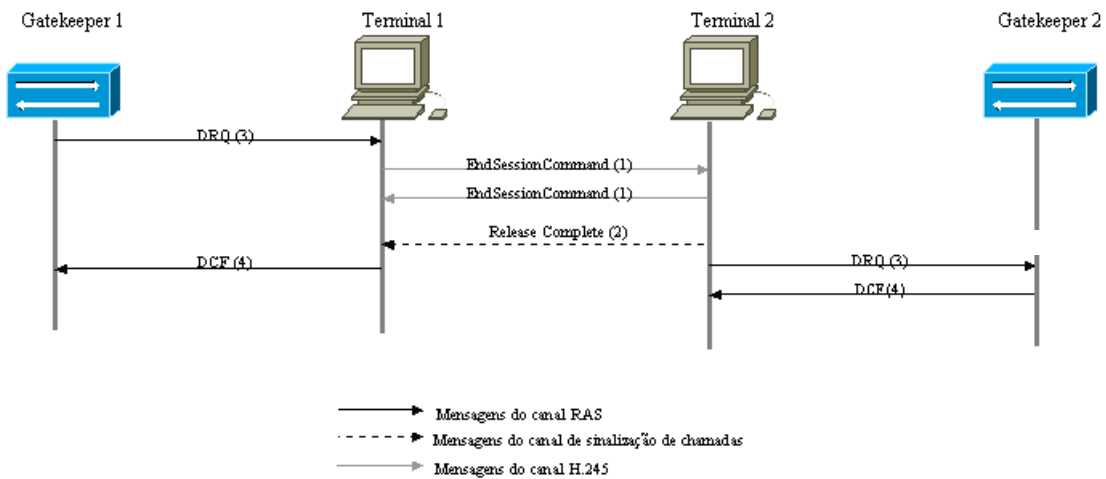


FIGURA 2.17– Encerramento de chamada iniciado pelo gatekeeper

Existe ainda, a situação onde a chamada é encerrada pelo próprio gatekeeper. Neste caso o gatekeeper envia a mensagem DRQ para o terminal. O terminal deverá então seguir os procedimentos de 1 até 6 e então responder ao gatekeeper com a mensagem DCF. O outro terminal após o recebimento da mensagem *endSessionCommand* deverá enviar uma mensagem DRQ ao gatekeeper que deverá responder com a mensagem DCF.

Se a conferência for multiponto o gatekeeper deverá enviar a mensagem DRQ para todos os terminais envolvidos de modo a encerrar com a conferência inteira.

É interessante mencionarmos que em muitas implementações reais do H.323, alguns terminais não são tão “educados” e simplesmente terminam com a conferência – através do comando *DropConference* – em vez de primeiramente encerrar com as chamadas.

A recomendação H.323, além de descrever e caracterizar as entidades que compõe um sistema H.323, definir as mensagens e procedimentos para o

estabelecimento e encerramento de conexões (H.225.0), comunicação com o gatekeeper (RAS) e controle de chamadas (H.245), como foi descrito até aqui, explica detalhadamente diversos outros pontos omitidos neste texto, como por exemplo, os diferentes modos de operação de uma conferência multiponto, a tradução de endereços no gatekeeper, a interoperação com terminais de outras tecnologias, o tratamento da privacidade e confidencialidade para obtenção de segurança, entre outras questões.

A opção por suprimir, neste trabalho, estas partes do H.323 visa manter a direção no objetivo do trabalho. Conhecer os procedimentos explicados anteriormente é requisito para o bom entendimento do restante do trabalho e por isso a recomendação H.323 foi explicada de forma resumida. Além disso, embora a grande maioria da bibliografia disponível, ainda sejam as próprias recomendações, já existem bons livros que tratam profundamente o assunto, como [HER 2000] e [KUM 2001].

2.7 Procedimentos adicionais para o estabelecimento de chamadas

2.7.1 Sinalização opcional usando a mensagem LRQ

A recomendação H.323 especifica ainda alguns modelos diferentes para o estabelecimento de chamadas, que devem ser utilizados em casos específicos. Vimos na sessão 2.3 e suas sub-seções, como são os procedimentos normais para o estabelecimento de chamadas, onde geralmente, uma mensagem inicial de *Setup* é enviada pelo terminal chamador para o terminal chamado ou para o gatekeeper.

Se o gatekeeper do terminal chamado quiser utilizar o modelo de chamada roteado pelo gatekeeper ele retornará o seu próprio endereço de transporte na mensagem ARJ. O terminal chamado usa a mensagem *Facility* para redirecionar a chamada para o seu gatekeeper. Todos estes procedimentos assumem que o terminal chamador ou o gatekeeper no qual o terminal chamador está registrado, apenas sabem o endereço de transporte do terminal chamado e não do gatekeeper do terminal chamado, o que pode ser observado na figura 2.15.

Se o gatekeeper do terminal chamador deseja utilizar o modo de chamada roteado, ele poderá enviar seu próprio endereço de transporte na mensagem LCF, o que permitirá o terminal chamador ou seu gatekeeper enviar diretamente a mensagem de Setup para o gatekeeper do terminal chamado, eliminando assim a necessidade de redirecionamento de chamada.

Um exemplo deste cenário é ilustrado a seguir. Neste cenário os dois pontos finais estão registrados com gatekeepers diferentes e ambos escolhem realizar a chamada pelo modo roteado.

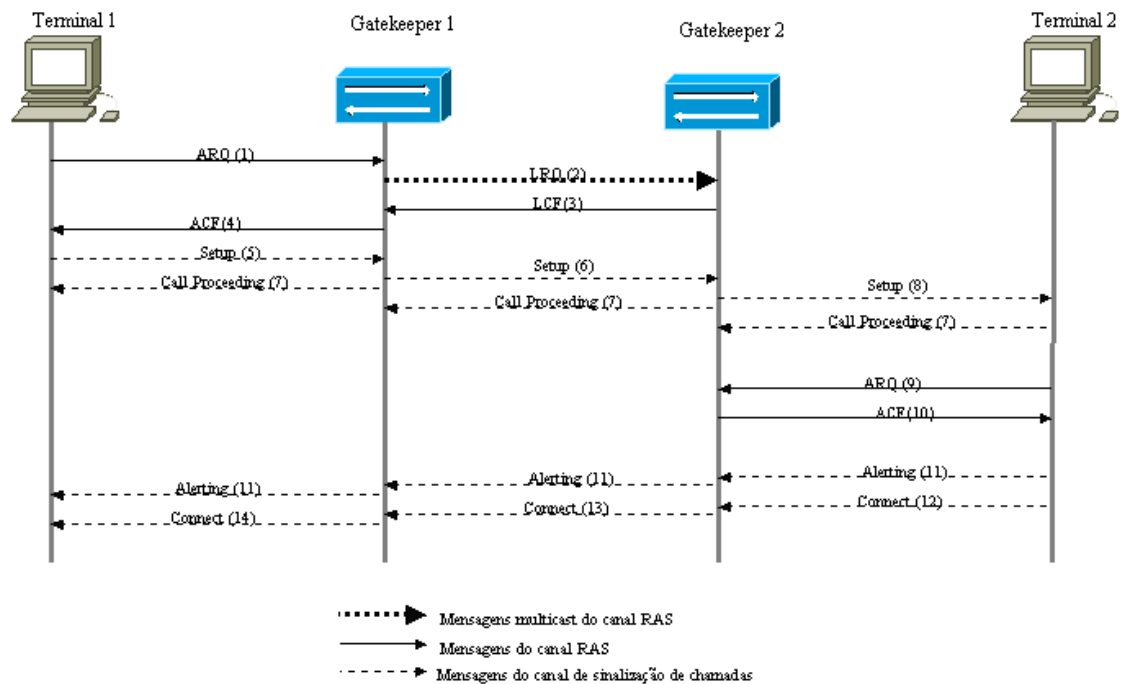


FIGURA 2.18 – Sinalização opcional usando a mensagem LRQ

Neste caso, o terminal 1 envia uma mensagem ARQ (1) para o seu gatekeeper (gatekeeper 1). O gatekeeper 1 envia via multicast uma mensagem LRQ, a fim de localizar o terminal 2. O gatekeeper 2 responde com a mensagem LCF enviando o seu próprio endereço de transporte, onde a conexão deverá ser aberta. Assim, o gatekeeper 1 irá enviar a mensagem de *Setup* (6) diretamente para o gatekeeper 2, que irá roteá-la até o terminal 2. (8). Neste ponto o terminal 2 inicia a fase de registro com o seu gatekeeper ARQ(9)/ACF(10). O terminal 2 responderá então com a mensagem *Connect* (12) para o seu gatekeeper que deverá conter o endereço de transporte no qual deverá ser estabelecido o canal de controle de chamadas H.245. O gatekeeper 2 redireciona a mensagem *Connect* (13), para o gatekeeper 1, esta mensagem pode conter o seu próprio endereço ou o endereço do terminal 2, dependendo se ele quiser estabelecer o canal de controle H.245 no modo roteado ou direto, respectivamente. O gatekeeper 1 envia então a mensagem *Connect* (14), para o terminal 1 contendo o seu próprio endereço de transporte ou o endereço de transporte enviado pelo gatekeeper 2, dependendo se ele quiser estabelecer o canal de controle H.245 no modo roteado ou direto, respectivamente.

O intuito de introduzir este mecanismo de conexão, é eliminar a sequência de *ReleaseComplete*, e restabelecimento de chamada visto na sessão 2.3.3.

2.7.2 Procedimento de conexão rápida

O H.323 v.2 resolveu um dos principais problemas contidos na primeira versão da recomendação: a possibilidade de se estabelecer rapidamente uma conexão. Mas na verdade o que isto quer dizer e qual a sua vantagem em relação aos métodos descritos anteriormente?

O problema existente com os procedimentos descritos anteriormente é que o tempo necessário para o estabelecimento da conexão, antes que se possa trocar sinais de mídia, é muito grande, pois envolve a troca das seguintes mensagens:

- Atraso de ida e volta da sequência ARQ/ACF;
- Atraso de ida e volta da sequência *Setup – Connect*;
- Atraso de ida e volta para as mensagens de troca de capacidades H.245;
- Atraso de ida e volta para as mensagens de determinação master/slave H.245;
- Atraso de ida e volta para as mensagens abertura de conexão para cada canal de mídia aberto

Além disso, não podemos esquecer que as mensagens H.225.0 e H.245 devem ser primeiramente estabelecidas através do protocolo TCP, isto requer o estabelecimento da conexão TCP o que implica um atraso adicional para o estabelecimento da cada uma das conexões H.225.0 e H.245.

O H.323 v.2 incluiu um método chamado de procedimento de conexão rápida que permite que conexões unidirecional e bidirecional, imediatamente após a mensagem de *Setup* e permitir comunicação de áudio imediatamente após a mensagem *Connect* ser recebida, melhorando o tempo de estabelecimento de conexões diminuindo o atraso envolvido.

Um terminal que decide usar o procedimento de conexão rápida deve incluir o parâmetro *fastStart* na mensagem *Setup*. Este parâmetro deverá incluir uma descrição de todos os canais de mídia que o terminal está preparado para receber e todos os canais de mídia que o terminal está preparado para enviar. Isto inclui os codecs utilizados, o endereço de transporte (portas) a serem utilizadas para cada conexão, etc.

Se o terminal chamado não implementa os procedimentos necessários para este tipo de sinalização rápido, ou não desejar realizar a conexão através deste método, ele não retornará o parâmetro *fastConnect* nas mensagens Q.931-H.225.0 subsequentes. Neste caso os procedimentos normais descritos na recomendação H.245, devem ser utilizadas, para troca de capacidades, abertura de canais de mídia, etc.

Este novo procedimento de conexão permite que a rede possa enviar qualquer tipo de mídia, para os endereços mencionados pelo terminal chamador antes que o terminal 2 tenha resolvido aceitar a chamada. Assim, mensagens como “A rede está congestionada, por favor, tente chamar mais tarde”, podem ser enviadas pelos dispositivos intermediários da rede ao terminal chamador antes do estabelecimento da chamada entre os dois terminais finais, coisa que não pode ser realizado através dos procedimentos normais de estabelecimento de chamada.

2.7.3 *Pregranted ARQ*

Até a versão 1 do H.323, um terminal necessitava enviar uma mensagem ARQ para um gatekeeper presente na sua zona, antes que pudesse realizar (fazer ou receber) uma chamada. Não era permitido enviar uma mensagem LRQ e uma mensagem *Setup*, isto é, a mensagem ARQ era necessária.

O H.323v2, definiu um modo excelente para otimizar o tempo de estabelecimento de chamadas, relacionadas a necessidade de admissão por parte do gatekeeper antes que um terminal possa realizar a chamada. Este mecanismo ficou conhecido como *PreGrantedARQ*.

Quando um terminal se registrar com um gatekeeper, o gatekeeper possui a capacidade de conceder previamente (*pre-grant*) requisições de admissão para o terminal. Isto permite ao terminal realizar e responder a chamadas mais rapidamente,

sem a necessidade de requisitar admissão para o gatekeeper. Esta capacidade ajuda a reduzir o tempo de estabelecimento da chamada, contribuindo para o aumento da qualidade percebida pelo usuário. Esta opção permite ao terminal [HER 2000]:

- Iniciar uma chamada sem enviar primeiramente uma mensagem ARQ.
- Iniciar uma chamada sem enviar primeiramente uma mensagem ARQ, mas somente se ele enviar a mensagem Setup para o gatekeeper (gatekeeper routed mode).
- Responder uma chamada sem primeiramente requisitar permissão ao gatekeeper.
- Responder uma chamada sem primeiramente requisitar permissão ao gatekeeper, mas somente a mensagem Setup for proveniente do gatekeeper (gatekeeper routed mode).

Para poder utilizar este mecanismo, o terminal precisa receber um indicativo do gatekeeper, na fase de registro com o gatekeeper. Neste caso, o gatekeeper irá usar o campo *preGrantedARQ* da mensagem *Registration Request* (RCF), indicando que o terminal possui permissão pré-concedida (pregranted) para receber e realizar chamadas.

2.8 Observações práticas e comportamento dos protocolos

2.8.1 Descrição do ambiente observado

O ambiente que compõe o sistema de colaboração visual observado, faz parte da infra-estrutura de TI do Centro de Processamento da Dados da Universidade Federal do Rio Grande do Sul. Os pontos que nos interessam deste ambiente é composto dos seguintes equipamentos:

- Roteador cisco 7507.
- Switch IBM GRS 8274.
- MCU da Meeting point sobre um servidor Pentium III 700 Mhz, 256MB RAM.

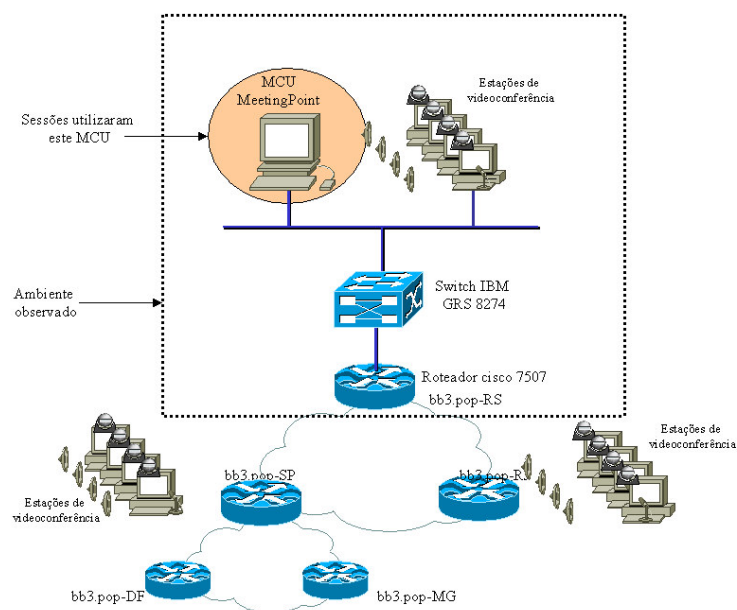


FIGURA 2.19 –Diagrama do ambiente observado

2.8.2 Problemas enfrentados

Durante a nossa pesquisa, procuramos monitorar diversas sessões de colaboração visual, para entender o comportamento dos protocolos utilizados pelo H.323. Participaram destas sessões de colaboração visual, alunos e professores de graduação e pós-graduação em diversas universidades do Brasil como: UFRGS, UFPE, Unicamp, Unijuí e UFBA.

Nestas sessões, todos os participantes estavam equipados com câmeras de vídeo e equipamentos de microfone, de modo que era possível a comunicação através de áudio, vídeo e dados entre todos os participantes.

O que conseguimos observar em relação a qualidade das sessões de colaboração visual, é que em algumas situações o MCU utilizado apresenta dificuldades em manipular a carga de tráfego solicitada. Os principais problemas apresentados foram os seguintes:

- Atraso – O atraso embora não fosse muito grande, foi perceptível, o que pode ser mitigado através de configurações de QoS apropriadas.
- Chamadas não completadas – Dependendo do número de participantes algumas chamadas não são completadas, afetando a qualidade percebida pelo conjunto de usuários como um todo.
- Desconexão – Em diversos momentos os usuários são simplesmente desconectados da sessão de colaboração visual.
- Cortes na voz – Em alguns momentos os sinais de voz de alguns participantes eram cortados, principalmente quando vários participantes falavam ao mesmo tempo.
- Bloqueio de envio de vídeo – Quando o terminal H.323 era configurado para usar largura de banda menor que 64 Kbps, ele era impedido de enviar sinais de vídeo, mesmo através de compressão H.263.

Destes problemas o que mais afetou o sistema observado, degradando de forma bastante grande as sessões de colaboração visual foram as sucessivas desconexões de usuários, causados principalmente pela falta de largura de banda. Neste sentido, é importante colocar que a ocupação dos canais de comunicação entre os pontos remotos possuíam carga de tráfego bastante grande, chegando ter uma taxa de ocupação constante superiores a 85% do canal, com picos momentâneos de 100% do canal.

Conforme mostrado no fragmento de log do MCU MeetingPoint – figura 2.20 –, quando o terminal de colaboração visual iniciava uma sessão T.120, e a sua conexão era encerrada devido a escassez de recursos de rede, mais especificamente escassez de largura de banda, o sistema inteiro, incluindo os sinais de áudio e vídeo H.323 eram encerrados, situação que degrada muito a percepção de qualidade do sistema, inclusive irritando os usuários participantes das sessões.

Também notamos que frequentemente, o servidor onde o MCU estava instalado, apresentava constantes picos de consumo de CPU iguais a 100%. Esta situação ocorria principalmente quando existia alta incidência de comunicação com voz, isto é, quando várias pessoas falavam ao mesmo tempo.

Outra situação interessante é o bloqueio de envio de vídeo, no caso do netmeeting ser configurado com largura de banda de 14400 bps ou 28800 bps. Neste caso, o MCU bloqueia o envio de vídeo pelo terminal H.323, entretanto permite que o mesmo receba sinais de vídeo. Quando o terminal é configurado para largura de banda 64 Kbps (ISDN/x.DSL) o MCU volta a permitir que o terminal envie vídeo.


```

Event> Mon Nov 26 17:15:54 2001 Pkts in 25655 Pkts
Event> client Leandro Bertholdo - T.120 session closed
Event> Mon Nov 26 17:16:54 2001 Pkts in 27438 Pkts
Event> Mon Nov 26 17:17:55 2001 Pkts in 1695 Pkts
Event> client Alexei Korb timeout -- holding down
Event> Mon Nov 26 17:18:55 2001 Pkts in 3324 Pkts
Event> client Alexei Korb - T.120 session closed due to insufficient bandwidth
Event> Mon Nov 26 17:19:56 2001 Pkts in 4708
Event> Mon Nov 26 17:20:56 2001 Pkts in 5850
Event> client Liane Tarouco - T.120 session closed due to insufficient bandwidth
Event> Mon Nov 26 17:21:57 2001 Pkts in 7114
Event> Mon Nov 26 17:22:58 2001 Pkts in 8182

```

FIGURA 2.20 - Log do MCU MeetingPoint mostrando encerramento da sessão

No cenário observado, chegamos a conclusão que a maior parte dos problemas de QoS existentes, está relacionado com as seguintes situações:

- Baixa capacidade de processamento do MCU.
- Alta ocupação dos canais de transmissão utilizados.

Assim, para resolver estes problemas, deve-se no mínimo, tomar duas ações bem específicas:

- Upgrade do Hardware onde o MCU está instalado.
- Utilização de mecanismos de QoS entre os terminais H.323 e o MCU, para otimizar a utilização dos canais de transmissão.

Como veremos nas próximas sessões, são diversos os motivos que podem influenciar na degradação dos sinais multimídia em sistemas de colaboração visual, por isso deve-se sempre se explorar o cenário macro, e verificar o que realmente pode estar impactando na qualidade dos sinais. Como exemplo, uma outra alternativa para o mesmo caso citado acima, seriam:

- Substituição do MCU em software por um dispositivo que implementa estas funcionalidades em Hardware.
- Aumento da capacidade dos canais de transmissão.

Como, a utilização de redes de pacotes têm em um dos seus objetivos, maximizar a utilização dos recursos de rede minimizando o custo de conectividade, acreditamos que uma solução baseada em QoS é a opção certa a se investir e assim veremos no capítulo seguinte, as necessidades de QoS para sistemas baseados no H.323, para depois explorar uma solução onde o terminal H.323 requisita automaticamente QoS para entidades de rede, com o objetivo de evitar que situações como esta ocorram.

3 O H.323 e os requisitos básicos de qualidade de serviço

Conforme visto nos capítulos anteriores, o H.323 foi concebido para operar em redes de pacotes que não oferecem qualidade de serviço. Entretanto, para que seja possível a utilização comercial em larga escala deste tipo de tecnologia, o aperfeiçoamento de alguns aspectos da tecnologia existente hoje devem ser considerados, incluindo a melhor utilização de mecanismos de QoS.

Neste sentido, cremos que as diversas pesquisas em desenvolvimento e o constante melhoramento dos padrões sobre os quais o transporte de sinais multimídia sobre IP acontece, permitirão que dentro de alguns anos estejamos realmente cobertos por uma infra-estrutura multimídia, completamente baseada no protocolo IP.

Se tivermos a capacidade de nos projetar alguns anos à frente, veremos redes de alta velocidade, tolerantes a falhas, usando o IP para entregar informações ao destino. Estes terminais deverão estar conectados aos backbones através de redes de banda larga na última milha, usando o estado da arte em mecanismos de qualidade de serviço que provavelmente entregarão sinais multimídia com maior performance e qualidade que as soluções atuais, com um custo muito mais atraente e justo.

Entregar sinais multimídia com qualidade até o usuário final requer desenvolver mecanismos que possam tratar adequadamente cada um dos diferentes tipos de sinais que compõe um sistema de colaboração visual. Esta tarefa não é fácil, pois requer empregar técnicas de QoS sofisticadas, além de tratar de uma série de outras questões como: Onde implementar as funcionalidades de QoS, na rede ou na aplicação? Quais tipos de problemas são prioritários? Como requisitar QoS da rede? Que estratégia de QoS deve ser adotada? Que fluxos de mensagens devem ser tratados com qualidade de serviço? Como manter o nível de qualidade entre diferentes provedores?

Todas estas e outras perguntas devem ser respondidas para criar uma estrutura funcional que ofereça qualidade de serviço fim-a-fim para usuários que desejam usar sistemas de colaboração tanto para a sua comunicação diária ponto-a-ponto, como para a comunicação multiponto no desenvolvimento de complexos projetos de negócios.

O grupo de trabalho TIPHON do ETSI, foi um dos primeiros grupos a estudar profundamente a questão de QoS direcionado a transmissão de sinais multimídia sobre redes IP. Este trabalho definiu dois pontos principais que devem ser tratados pelos sistemas de VoIP [EUR 2002a]:

- Qualidade do estabelecimento da chamada, que é determinada principalmente pelo tempo de estabelecimento de conexão percebida pelo usuário, influenciado por consultas a diretórios de dados, handshakes necessários para o estabelecimento de canais e outras ações que podem introduzir atraso.
- Qualidade da chamada, que é determinada principalmente pela degradação dos sinais de mídia (principalmente voz e vídeo), causado por codecs e perda de frames pela rede.

Particularmente os sistemas baseados no padrão H.323, consistem de três tipos de sinais que devem ser tratados adequadamente pela rede para prover a qualidade de serviço necessária ao usuário final. Estes sinais são: sinais para o estabelecimento de chamadas (H.225.0-Q931/RAS), sinais de controle de chamadas (H.245) e sinais de mídia. A qualidade de serviço percebida pelo usuário final está atrelada à qualidade das operações de cada um destes canais.

3.1 Qualidade de Serviço

Qualidade de Serviço (QoS) é a habilidade da rede priorizar determinados tipos de fluxos de informações que estão trafegando para oferecer melhores serviços aos usuários. Basicamente uma rede IP faz isso, aumentando a prioridade dos pacotes pertencentes a determinados fluxos ou limitando a prioridade de outros fluxos. Geralmente esta prioridade está relacionada à largura de banda dedicada, atraso e variação no atraso (jitter) controlados.

Para medir esta prioridade, geralmente os usuários estabelecem um acordo (contrato) com o provedor de serviço. Este contrato pode ter implicações legais e por isso deve ser estabelecido em cima de parâmetros mensuráveis. Este contrato deve possuir no mínimo duas partes:

- 1) A descrição do tráfego a ser enviado para rede
- 2) A descrição do nível de qualidade requerido.

Assim, o provedor deve garantir o nível de qualidade requerido, se e somente se, o tráfego enviado para a rede estiver de acordo com os parâmetros de tráfego estabelecidos. O descritor de tráfego caracteriza o a carga a ser demandada da rede. O descritor de QoS especifica a qualidade de serviço requerida do provedor. Alguns dos principais parâmetros para descrição do tráfego e descrição de QoS são:

Descrição do tráfego:

- a. Taxa de Pico de Dados – É a taxa máxima que o usuário está planejando enviar informações.
- b. Taxa Média de Dados – É a taxa média requerida, esperada que o usuário envia para rede, medida em um intervalo de tempo longo.
- c. Tamanho Máximo da Rajada – É o número máximo de pacotes que o usuário pode enviar a uma taxa igual a taxa de pico de dados, em uma rajada.
- d. Tamanho máximo do Pacote – É o tamanho máximo do pacote que o usuário está planejando enviar para rede.

Descrição do QoS:

- a. Taxa de Perda – Mede a fração de pacotes que não são entregues, ou são entregues tarde demais.
- b. Atraso Máximo – É o tempo máximo permitido para que um pacote saia da origem e chegue correto no destino.
- c. Variação Máxima do Atraso – Mede a tolerância máxima no atraso (Jitter).
- d. Taxa máxima de Erro – É a fração de pacotes que são entregues com erros (um ou mais bits errados).
- e. Banda Mínima – Especifica a quantidade mínima de largura de banda que o usuário aceita ter disponível.

Dependendo da característica de cada aplicação, um ou mais destes parâmetros devem compor o descritor de tráfego bem como o descritor de QoS. Esta combinação estabelece o nível de garantia de serviço que será oferecida ao usuário e geralmente definem uma categoria de serviço específica.

3.2 Mantendo a qualidade para sinalização e controle de chamadas

Em geral, três parâmetros de qualidade devem ser levados em consideração em relação aos canais de sinalização e controle de chamadas, [KUM 2001]:

- Probabilidade de completar a chamada.
- Tempo de estabelecimento da chamada.
- Acuracidade no estabelecimento da chamada.

Conforme vimos, o H.323 emprega diferentes mensagens e protocolos para realizar o estabelecimento, a sinalização e o controle de chamadas. Estes diferentes protocolos são: H.225.0-RAS para comunicação com o gatekeeper, o H.225.0-Q.931 para a sinalização e estabelecimento/encerramento de chamadas e ainda o H.245 para controle de chamadas. Cada um destes sinais representa diferentes estágios da comunicação H.323 e realizam diferentes operações devendo ser tratados de forma independente, para assegurar a manutenção dos parâmetros indicados acima.

O canal RAS é utilizado para a localização e registro com o gatekeeper, admissão de chamadas, negociação de largura de banda e posterior desconexão com o gatekeeper. Em redes onde o gatekeeper está presente, se ele falhar e não responder a requisição destes sinais, o sistema poderá ficar indisponível, pois os usuários não receberão permissão para realizar a chamada. Assim, verificamos que a disponibilidade do sistema é um aspecto chave a ser considerado no planejamento e implementação de sistemas H.323. Conforme especificado em [BSI 99] a disponibilidade é assegurar que usuários autorizados tenham acesso a informação e ativos associados, quando requeridos. Portanto, o planejamento adequado da capacidade do gatekeeper, links de comunicação e outros dispositivos críticos são essenciais para que a qualidade do serviço seja mantida apropriadamente. Por isso, servidores de alta capacidade com recursos apropriadamente dimensionados, que garantam redundância e contingência devem ser considerados.

Além da garantia da disponibilidade, o processo de sinalização através do canal RAS depende de outros fatores como: o mecanismo de descoberta utilizado (unicast ou multicast) e a performance de processamento das requisições pelo gatekeeper, que depende dos recursos computacionais do equipamento. Outro ponto importante é que quando a mensagem ARQ não é pré-concedida, o resultado é a introdução de atraso na fase de admissão de chamadas, o que deteriora a qualidade percebida pelo usuário final. Por isso, a comunicação com o gatekeeper deverá ser bem planejada de forma a garantir principalmente a disponibilidade do serviço e assegurar a pronta resposta do gatekeeper.

Em relação ao canal H.225.0-Q.931, que é usado para estabelecer e encerrar as conexões, a utilização de diferentes protocolos de transporte nesta fase (TCP ou UDP) pode determinar uma diferença bastante grande no tempo de estabelecimento da chamada, uma vez que o TCP introduz atraso no momento de estabelecimento da conexão (“*hand shake*”), devido à sua natureza orientada à conexão. A priorização destes sinais empregando mecanismos para minimizar o atraso é uma boa alternativa para acelerar o estabelecimento da conexão e melhorar a qualidade percebida. A utilização do UDP, embora não padronizada na versão 2 do H.323, parece ser uma saída muito tentadora para reduzir o tempo de estabelecimento da chamada. Entretanto, em redes de grande alcance geográfico, a utilização do UDP poderá introduzir perda de pacotes e em conseqüência deteriorar ainda mais a qualidade percebida. Nesta situação o estabelecimento de uma política de redundância pode ajudar a minimizar o problema.

A política de redundância ajuda um terminal na recuperação de pacotes perdidos. Esta política pode ser baseada simplesmente em copiar um frame duas vezes em pacotes IP consecutivos, para assegurar a chegada de pelo menos um frame de sinalização. Por outro lado, a utilização de redundância, poderá piorar ainda mais a situação, por que o destino precisará ajustar o seu buffer, de forma a receber todos os frames redundantes antes de transferir os frames para o decoder (introduzindo atraso). Portanto, a utilização de uma política de redundância ainda não foi padronizada no H.323, além de depender das características particulares do ambiente em questão, não sendo recomendado para redes espalhadas por uma grande área geográfica. Neste sentido a saída mais elegante e apropriada para assegurar maior qualidade até o momento, seria assegurar que os pacotes IP que carregam informações de sinalização H.225.0-Q.931, já trafegassem em um canal com QoS estabelecido.

O canal H.245 é utilizado para troca de capacidades, abertura e encerramento dos canais de mídia. Estes sinais também precisam ter o atraso minimizado. Mecanismos de QoS que possam assegurar a prontidão de entrega destas mensagens ao longo do caminho entre origem e destino devem ser empregados sempre que possível.

O ETSI através de um estudo [EUR 2002b], conforme comentado em [TOB 99], definiu quatro classes para classificação da qualidade de serviço em redes IP. Estes níveis de serviço ficam caracterizados por três parâmetros principais: qualidade da voz, atraso fim-a-fim e atraso para o estabelecimento da chamada, conforme mostrado na tabela 3.1:

TABELA 3-1 – Nível de serviço para o estabelecimento de chamadas em aplicações de VoIP

Nível de serviço percebido				
Parâmetro	Melhor	Alto	Médio	“Best effort”
Qualidade da voz (medida não interativa “one way”)	Melhor ou equivalente a G.711 para todos os tipos de sinais	Melhor ou equivalente a G.726 a 32 kbps para todos os tipos de sinais	Melhor ou equivalente a GSM FR para todos os tipos de sinais	
Atraso fim-a-fim	< 150 ms	< 250 ms	< 450 ms	
Atraso para estabelecimento da chamada (endereçamento IP direto)	< 1,5 s	< 4 s	< 7 s	
Atraso para estabelecimento da chamada (necessidade de conversão E.164 para IP)	< 2 s	< 5 s	< 10 s	
Atraso para estabelecimento da chamada (conversão E.164 para IP via clearing house)	< 3 s	< 8 s	< 15 s	
Atraso para estabelecimento da chamada (conversão e-mail para IP)	< 4 s	< 13 s	< 25 s	

Os membros do ETSI, acharam muito conveniente atribuir valores limites de atraso para a duração do estabelecimento de chamada e a tabela acima, foi estabelecida após muito debate entre os membros do grupo de trabalho, sendo atualmente considerado um acordo entre fabricantes e provedores de serviço.

Assim, embora existam diversos fatores e parâmetros que contribuem para determinar a qualidade de serviço em cada tipo de canal de sinalização existente nos sistemas H.323, a redução do tempo de estabelecimento da chamada, é um problema comum a todos os sinais e deve ser tratado adequadamente já no projeto de implantação

de sistemas de colaboração visual. O objetivo geral é atingir as métricas acima propostas e se possível reduzi-las.

3.3 A qualidade dos canais de mídia

É importante frizar que a percepção de qualidade dos sistemas de colaboração visual para os usuários finais, estão muito mais ligados aos canais de mídia do que aos canais de sinalização.

Podemos facilmente verificar isto, quanto tentamos ligar para alguém em um número de telefone celular. Em algumas situações devido à localização da pessoa (pode estar viajando – roaming), demora-se até 10 segundos para que seja enviado o primeiro sinal de chamada, ao telefone celular do destinatário. Mesmo assim, caso conseguirmos falar com a pessoa com uma qualidade boa, a nossa percepção de qualidade será igualmente boa.

Para os canais de mídia, essencialmente três características devem ser tratadas conforme apontados em [MAS 2000]:

- Minimizar o atraso unidirecional.
- Eliminar ou reduzir ao mínimo possível a variação no atraso (jitter).
- Eliminar ou reduzir a perda de pacotes.

Os dois primeiros parâmetros – atraso unidirecional e jitter – devem obrigatoriamente ser tratados quando existir a transmissão de sinais de tempo real, como áudio e vídeo. A perda de pacotes deve ser minimizada principalmente para as aplicações de colaboração de dados, com o intuito de evitar a retransmissão de pacotes perdidos ou errados.

Antes de qualquer coisa, precisamos entender como o atraso é introduzido, por que precisa ser adequadamente tratado e onde ele pode ser tratado. Devemos entender que existem, diversos pontos de inclusão de atraso na transmissão dos sinais de áudio e vídeo e que as causas do atraso podem ser tratadas em diferentes locais. A partir deste entendimento, é possível adotar uma solução adequada, diretamente nos locais onde eles acontecem.

Para possibilitar a transmissão de sinais de áudio ou vídeo através de uma rede digital, estes sinais devem ser digitalizados antes de serem transmitidos através da rede, a menos que já tenham sido produzidos no formato digital. Este processo inclui a conversão dos sinais analógicos em uma corrente de dados digitais que possam ser transmitidos em pacotes IP através da rede. Após a digitalização, o codificador (codec) tem a função de comprimir as amostras de dados digitais removendo as informações redundantes, minimizando a quantidade de informações duplicadas a serem transmitidas pela rede.

Uma vez codificados, os sinais são transmitidos através da rede. No destino ocorre o inverso. Neste caso, o decoder recupera as amostras digitais originais, a partir dos dados comprimidos enviados pelo codificador. Após isso, as amostras digitais são transformadas em sinais analógicos que serão reproduzidos no alto-falante ou monitor de vídeo do destino.

A figura a seguir ilustra os pontos sensíveis a introdução de atraso, na seqüência completa de transmissão de informações entre origem e destino:

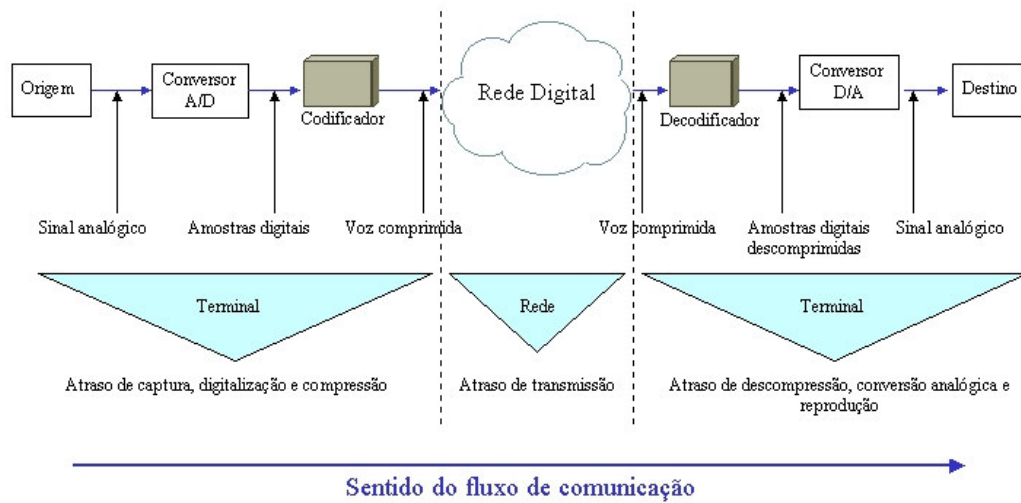


FIGURA 3.1 – Introdução de atraso para sinais multimídia.

Como podemos verificar, o atraso é introduzido em diferentes locais entre origem e destino. A compensação do atraso de captura, digitalização, compressão/descompressão e reprodução, bem como o atraso de transmissão introduzido pela rede de comunicação podem ser tratados através da utilização dos modelos e técnicas explicadas nas próximas sessões.

3.4 Adicionando qualidade de serviço em aplicações de Colaboração Visual

Como as questões relacionadas com a qualidade de aplicações de colaboração visual são muito amplas, foi reconhecido que para lidar com estes diferentes tipos de questões, uma única abordagem para o tratamento de QoS não seria suficiente [HER 2000]. Por isso, mecanismos de qualidade de serviço podem ser empregados em diferentes partes da solução, para minimizar a percepção de má qualidade por parte do usuário:

- Mecanismos de Qualidade de Serviço podem ser desenvolvidos em nível de aplicação/terminal.
- Mecanismos de Qualidade de Serviço podem ser desenvolvidos/configurados na rede IP.

Além disso, a sintonia adequada das aplicações, depende de diferentes condições da rede. Por exemplo, uma aplicação que precisa trabalhar com uma conexão discada de 28.8 Kbps via modem, precisa colocar múltiplos frames em um único pacote IP, enquanto que um terminal projetado para trabalhar em uma rede de 10 Mbps poderá minimizar o atraso, colocando apenas um frame por pacote IP. Estas particularidades tornam ainda mais difícil o estabelecimento de um padrão único para adoção de QoS em aplicações de Colaboração Visual.

3.4.1 A influência do sistema operacional

Conforme explicado anteriormente, a introdução de atraso de captura, compressão/descompressão, reprodução, etc. podem ser causados pela má configuração

dos terminais/aplicações, relacionadas ao ambiente total utilizado (dimensão do link, etc.).

A maioria das aplicações de colaboração visual, são programas de computadores normais, rodando em cima de um sistema operacional [HER 2000]. Neste cenário, todos os periféricos utilizados (microfones, caixas de som, vídeo câmera), são acessados através de APIs. Da mesma forma o acesso à rede de comunicação IP, é realizado através de APIs.

Como mostrado anteriormente, conforme os sinais vão sendo gerados – pelo microfone ou câmera de vídeo – as amostras destes sinais vão sendo armazenadas em um buffer de memória e alguma compressão é realizada no codec. Quando o buffer está completo, a placa de som, requisita ao sistema operacional – através de uma interrupção – que ele retire estes frames do buffer e os entregue a respectiva aplicação.

Estas interrupções param as atividades normais do sistema operacional e dispara um pequeno programa que é o responsável por avisar a aplicação que o buffer está completo e pode ser acessado pela aplicação. A aplicação irá, então, verificar se existem amostras suficientes no buffer para a realização da compressão. Se existir quantidade suficiente de amostras, para montar um frame completo (comprimido através de um algoritmo como o G.711, por exemplo) esta tarefa é realizada e o frame é despachado para a rede através da API de sockets.

Como as amostras de mídia são enviadas do microfone (ou câmera de vídeo) ao sistema operacional em “pedaços” e a maioria dos sistemas operacionais não consegue manipular muitas interrupções por segundo, o resultado é que uma quantidade mínima de atraso vai se acumulando em decorrência destas interrupções.

Para o sistema operacional Windows, por exemplo, muitos “drivers” de placas de som ou vídeo, tentam não gerar mais de uma interrupção a cada 60 ms. Isto significa dizer que nestes sistemas, as amostras são geradas em pedaços maiores que 60 ms, independente do codec utilizado. Por exemplo, um programa usando o codec G.729 pode gerar seis frames, enquanto que um programa usando o codec G.723.1 irá gerar apenas dois frames. Entretanto, para os dois casos o atraso introduzido até este estágio é de 60 ms, em razão da taxa máxima de interrupção do sistema operacional.

A conclusão é que o sistema operacional contribui para a introdução do atraso fim-a-fim em aplicações de voz sobre IP. Uma saída para esta situação é empregar sistemas operacionais de tempo real, que são desenvolvidos para manipular tantas interrupções quantas forem necessárias, para redução do acúmulo de atraso. Outra maneira encontrada por fabricantes de sistemas de VoIP é a utilização de hardware dedicado para a manipulação de funções de tempo real, tais como aquisição de amostras, compressão e empacotamento RTP.

3.4.2 Qualidade de Serviço em nível de aplicação

Mecanismos de qualidade de serviço no nível de aplicação são utilizados para tratar principalmente os atrasos introduzidos na origem e no destino, bem como para compensar eventuais falhas de transmissão e perdas de pacotes ocorridas durante a transmissão pela rede. Geralmente estes mecanismos de qualidade de serviço se aplicam muito bem aos sinais de mídia, mas raramente são utilizados para agregar qualidade de serviço aos canais de sinalização H.323.

O transporte de sinais multimídia em redes IP é geralmente realizado através do protocolo de transporte UDP, uma vez que o uso do TCP não se mostra apropriado, pelo atraso que introduz no momento de estabelecimento da conexão. Entretanto, o UDP, ao contrário do TCP, não provê nenhum mecanismo para garantir que os pacotes enviados

pela origem irão chegar ao destino. Para prover este tipo de garantia, as funcionalidades de re-ordenamento de pacotes, controle de fluxo e controle de congestionamento, devem ser implementados na aplicação, com o objetivo de introduzir controles apropriados ao sistema.

Assim, além de se preocupar com os atrasos introduzidos com a captura, compressão, descompressão e reprodução, o terminal precisa empregar técnicas de comunicação entre origem e destino, de modo a permitir que a origem se ajuste aos parâmetros determinados, sob demanda, pelo destino. Logo que o destino detecta problemas na recepção dos pacotes, ele deve notificar a origem sobre a taxa de pacotes perdidos, sobre o tempo de atraso ou congestionamento na rede. A partir destas informações, providas pelo destino, a origem deverá agir a fim de readequar a qualidade da transmissão. Em redes de pacotes como a Internet, foi identificado que quando a origem não reage apropriadamente às informações providas pelo destino em situações de congestionamento, o problema poderá piorar ainda mais [KUM 2001]. Por isso, é importante que a origem reduza a quantidade de informações enviadas para a rede enquanto ela encontra-se congestionada, voltando a transmitir em taxas maiores assim que a situação normal for reestabelecida.

Para sistemas H.323, alguns dos mecanismos que a origem pode usar para ajustar seus parâmetros de transmissão são [KUM 2001]:

- A origem poderá ajustar a largura de banda usada para a transmissão de mídias: Permissões devem ser concedidas pelo gatekeeper da origem, antes que ele possa aumentar sua taxa de transmissão. Quando codecs que permitem a escolha de múltiplas taxas de transmissão são utilizados, é possível mudar a taxa de output do codec. Alternativamente a origem poderá mudar para outro codec, com menor taxa de transmissão.
- Para codecs multicamadas, o número de camadas poderá ser ajustado: Se um codec de múltiplas camadas está sendo empregado, o número de camadas a ser usado na transmissão poderá ser ajustado conforme a informação de situação da rede gerada pelo destino. Se a rede estiver congestionada, o número de camadas transmitidas poderá ser reduzido.
- Técnicas de correção de erro de envio (FEC) podem ser empregadas: As técnicas de FEC diminuem o número de pacotes descartados no destino, aumentando o nível de qualidade dos sinais. Como o transporte de mídias requer transmissão em tempo real, técnicas de retransmissão são inadequadas enquanto que técnicas como FEC são mais apropriadas.
- Técnicas de retransmissão podem ser empregadas para o transporte dos sinais de sinalização via UDP: Os sinais de estabelecimento e controle de chamadas (H.225.0 e H.245) usam o TCP como protocolo de transporte. Nesta configuração, o tempo de estabelecimento da conexão, devido ao “*handshake*” do TCP, é bastante grande. Para diminuir o tempo de estabelecimento de chamadas, garantindo que eventuais perdas de pacotes sejam tratadas adequadamente, mecanismos de controles devem ser implementados em nível de aplicação.

As técnicas para obtenção de qualidade de serviço em nível de aplicação ajudam a melhorar o nível de qualidade percebido pelo usuário final, remediando algumas falhas existentes nos protocolos de rede e devem ser empregadas sempre que necessário.

A solução encontrada pelas aplicações H.323, foi transmitir os sinais de áudio e vídeo utilizando o conjunto de protocolos RTP/RTCP, que ajuda a realizar as funções descritas acima. O RTP e o RTCP permitem que o terminal possa realizar alguma

compensação de jitter introduzido pela transmissão da rede, além de fornecer informações sobre o estado atual de carga da rede, o que ajuda na escolha de medidas corretivas apropriadas, como por exemplo, a escolha de codecs de menor taxa ou na redução do número de camadas do codec, para codecs multicamadas.

Entretanto, como estas técnicas agem de forma isolada na origem e no destino, em vez de acompanhar o caminho completo da transmissão, elas são incapazes de agir sobre o modo como as mensagens são tratadas pela rede. Para controlar a manipulação dos pacotes nos elementos de rede intermediários ao longo do caminho entre origem e destino e prover qualidade de serviço fim-a-fim, técnicas de QoS em nível de rede precisam ser adotadas.

3.4.3 *Qualidade de serviço na rede*

Manter o controle da qualidade de serviço na rede é uma opção muito mais apropriada e poderosa do que empregar técnicas de QoS em nível de aplicação, por que as técnicas de qualidade de serviço em nível de aplicação não possuem nenhum controle sobre como a rede irá tratar os pacotes e por isso age apenas de forma reativa no caso de pacotes perdidos, atrasados, errados ou fora de ordem. Esta ação reativa provê um nível de qualidade de serviço limitado.

Em contra partida, quando técnicas de qualidade de serviço são aplicadas na rede, os dispositivos intermediários podem garantir um tratamento privilegiado para os pacotes de uma aplicação, ao longo do caminho, desde a origem até o destino, pois eles são capazes de diferenciar entre cada pacote, dando preferência para os pacotes que devem ser tratados de forma privilegiada. Além disso, pacotes pertencentes à mesma aplicação, mas que carregam diferentes tipos de sinais com diferentes requisitos de qualidade (áudio e vídeo, p.ex.) poderão ser tratados individualmente.

As técnicas de QoS em nível de rede elevam a qualidade dos sinais, além dos níveis de qualidade de serviço implementadas em aplicação, por que permitem estabelecer limites de atraso e perda de pacotes em cada dispositivo de rede ao longo do caminho entre origem e destino, característica impossível de se obter usando técnicas de QoS em nível de aplicação.

É importante deixar claro que, para implementar QoS em nível de rede com eficácia, cada um dos dispositivos de rede no caminho entre origem e destino, devem implementar as técnicas de QoS e todos eles devem estar configurados de tal forma que os pacotes que saem de um roteador sejam tratados com políticas de QoS que assegurem um tratamento semelhante no próximo roteador.

Quando o caminho entre origem e destino, passa por diferentes provedores de serviços (ou domínios), é possível que, as técnicas de QoS escolhidas não sejam as mesmas. Neste caso, deve necessariamente existir uma maneira de mapear a classe de serviço utilizada por um provedor, para uma classe de serviço semelhante utilizada em outro provedor. Não existindo um acordo de nível de serviço entre os provedores, não será possível garantir a qualidade do sinal durante todo o trajeto entre origem e destino. O acordo de nível de serviço (SLA), permite a implementação de diferentes níveis de qualidade de serviço entre diferentes provedores, podendo estender o alcance dos serviços de Telefonia IP e colaboração visual com qualidade, por uma grande área metropolitana, um país inteiro ou mesmo diversos países ao redor do globo.

Em redes IP, existem dois modelos principais para o provimento de qualidade de serviço: o modelo IntServ (serviços integrados) que utiliza o RSVP como protocolo de sinalização entre os elementos presentes na rede, e o DiffServ (serviços diferenciados).

As aplicações de colaboração visual são notáveis candidatas a utilização destes mecanismos, para oferecer qualidade de serviço aos usuários. Para que isso seja possível, é necessário que estas aplicações estejam aptas a especificar o nível de qualidade de serviço desejado e requisitem este nível de serviço para a rede.

3.4.3.1 O modelo IntServ

O modelo IntServ [BRA 94], foi o primeiro sistema bem estruturado de QoS, proposto para redes IP. No modelo IntServ, cada fluxo unidirecional de dados entre duas aplicações é tratado separadamente pelos elementos de rede, o que permite que cada fluxo de comunicação (áudio, vídeo e dados) entre os terminais participantes da comunicação, receba tratamento de QoS diferentes.

A RFC 1633 [BRA 94], propõe dois elementos fundamentais que compõe a estrutura IntServ: o próprio modelo IntServ e uma referência para implementação deste modelo.

O modelo IntServ define dois tipos de serviços: serviço garantido e serviço de carga controlada. O primeiro negocia limites para atraso fim-a-fim e assegura a largura de banda especificada, para os fluxos que obedecem às especificações. O segundo serviço é semelhante ao tradicional serviço “*best effort*”, quando a rede encontra-se com baixa carga de tráfego.

A referência para implementação do modelo, é um guia que define os componentes necessários para implementação do modelo IntServ. Esta referência é formada por quatro componentes:

- Escalonador de pacotes – O escalonador de pacotes deve ser implementado onde os pacotes são enfileirados e é o responsável por gerenciar o encaminhamento dos diferentes pacotes para o próximo dispositivo de rede usando um conjunto de diferentes mecanismos de filas que permitem a diferenciação dos diversos fluxos de dados de acordo com o QoS definido para cada um dos fluxos.
- Rotina para controle de admissão – Este componente, decide se o roteador (ou terminal), permitirá que novas requisições de QoS, serão ou não aceitas. Este algoritmo permite verificar se novas requisições irão ou não impactar as reservas já realizadas. O algoritmo de controle de admissão deve ser consistente com o modelo de serviço utilizado e é parte integrante do controlador de tráfego.
- Classificador de pacotes – É o responsável por mapear quais pacotes pertencem a que fluxo de dados e então, classificá-los em uma mesma classe para que recebam o tratamento adequado. A escolha de mapear um pacote em uma determinada classe está baseada no conteúdo do cabeçalho do pacote ou outros parâmetros encontrados em cada pacote.
- Protocolo para reserva de recursos – Uma aplicação IntServ utiliza este protocolo para requisitar a reserva de recursos (largura de banda, limite de perda de pacotes, etc.) para os roteadores, ao longo do caminho até o destino. Uma vez criado o caminho entre origem e destino, o protocolo é utilizado para manter as reservas ativas (através de renovações periódicas), em cada dispositivo de rede por onde o fluxo passa. O protocolo de reservas de recursos utilizado é o RSVP.

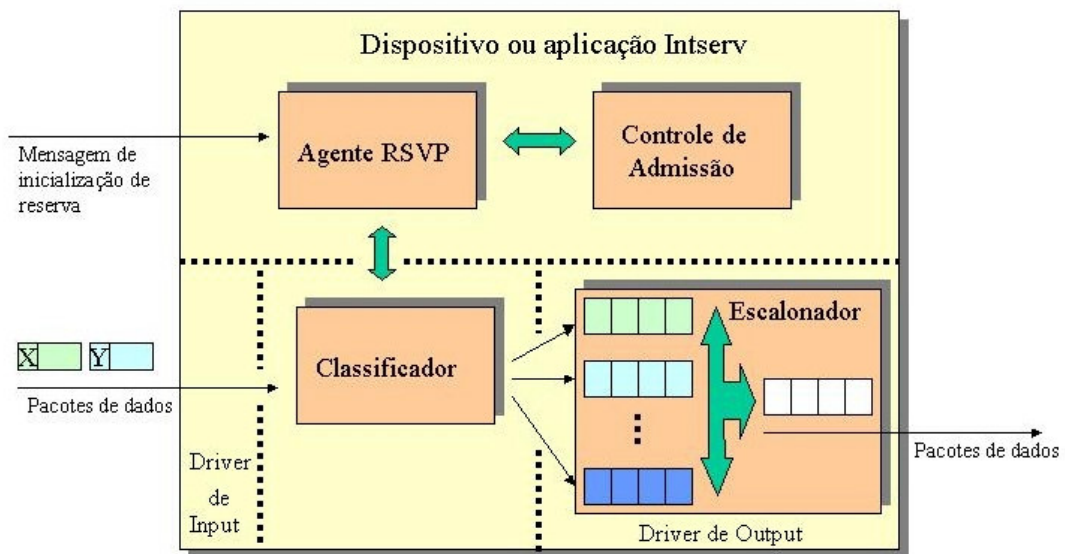


FIGURA 3.2 – Os componentes básicos da arquitetura Intserv.

Entretanto, existem alguns fatores, que dificultam o uso do IntServ, como a estrutura de QoS preferencial para utilização em grande escala [CIS 2001]:

- Cada dispositivo da rede, entre a origem e destino, incluindo servidores e estações fim, devem implementar o RSVP para poder requisitar QoS para os dispositivos de rede.
- As reservas de recursos nos dispositivos de rede que implementam QoS na prática, não são permanentes, isto é, devem ser renovadas periodicamente, adicionando carga de tráfego à rede e aumentando a possibilidade da reserva expirar no caso dos pacotes de sinalização destinados à renovação, serem perdidos.
- A manutenção dos estados de cada fluxo de comunicação em cada um dos roteadores, adiciona complexidade nos dispositivos de rede, além de aumentar os requerimentos de memória dos mesmos.
- Como o estado de cada reserva (para cada fluxo), deve ser conhecido em cada um dos dispositivos, a manutenção de milhares de estados de fluxo na rede, pode ocasionar um problema de escalabilidade conforme o tamanho da rede.

Outro ponto interessante, comentado em [WYD 2002], é que quando a capacidade da rede chega a seu limite e se usa a combinação RSVP/InteServ, novos usuários serão rejeitados, como forma de manter as garantias de serviços para os usuários já admitidos anteriormente, pois não existe recursos disponíveis para garantir o nível de QoS requisitado por todos. Entretanto, a satisfação dos usuários rejeitados também deve ser levada em consideração. Neste sentido, a qualidade de serviço oferecida pelo modelo Intserv é também limitada.

3.4.3.2 O Protocolo de Reserva de Recursos (RSVP)

O RSVP [BRA 97], é um protocolo, originalmente concebido para o estabelecimento de reserva de recursos em dispositivos de rede que implementam o

IntServ [BRA 94]. Este protocolo é utilizado pelos terminais, para requisitar qualidade de serviço para os fluxos de comunicação dados, vídeo ou voz, oferecidos pela rede.

Através deste protocolo os roteadores solicitam para todos os dispositivos de rede no caminho entre origem e destino, que reservem os recursos necessários para oferecer a qualidade de serviço requisitada pelo terminal.

O RSVP é um protocolo de transporte simplex, projetado para operar com o IPv4 e IPv6. Por ser um protocolo simplex, ele trata cada sessão de forma independente e portanto, diferentes reservas terão de ser realizadas para cada sessão. Uma sessão é caracterizada por um fluxo de dados (áudio, vídeo ou dados), que possui um destino e protocolo de transporte específico, portanto caracterizado pela tupla [Endereço de Destino, Id de Protocolo, Porta de Destino].

Para realizar uma reserva de recursos para um fluxo de dados, o terminal deve enviar uma mensagem *PATH*, a qual contém a especificação de fluxo (largura de banda, delay máximo permitido, etc..). Esta mensagem é enviada para o Endereço e porta de destino específicos. Esta mensagem passa por cada dispositivo de rede entre origem e destino. Em cada dispositivo fica armazenado, um estado de caminho que inclui no mínimo, o endereço do dispositivo anterior. Este estado é mantido nos roteadores intermediários para garantir que o fluxo de reserva possa voltar pelo mesmo caminho pelo qual foi iniciado.

Como o RSVP foi projetado de forma a deixar o destino como responsável por requisitar a reserva de recursos, para um determinado fluxo de comunicação, o destino, após receber a mensagem *PATH*, enviada pela origem, deverá enviar de volta uma mensagem *RESV*, responsável por criar e manter o estado de reserva em cada roteador ao longo do caminho entre origem e destino.

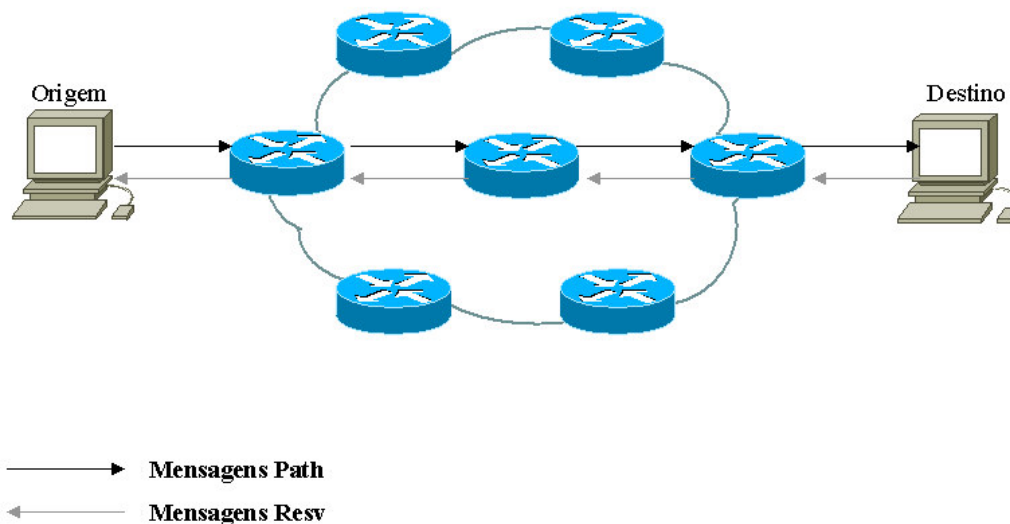


FIGURA 3.3 - Modelo de reserva de recursos usando RSVP

As reservas de recursos em cada roteador são “soft state”. Isto significa que periodicamente cada reserva de recurso para cada fluxo precisa ser renovada. Caso o estado não seja renovado até um limite máximo de tempo, o estado será deletado e o fluxo não será mais tratado com a qualidade de serviço esperada. A renovação de estado é realizada por mensagens *PATH* e *RESV*. Os estados de caminho (*PATH*) e de reserva

(*RESV*) podem também ser removidos através de mensagens específicas de “teardown”, usadas para encerrar uma comunicação IntServ.

3.4.3.3 O modelo DiffServ

Com a finalidade de simplificar o modelo para obtenção de QoS em redes IP, mantendo os mesmos níveis de qualidade de serviço sem adicionar complexidade, custo e dificuldades de escalabilidade, o DiffServ foi concebido. Este modelo está baseado na simples idéia de classificar os fluxos de transmissão (pacotes) em diferentes classes de serviço e oferecer diferentes níveis de qualidade, baseados na classe a que cada pacote pertence. Esta abordagem elimina a necessidade de se manter uma tabela de estados de cada fluxo de comunicação em cada nó presente na rede.

Os serviços DiffServ podem ser obtidos através da combinação dos seguintes mecanismos:

- Setar bits no cabeçalho IP, nos limites da rede dentro do perímetro (limites de um sistema autônomo, domínio administrativo ou hosts);
- usar estes bits para determinar como os pacotes serão encaminhados para os próximos elementos de rede dentro do perímetro;
- condicionar os pacotes marcados com estes bits de acordo com os requisitos ou regras de cada serviço.

Os requisitos de qualidade para cada serviço serão implementados em cada nó de rede, através de uma variedade de serviços de filas ou gerenciamento de disciplinas de filas nas interfaces dos roteadores, como por exemplo: weighted fair queeing, CBQ, RED, WRED, entre outros.

A estrutura DiffServ define um código de serviço diferenciado (DSCP - Differentiated Service Code Point), composto de seis bits que são atribuídos ao campo “Type of Service” do cabeçalho do pacote Ipv4 ou campo “Traffic Class” do Ipv6. As 64 possíveis combinações do DSCP são usadas para marcar o pacote. Todos os pacotes que possuem o mesmo valor de DSCP – em um nó de rede - cruzando uma direção específica é chamado de “Behavior Aggregate” (BA). Assim, pacotes de diferentes origens/aplicações podem pertencer ao mesmo BA (conceito muito parecido com o conceito de classe de serviço).

Para a obtenção de tratamento diferenciado em cada dispositivo de rede, cada BA deve receber um tratamento específico, que define os mecanismos de escalonamento, técnicas de filas, policing e traffic shaping em cada dispositivo. Este conjunto de técnicas que especificam o comportamento de cada pacote em cada nó da rede são definidos como “Per Hop Behavior” (PHB).

Sempre que um roteador receber na entrada um pacote com o campo TOS marcado com um valor de DSCP ele irá tratar este pacote de acordo com o PHB relacionado ao DSCP e assim obter um comportamento de roteamento específico, garantindo um certo nível de QoS.

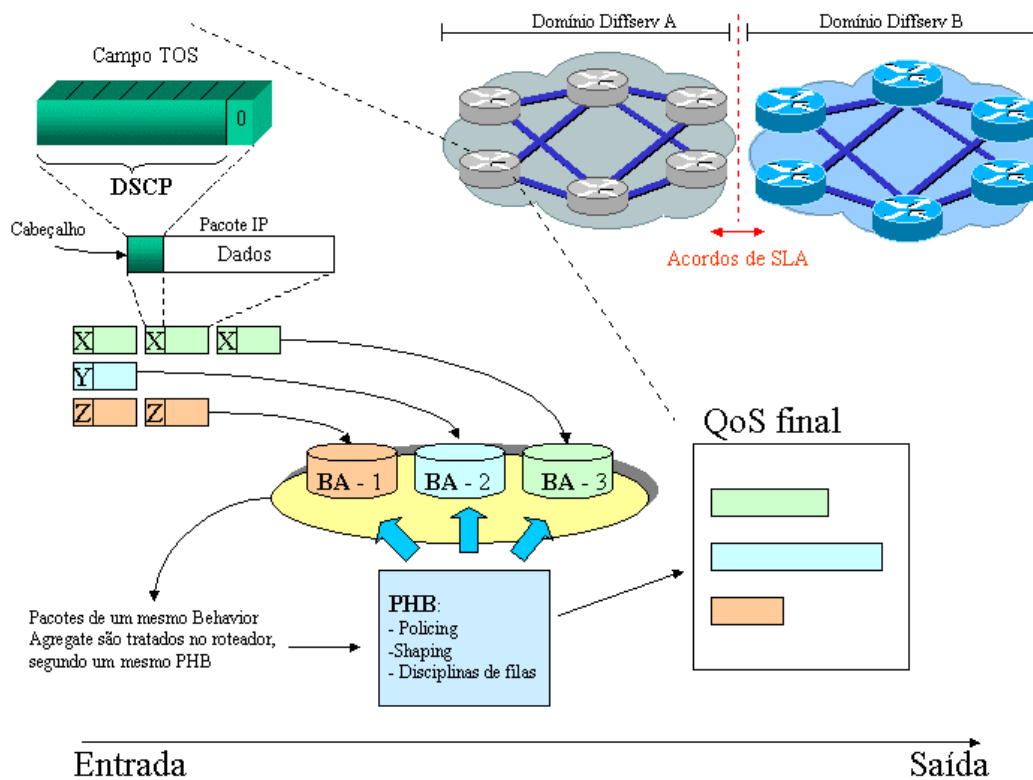


FIGURA 3.4– O modelo DiffServ.

Neste modelo não é necessário manter o estado de qualidade de serviço para cada fluxo existente, o que aumenta consideravelmente a escalabilidade da rede. No DiffServ em vez de cada fluxo ser tratado individualmente nos roteadores, cada pacote é tratado de acordo com o PHB referente ao valor de DSCP presente no campo TOS do cabeçalho do pacote IP. Neste modelo todos os pacotes que possuem o mesmo valor de DSCP serão tratados de forma semelhante pelo roteador.

Para a obtenção de serviços com qualidade fim-a-fim, a RFC 2474 definiu dois ‘per hop behaviors’ (PBH) padronizados, que permitem a configuração de redes baseadas no DiffServ. Outros padrões foram definidos pelas RFCs 2597 e 2598. A seguir, exploramos melhor cada um dos quatro PHB atualmente padronizados. Lembramos que além dos PHB padronizados, cada domínio poderá especificar a combinação que bem entender, para os valores de DSCP. Entretanto, é adequado que quando as características do serviço oferecido forem próximas ao proposto pelo padrão definido pelas RFCs acima descritas, estes padrões sejam utilizados.

PHB default

O PHB default especifica que os pacotes com o campo ToS marcados com o valor ‘000000’ de DSCP obterão o tradicional tratamento “*best effort*” de um roteador compatível com a recomendação DiffServ. Além disso, quando um pacote chegar até um roteador e o seu valor de DSCP não condizer com nenhum outro PHB válido, ele será mapeado para o PHB default (e será tratado normalmente como serviço “*best effort*”). Este PHB não garante nenhum tipo de QoS, mas mantém a compatibilidade com o serviço “*best effort*” atual.

Class selector PHB

Para manter a compatibilidade com o esquema de precedência, definido na RFC 1122, qualquer DSCP na forma 'xxx000' onde x pode ser tanto '1' como '0', foram também definidos.

Neste esquema oito diferentes códigos existem : '000000', '001000', '010000', '011000', '100000', '101000', '110000', '111000'. O significado destes DSCPs são os seguintes:

- 000000 – Rotina (PHB default)
- 001000 – Prioridade
- 010000 – Imediato
- 011000 – Flash
- 100000 – Flash Override
- 101000 – CRITIC/ECP
- 110000 – Controle interredes
- 111000 – Controle da rede

Diferentes níveis de qualidade de serviço poderiam ser obtidos usando estes códigos de DSCP. O exemplo abaixo ilustra o caso onde um provedor de serviço de rede poderia ter configurado os roteadores para oferecer a seus clientes três níveis de serviços: ouro, prata e bronze [KUM 2000]:

TABELA 3-2 – Exemplo de classes de serviço usando Diffserv class selector PHB

Nível de Serviço	Bronze			Prata			Ouro	
Código	000000	001000	010000	011000	100000	101000	110000	111000
Prioridade	2	2	2	1	1	1	0	0

Assured Forwarding (AFxy) PHB

Este modelo especificado pela RFC 2597, define um método onde para cada BA pode ser configurado diferentes classes e os parâmetros configurados devem ser cumpridos pela rede garantindo a qualidade esperada. Este método é equivalente ao serviço de carga controlada do modelo IntServ.

Este modelo define quatro classes AFxy (AF1y, AF2y, AF3y, AF4y), onde x define a classe . Para cada uma destas classes é alocado uma certa quantidade de espaço em buffer e largura de banda na interface. Para o valor do y é possível definir três valores de precedência de descarte para cada classe, de acordo com o SLA definido pelo provedor.

Em caso de congestionamento do link, onde pacotes precisam ser descartados, os pacotes dentro de uma determinada classe AFxy (por exemplo AF3y), os pacotes serão descartados na seguinte ordem de precedência: $dP(AFx1) \leq dP(AFx2) \leq dP(AFx3)$, onde dP é a probabilidade dos pacotes de AFx serem descartados. Seguindo este modelo os pacotes AF33 serão descartados primeiro que os pacotes AF31.

O quadro abaixo define os valores para as classes e precedência de descarte AFxy, onde para o DSCP na forma 'xyzab0', xyz (001/010/011/100) representam a classe e ab (01/10/11) representam a precedência de descarte :

TABELA 3-3 - Exemplo de classes de serviço usando Diffserv Assured Forwarding (AFxy) PHB

Precedência de descarte	Classe 1	Classe 2	Classe 3	Classe 4
Precedência Baixa (descartado por último)	(AF11) 001010	(AF21) 010010	(AF31) 011010	(AF41) 100010
Precedência Média (descartado após pacotes de precedência ALTA terem todos sidos descartados)	(AF12) 001100	(AF22) 010100	(AF32) 011100	(AF42) 100100
Precedência Alta (descartado primeiro)	(AF13) 001110	(AF23) 010110	(AF33) 011110	(AF43) 100110

Expedited Forwarding (EF) PHB

Este modelo de PHB, garante que o BA no elemento de rede, será processado de acordo com uma taxa de ‘throughput’ situada entre uma faixa limite de valores pré - determinada. O processamento dos pacotes que possuem DSCP EF, terá prioridade sem levar em consideração a carga total do elemento de rede. O valor de DSCP recomendado para o modelo EF de PHB é ‘101110’.

O modelo EF oferece a garantia de pequena taxa de perda de pacote, baixo atraso e variação de atraso e serviço de largura de banda garantido, semelhante ao serviço garantido do modelo IntServ. Este tipo de serviço deve ser direcionado apenas àquelas aplicações mais críticas, pois em situações de congestionamento não é possível tratar todos os fluxos de tráfego com prioridade alta.

Além dos valores de DSCP padronizado, vistos acima, é possível que um domínio configure outros valores de DSCP e implemente sua própria política de definição de classes de serviço. Entretanto, vale ressaltar que nesta situação os DSCPs padronizados devem ser mapeados, no dispositivo de rede de entrada em um domínio, para os valores de DSCP que oferecem o mesmo tipo de serviço, conforme um contrato de SLA entre os provedores de serviço.

Geralmente um domínio DiffServ é composto por diversos elementos de rede. Este domínio é limitado nas bordas através de ‘hós limite’ que controlam o ingresso e egresso (saída) de tráfego dentro do domínio. Tipicamente os ‘hós limite’ executam o condicionamento de tráfego, que consiste em classificar os pacotes que estão entrando no domínio, em BAs pré-definidos, verificar a conformidade de tráfego dos pacotes com os parâmetros estabelecidos, marcar ou remarcar os pacotes usando códigos DSCP e finalmente modelar o tráfego ou descartar os pacotes conforme a carga da rede e os BA definidos. A figura 3.4 dá uma idéia bastante clara deste conceito.

Os ‘hós limite’, são os responsáveis por manter os acordos de níveis de serviço (SLA), conforme definido na política do provedor de serviço. Executando as ações explicadas acima, pacotes marcados com um determinado valor de DSCP que recebe alta prioridade no domínio A poderá continuar recebendo este tratamento prioritário, independente do valor de DSCP escolhido pelo provedor B, no caso de serviços de alta prioridade.

Além dos mecanismos discutidos acima (IntServ, RSVP e DiffServ), é possível provisionar QoS através de outras alternativas e protocolos não discutidos neste texto. Algumas destas alternativas são mais apropriadas para tráfego de dados e não para tráfego de voz e vídeo como é o caso da técnica CAR (Committed Access rate). Outras técnicas como o PBR (Policy Based Routing) da Cisco, são técnicas simples mas configuradas estaticamente nos roteadores. Outras técnicas como MPLS para ambiente WAN e padrões como IEEE 802.1D e IEEE 802.1Q, para o ambiente LAN podem

também serem utilizadas, mas nenhuma apresentam tanta flexibilidade ou encontram-se em estágios tão maduros em termos de padronização para o protocolo IP, como as técnicas discutidas acima. Por este motivo a grande maioria de sistemas baseados no IP utilizam o IntServ e mais recentemente o DiffServ.

3.5 Controlando QoS em sistemas H.323

A recomendação H.323 [ITU 98], em seu Anexo II, especifica como o RSVP pode ser usado como um possível mecanismo para o provisionamento de QoS em redes IP, embora outros protocolos possam também ser utilizados. O anexo descreve o método geral da utilização do RSVP por entidades H.323, com o intuito de evitar conflito entre as implementações dos fabricantes.

O padrão atual define que existem duas entidades H.323 capazes de controlar QoS: o terminal ou o gatekeeper. Se o terminal possui capacidade de controlar QoS, ele pode fazer sem nenhuma assistência do gatekeeper. Entretanto, quando o terminal não possui a capacidade mínima para controlar QoS, o gatekeeper poderá fazê-lo em benefício do terminal.

Quando um terminal requisita admissão para o gatekeeper ele deverá indicar na mensagem ARQ, a sua capacidade para manipulação de QoS. Baseado nesta informação, o gatekeeper poderá decidir se ele permitirá que o terminal utilize as suas próprias capacidades de controle de QoS ou se o gatekeeper irá manipular o QoS para o terminal. Esta decisão é transmitida para o Terminal no campo *TransportQOS* da mensagem ACF. Quando o terminal coloca o valor *endpointControlled*, significa que ele deseja controlar o QoS. Quando o valor é *gatekeeperControlled*, o terminal deseja que o gatekeeper controle o QoS. Se o valor for *noControl*, significa que não será necessário o uso de QoS.

Além desta indicação, o terminal deve calcular a quantidade de banda que ele pretende utilizar para todos os canais da sessão. Esta informação deverá ser atribuída ao campo *bandWidth* da mensagem ARQ, independente do uso do RSVP. Se durante a sessão existir a necessidade de aumento de banda, este requisito deverá ser indicado conforme os procedimentos normais, usando a mensagem BRQ.

Caso o gatekeeper decida que o terminal deve ser o responsável por manipular as requisições de QoS, mas o terminal não possuir esta capacidade, o gatekeeper deverá rejeitar a conexão do terminal através da mensagem ARJ.

As mensagens RSVP deverão ser parte dos procedimentos de abertura dos canais lógicos, após as mensagens de sinalização H.225.0-Q931 e troca de capacidades terem sido realizadas entre os participantes da conferência. Neste momento, o terminal deve especificar os parâmetros RSVP do fluxo de dados no campo *QOSCapability* da mensagem *OpenLogicalChannel*. Apenas após a origem receber a resposta *OpenLogicalChannelAck*, ela poderá iniciar a emitir mensagens *PATH* para o destino, pois o endereço de transporte (porta) de conexão virá indicado na mensagem *OpenLogicalChannelAck*.

Caso o destino não desejar receber pacotes do fluxo de dados que está sendo estabelecido, antes que as reservas RSVP sejam completamente estabelecidas, ele deverá atribuir o valor TRUE, para o campo *flowcontrolToZero* da mensagem *OpenLogicalChannelAck*, indicando para a origem que ele não deseja receber nenhum tráfego neste canal antes que as reservas apropriadas sejam estabelecidas. Quando a origem receber a mensagem *OpenLogicalChannelAck*, com o campo

flowcontrolToZero, possuindo o valor TRUE a origem não deverá enviar nenhum fluxo de dados por este canal.

Após o destino receber uma mensagem RSVP *RESVCONF*, confirmando que as reservas foram enfim estabelecidas, ele pode enviar um comando *FlowControlCommand*, para a origem, para cancelar o efeito que o campo *flowcontrolToZero* causou anteriormente.

Assim que a origem receber o comando *FlowControlCommand* ele inicia a transmissão dos dados do canal. O destino deverá implementar timers, para controlar possíveis perdas da mensagem *RESVCONF*, e evitar que o envio dos sinais de mídia sejam retardados indefinidamente.

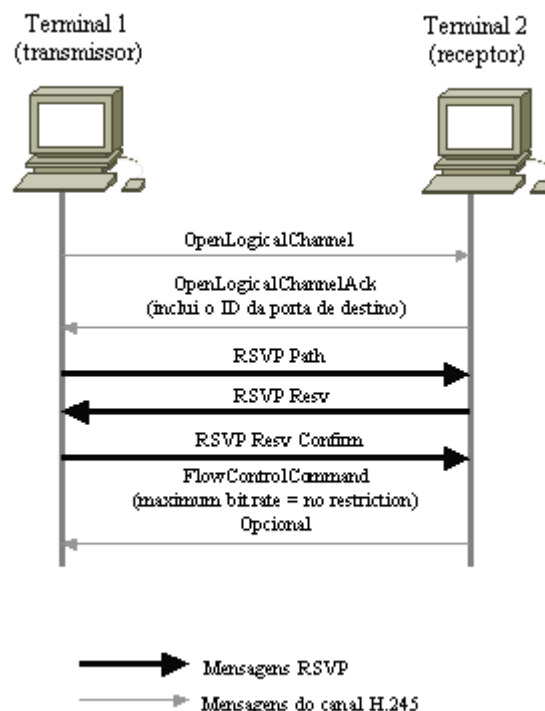


FIGURA 3.5 - Sequência de mensagens para estabelecer um canal RSVP unicast

O comportamento dos terminais, caso o procedimento de reserva RSVP falhar, é deixado a cargo dos fabricantes. Entretanto, se o destino decidir que o nível de qualidade “best effort” não é suficiente, ele deverá encerrar os canais de transmissão abertos através do comando *RequestChannelClose*.

Outro ponto importante é que o H.323 especificou o estilo de reserva RSVP Fixed Filter. Neste estilo, as reservas se aplicam apenas a um terminal específico. Outros estilos permitem o compartilhamento de reservas entre diferentes terminais.

3.6 Uma nova perspectiva para qualidade de serviço em sistemas H.323

Embora [ITU 98], especifique como aplicações H.323 podem utilizar o conjunto RSVP/IntServ como possível mecanismo de QoS para os canais de mídia, em redes que

empregam mecanismos de transporte homogêneos – conforme descrito na sessão anterior – este modelo ainda não é empregado de fato.

Desde a inclusão destes procedimentos em 1998, os modelos propostos para o provimento e controle de QoS em redes IP sofreram alterações bastante grandes [RAJ 99]. Entre estas mudanças podemos citar duas muito significativas:

- Atualmente o modelo DiffServ é o mecanismo preferencial para prover QoS no centro (backbone) das redes IP.
- Migração na direção de um sistema de redes baseado em Políticas [FIN 2002].

Trabalhos desenvolvidos dentro de organizações de padronização como o IETF, vêm mostrando que o modelo proposto pelo conjunto RSVP/IntServ apresenta algumas limitações que restringem a utilização deste modelo na parte central de uma rede IP. O ETSI e o ITU-T mostram que mecanismos robustos de QoS são necessários, justamente no “backbone” da rede para aplicações de tempo real, como a colaboração visual.

A primeira limitação do RSVP é a necessidade em manter informações sobre o estado de cada fluxo de comunicação em cada dispositivo de rede ao longo do caminho entre origem e destino. Além disso, estes estados requerem renovações periódicas, o que reduz consideravelmente a escalabilidade do sistema além de agregar carga de tráfego à rede como um todo, conforme apontado na sessão 3.3.3.

Em segundo lugar, a maioria dos hosts/terminais de hoje não geram sinalização RSVP e este modelo pressupõe que todos os dispositivos ao longo do caminho entre origem e destino possuam capacidades RSVP. Embora este número tende a crescer muito nos próximos anos, ainda são limitadas as aplicações que utilizam RSVP para requisitar QoS da rede. Na verdade, os principais movimentos comerciais vêm sendo no sentido de prover os terminais com capacidades de marcação de pacotes, para utilização do DiffServ (e não do Inteserv), como a tecnologia IPriority™, desenvolvida pela Polycom em seus terminais [ONE 2002] e a capacidade do Cisco IP phone configurar o campo ToS de pacotes Ipv4 [LI 2000].

Em terceiro lugar, algumas aplicações possuem necessidades de QoS, mas não são aptas a expressarem estes requisitos utilizando o modelo IntServ. Conforme visto na sessão 3.4, o DiffServ apresenta uma arquitetura mais robusta e escalável, permitindo a configuração de diversas classes de serviços, de forma a acomodar os mais diversos requisitos de QoS das aplicações.

O segundo ponto de limitação do modelo proposto em [ITU 98], é que este modelo não prevê nenhum tipo de entidade de controle, capaz de gerenciar os recursos da rede de maneira organizada e escalável. Na verdade, [BUC 2000] prevê a necessidade de introduzir entidades funcionais de QoS para sistemas H.323, responsáveis por controlar, autorizar e sinalizar integralmente as requisições de QoS em redes que possuem diferentes mecanismos de transporte. Além disso, conforme exposto por [FIN 2002], um servidor de QoS é uma entidade funcional essencial que deve centralizar a visão geral da rede, provendo funcionalidades como: sinalização de VoIP, controle de políticas, acesso a servidores de diretórios e banco de dados, etc. As entidades propostas por [BUC 2000] são:

- Entidade de QoS do Terminal (EPQoSE) – Esta entidade funcional deve residir dentro do terminal (endpoint) e é responsável por questões de QoS relacionadas com o terminal. Esta entidade deve ser responsável por emitir requisições de QoS em benefício do terminal.
- Gerenciador de QoS (QoSM) – Esta é a entidade funcional que interage com cada uma das outras entidades funcionais, inclusive outros gerenciadores de

QoS. O gerenciador de qualidade de serviço, recebe requisições da entidade de QoS do terminal e checa com a entidade de políticas de QoS, para verificar se a requisição pode ser concedida ou não. Ele então contacta a entidade de terminal para entregar a resposta. O gerenciador de QoS, também interage com o gerenciador de recursos para reservar os recursos necessários para esta tarefa.

- Entidade de Políticas de QoS (QoSPE) – Esta entidade é responsável por determinar se uma determinada requisição de QoS de um terminal pode ou não ser autorizada. A partir de uma requisição do gerenciador de QoS (QoSM), a entidade de políticas de QoS responde com informações de autorização.
- Gerenciador de Recursos de Transporte (TRM) – Esta entidade é responsável por reservar recursos de transporte dentro do seu domínio administrativo, a partir de requisições do Gerenciador de QoS (QoSM). O TRM é o responsável por configurar um conjunto de políticas dentro de um conjunto de recursos de transporte presentes no domínio.

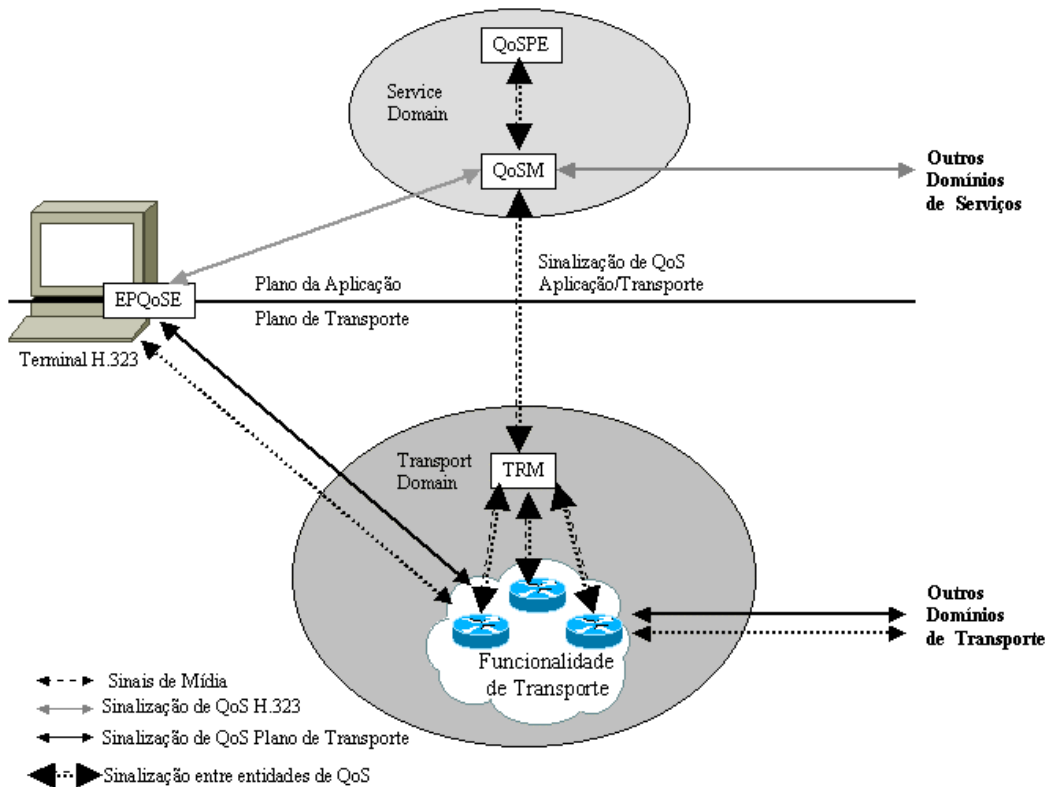


FIGURA 3.6 - Relacionamento entre as entidades funcionais de QoS

Embora o local físico destas entidades possa variar, o modelo proposto por [BUC 2000], apresenta o EPQoSE localizado no Terminal H.323, o QoSM funcionalmente localizado dentro de um Gatekeeper H.323, o QoSPE localizado dentro de um repositório de políticas, enquanto o TRM está localizado dentro de um negociador de largura de banda (bandwidth broker). Os recursos de transportes que o TRM é responsável por configurar são todos os recursos de rede presentes no domínio, como roteadores, switches, firewalls, etc..

O relacionamento entre estas novas entidades funcionais, conforme proposto em [BUC 2000], está ilustrado na figura 3.6. O modelo, adota o conceito onde as reservas de recursos de transporte são realizadas através de sinalização entre o QoSM e o TRM (sinalização de QoS H.323), embora mantenha o modelo explorado em [ITU 98], onde a reserva de recursos pode ser executada pelo terminal através de protocolos de sinalização de Transporte, como o RSVP (sinalização de QoS do plano de transporte).

O protocolo usado entre QoSMs e TRMs, bem como entre QoSMs e QoSPEs não fazem parte do escopo proposto por [BUC 2000] e é deixado a cargo de outros grupos de padronização ou dos próprios fabricantes.

3.7 Um sistema de QoS baseado em políticas

A introdução de um sistema de gerenciamento de QoS baseado em Políticas, conforme exposto em [FIN 2002] é uma idéia bastante nova e atualmente movimentos no sentido de padronizar o funcionamento destes sistemas começam a aparecer. Estes movimentos estão sendo puxados principalmente pelo grupo de Estruturas de Políticas do IETF, mas trabalhos importantes neste sentido também estão sendo realizados pelo grupo de trabalho em Qualidade de Serviço da Internet2 [TEI 2000]. É importante salientar que até hoje os modelos de gerenciamento da Internet estavam baseados no simples gerenciamento dos dispositivos da rede, entretanto, o foco vem mudando rapidamente para o gerenciamento de serviços, caracterizado principalmente pela manutenção adequada do nível de qualidade de serviço esperado pelos clientes [PAR 2001].

A estrutura apresentada por [BUC 2000] e explorada anteriormente, está muito próxima do modelo de gerenciamento baseado em políticas proposto por [MAH 2000] – direção que o IETF vem trabalhando de maneira muito forte. Este modelo apresenta elementos funcionais semelhantes aos explicados anteriormente.

Realizando uma simples tradução dos nomes das entidades funcionais podemos verificar a proximidade das propostas. No modelo de [MAH 2000] o QoSPE é chamado de Repositório de Política, o QoS M é conhecido como PDP (Policy Decision Point) e o TRM é chamado de Policy Consumer. Os papéis desempenhados por estas entidades são praticamente os mesmos apresentados anteriormente, podendo haver alguma diferenciação. Um esquema deste modelo é visto na figura 3.7.

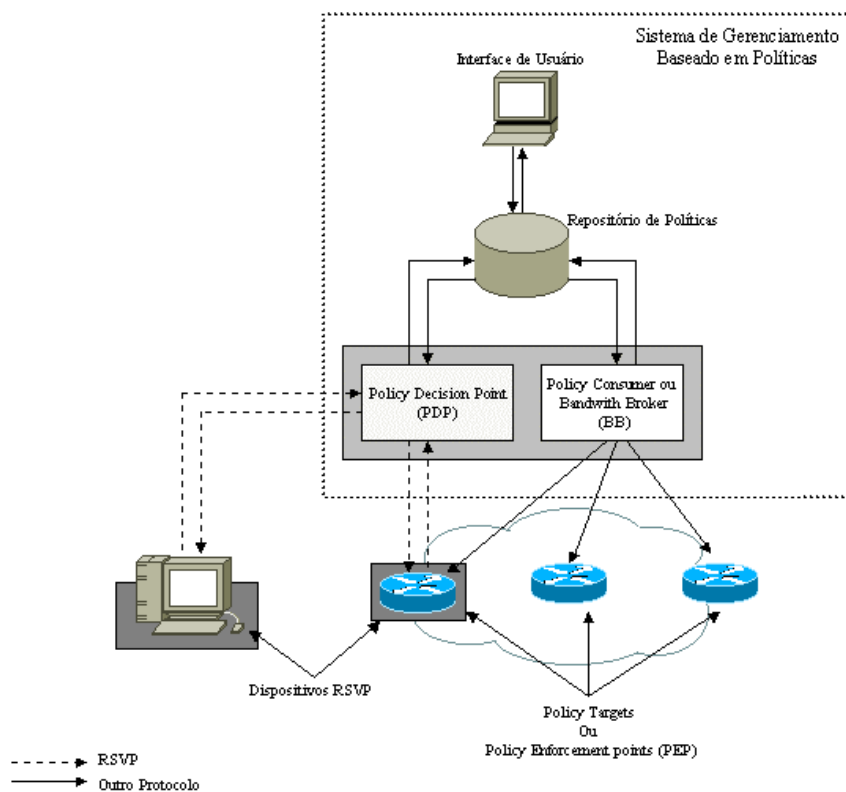


FIGURA 3.7 – Visão esquemática de um sistema de Gerenciamento baseado em Políticas

A razão pela qual um mecanismo para alocação de QoS baseado em políticas deve existir é bastante simples: não parece coerente deixar que os próprios usuários controlem de forma independente a sua necessidade de QoS. Segundo [NIC 99], o controle independente pelos usuários pode ser bastante simples de ser implementado, mas improvável de ser suficiente, pois não é razoável esperar que todos os usuários saibam as necessidades reais nem a carga de utilização de rede atual das suas organizações, e principalmente, marquem seu tráfego de acordo com estas prioridades (respeitando as necessidades de tráfego de outros usuários). Portanto, a utilização de agentes que tenham conhecimento das prioridades e políticas da organização, deve ser utilizada para alocar e controlar de forma centralizada os recursos de rede disponíveis.

Outro ponto a ser considerado, conforme apontado por [RAJ 99], é sobre a real necessidade do Gatekeeper controlar QoS em sistemas H.323. Em seus apontamentos, [RAJ 99] coloca que independente do QoSM ou QoSPE estarem ou não localizados funcionalmente dentro do Gatekeeper, são eles que determinam se uma certa requisição de QoS de um terminal é permitida ou não, e não o Gatekeeper. Portanto, em seu raciocínio não seria o Gatekeeper a entidade responsável por controlar a qualidade de serviço em sistemas H.323, mas sim entidades como o QoSM e o QoSPE.

Além disso, [STA 2001] e [OPE 2001], colocam que as atividades de gerenciamento e controle de banda, previstas para serem realizadas pelo Gatekeeper, encontram-se em um estágio de implementação bastante rudimentar, por que os administradores de rede podem apenas limitar a utilização da banda disponível, através da especificação do número máximo de chamadas simultâneas que podem ser aceitas pelo Gatekeeper. Isso significa dizer que o gatekeeper irá aceitar ou rejeitar a chamada, baseado apenas no número máximo de conexões que ele está configurado para receber e não no estado real de carga da rede.

Neste sentido a abordagem proposta, sugere retirar do gatekeeper a responsabilidade por controlar a carga da rede e conseqüentemente o QoS relacionado. Esta solução propõe que este controle seja realizado por outras entidades funcionais que integram um sistema de gerenciamento de QoS baseado em Políticas.

3.7.1 Funcionamento geral de um sistema de QoS baseado em políticas

Em um sistema de QoS baseado em Políticas, o administrador deve ser capaz de inserir novas políticas, editar políticas existentes e remover políticas no Repositório de Políticas (QoSPE), conforme especificado por [MAH 2000]. Este repositório reflete as prioridades da organização, as políticas estabelecidas, os acordos de SLAs firmados entre usuários e outros domínios, etc.. É importante lembrar que cada política deve estar relacionada com os dispositivos de rede do domínio, de modo que possam ser configuradas apropriadamente nestes dispositivos.

Cada vez que um terminal desejar iniciar uma sessão de colaboração visual com qualidade de serviço superior ao tradicional “best effort”, ele poderá fazê-lo de duas formas:

- Através dos acordos pré-estabelecidos com o provedor de serviço ou administrador de rede.
- Através de sinalização sob-demanda para o QoSM.

Para os casos de acordos pré-estabelecidos, uma interface de comunicação não dinâmica como telefone ou fax pode ser empregada. Já no segundo caso, um protocolo de comunicação entre o EQoSPE e o QoSM deve necessariamente existir. Em um cenário mais sofisticado, o terminal poderia avisar ao QoSM sobre qual é o modo que deseja realizar a chamada: provisionado usando valores “default” ou sinalizado usando parâmetros informados sob-demanda. Neste caso, o terminal sinaliza seu desejo ao QoSM através do EPQoSE. A maneira pela qual estas entidades devem se comunicar é discutida mais adiante.

Depois que o QoSM é sinalizado, ele deve consultar o QoSPE, para verificar se existem políticas estabelecidas que permitam a reserva de recursos para esta requisição. Para os acordos pré-estabelecidos, esta consulta pode ter sido feita anteriormente. Depois que o QoSM receber a resposta do QoSPE, o QoSM deverá interagir com o TRM de forma a indicar para esta entidade que novas políticas devem ser configuradas nos dispositivos de redes correspondentes (roteadores, switches, firewalls, etc).

O TRM deve então, sinalizar as novas políticas para cada um dos dispositivos de rede presentes no domínio e instalar estas políticas nestes dispositivos. O TMR pode executar esta configuração empregando diversos protocolos de comunicação como SNMP, COPS, Telnet ou outra forma de comunicação, de acordo com a semântica e capacidade de cada dispositivo. O TMR deve responder ao QoSM sobre o sucesso ou falha da configuração dos dispositivos. Uma vez com a resposta, o QoSM deverá sinalizar ao EPQoSE se a sua requisição foi ou não aceita.

Se a resposta foi positiva e o QoSM concedeu a chamada com os requisitos de QoS solicitados pelo terminal ele prosseguirá realizando a conexão com o destino. Caso a resposta foi negativa, o terminal poderá decidir prosseguir com a sessão sem nenhuma qualidade de serviço, ou ainda cancelar a sessão.

3.7.2 *Características da requisição de QoS por mecanismos provisionados e sinalizados*

O processo de alocar os recursos de rede pode ser realizado por diferentes mecanismos: o modelo provisionado, também chamado de estático ou pré-acordado e o modelo sinalizado, também chamado de alocado ou sob-demanda. Existem motivos para justificar a presença destes dois modelos para a requisição de QoS.

O modelo provisionado, pode ser configurado usando como base, especificações de períodos do tempo, como por exemplo hora do dia, mês do ano, período de vigência de um contrato, etc., mas as especificações não podem ser modificadas em resposta a mensagens dinâmicas [NIC 99]. Este mecanismo permite que os contratos entre usuários e prestadores de serviços de telecomunicações sejam realizados, baseados em observações estatísticas e padrões de tráfego da rede, permitindo também que o estado desejado seja forçado, em função das necessidades operacionais ou de negócio da organização [MAH 99]. O mecanismo de alocação de QoS provisionado é uma boa opção para tráfego tolerante a variação na qualidade do fluxo, como ftp, http, telnet, entre outros.

As reservas de recursos neste modelo poderiam funcionar como descrito em [TEI 2000]: Neste caso existe uma pessoa que faz o papel do QoSM/TMR, para o domínio. Esta pessoa possui uma matriz de tráfego para o domínio em questão e baseado nesta matriz de tráfego e nos parâmetros de QoS solicitados pelo usuário, o administrador do domínio (“QoSM humano”) pode aceitar ou rejeitar a concessão de QoS para o fluxo de comunicação. Esta prática simples é bastante utilizada atualmente, em diversas organizações que possuem o seu canal de comunicação bem utilizado, mas que necessitam de uma reserva de largura de banda para a realização de sessões de colaboração visual ocasionalmente. Nestes casos, o administrador da rede configura os dispositivos manualmente para o período de duração da sessão de colaboração visual, retirando a configuração após a realização da sessão de colaboração visual.

Este modelo provisionado apresenta um mecanismo de cobrança estático. Em outras palavras, isto significa que você pode estar pagando pela transferência de dados que nunca são enviados para a rede, ou como muito bem colocado por [NIC 99], paga-se pelo nível de serviço que se espera ter disponível em algum tempo.

O modelo sinalizado, em contra partida, permite que parte da rede ofereça garantias de qualidade ao mesmo tempo em que utilizam os recursos da rede de forma eficiente e conseqüentemente, permite tarifar o usuário de maneira mais justa (pela quantidade de dados que trafegam na rede com qualidade de serviço requisitada). Entretanto, mecanismos de sinalização incorrem em tráfego adicional na rede, atraso no tempo de estabelecimento da chamada e recursos de processamento nos dispositivos envolvidos na troca da sinalização.

Este “overhead”, é facilmente justificado para conexões que irão durar por um período de tempo maior, ou onde a variação dos parâmetros de qualidade podem produzir efeitos indesejáveis ou degradar significativamente o resultado final da comunicação. Sistemas de Colaboração Visual (VoIP), se incluem nestes critérios [MAH 99].

Portanto, quando se deseja utilizar um modelo de requisição sob-demanda, as aplicações de Colaboração Visual precisam possuir a capacidade de sinalizar para rede as suas necessidades de QoS. Isto significa empregar um esquema de comunicação/sinalização entre terminais e a rede .

3.8 Negociação de QoS entre terminais e o sistema de políticas

Como vimos na sessão anterior, a requisição de QoS de forma provisionada, embora trabalhosa de ser realizada quando não existem mecanismos automáticos, pode ser feita atualmente sem maiores problemas, bastando para isso tempo do administrador do domínio.

Entretanto quando falamos na necessidade de um modelo para obtenção de QoS de forma sinalizada, a complexidade aumenta consideravelmente. Como os protocolos de Telefonia IP, não foram concebidos na sua origem, para tratar adequadamente as questões relacionadas com QoS, até por que os padrões ainda não estavam completamente maduros, não existe um processo de requisição e alocação de QoS, na fase de sinalização da chamada.

Além disso, os protocolos existentes, como o H.323, foram concebidos para alocar de forma dinâmica os endereços de transporte das conexões de mídia, impossibilitando o gatekeeper, o QoSM ou o QoSPE, de saber antecipadamente que endereços de transporte serão utilizados para o tráfego de mídia. Para se ter uma idéia do leque de opções, [MIC 2002] e [FVC 2001], indicam que quando um terminal H.323 é utilizado para realizar sessões de colaboração visual, diversos endereços de transporte são utilizados para estabelecer a conexão com outro participante. As portas utilizadas nesta comunicação são:

- Porta TCP e UDP 1720 para sinalização da chamada H.323.
- Porta UDP 1718 para descobrimento multicast do gatekeeper (opcional).
- Porta UDP 1719 para comunicação com o gatekeeper (opcional).
- Portas dinâmicas TCP entre 1025 – 655365 para controle de chamada H.245.
- Portas dinâmicas UDP entre 1025 – 655365 para enviar e receber mídia através de RTP sobre UDP.

Como a definição das portas para enviar e receber os sinais de áudio e vídeo, serão conhecidas, apenas momentos antes do estabelecimento do canal de mídia, o primeiro impulso é colocar o processo de alocação de QoS, após o estabelecimento da chamada e antes da abertura do canal de mídia, logo após o recebimento da mensagem *OpenLogicalChannelAck*, como ilustrado na figura 3.8 ou proposto em [ITU 98].

Esta característica não apresenta grande vantagem, por que a sinalização de QoS acontece no meio dos procedimentos de abertura dos canais de mídia. Neste caso, quando o terminal destino começa a sinalizar uma chamada entrante, mas nos procedimentos de sinalização de QoS, um dispositivo intermediário não possui recursos suficientes para alocar o QoS requerido, a chamada seria rejeitada logo após o participante localizado no terminal destino ter atendido a chamada. Este comportamento não é aceitável, além de confundir os participantes [STA 2001]. Desta forma, o terminal destino deveria apenas alertar o usuário, sobre uma chamada entrante, após a reserva dos recursos terem sido completamente realizadas e confirmadas. Este modelo de funcionamento não é possível de ser obtido com as definições apresentadas em [ITU 98].

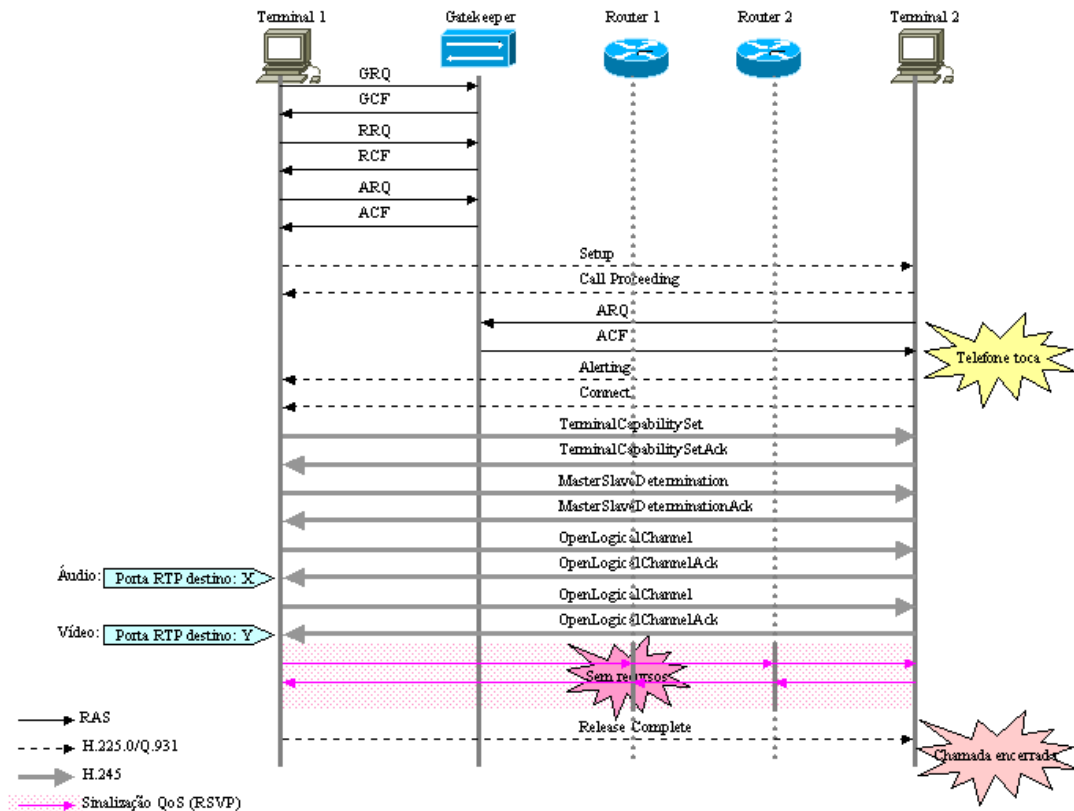


FIGURA 3.8 – Encerramento de chamada por falta de recursos de rede

Outro problema existente nesta abordagem é a introdução de atraso na fase de estabelecimento da conexão propriamente dita, o que representa um comportamento indesejável neste estágio avançado da comunicação, além de ir contra os preceitos de QoS definidos pelo ETSI, conforme discutido na sessão 3.1. Imagine que a fase de abertura dos canais lógicos H.245 seja interrompida, para que o EPQoSE negocie os parâmetros de QoS com a rede, enquanto o usuário que já se prontificou a atender a chamada fica aguardando até que esta negociação de QoS seja encerrada. Somente após a negociação de QoS ser encerrada, é que o procedimento de abertura dos canais de mídia é continuado e a comunicação entre os terminais é estabelecida. Este atraso, bem no meio dos procedimentos de estabelecimento de conexão, degrada significativamente a percepção de qualidade, por parte dos usuários, principalmente do usuário que está recebendo a chamada.

Conforme exposto por [PLA 95], o objetivo da negociação de QoS é acordar parâmetros de QoS que satisfaçam tanto o usuário quanto o provedor de serviço. Em nossa opinião esta negociação deve permitir que o usuário requisite o tipo de QoS que ele pensa ser necessário para a sua aplicação, bem como permita que o provedor de serviços o tarife e controle a utilização dos recursos da rede de maneira apropriada.

É fundamental que a base sobre a qual se estabeleça esta negociação seja um acordo de nível de serviço, o qual tanto o usuário quanto o provedor de serviço se comprometam a respeitar. Este acordo pode ser flexível o bastante para prever requisições sob-demanda, ou pode ser suficientemente rígido para ignorar requisições adicionais ao acordo estabelecido. Esta decisão fica a cargo das entidades envolvidas – usuário e provedor de serviço (ou administrador do domínio). Um acordo de nível de serviço para redes IP pode conter, por exemplo, as seguintes métricas [RAY 2000]:

- Disponibilidade do serviço.
- Atraso da rede.
- Variação no atraso (jitter).
- Tempo de resposta.
- Taxa de perda.
- Tempo médio de reparo.
- Tempo médio entre falhas.
- Disponibilidade da rede.

O modelo de negociação entre terminais e um sistema de políticas de QoS, deve permitir que o usuário, antes de iniciar a comunicação, verifique se a rede pode lhe prover um nível de QoS superior ao acordado em contrato. Caso o provedor possua os recursos necessários para oferecer este novo requisito, sob-demanda, ele poderá optar por fazê-lo e tarifar o usuário apropriadamente.

Caso a requisição extrapolar os recursos existentes, o provedor poderá negar esta solicitação sob-demanda, mas ficará obrigado a proporcionar o nível de QoS acordado previamente. Este nível de serviço acordado previamente poderá ser superior ao tradicional serviço “best-effort” ou no caso de não existir um acordo prévio entre as partes, nenhum nível de qualidade além de “best-effort” será oferecido ao usuário, que poderá optar por fazer a chamada com nível “best-effort” ou não realizar a chamada.

4 Mecanismo de requisição de QoS para um ambiente de colaboração visual baseado em H.323

4.1 Introdução

Como se pode perceber nas últimas sessões do capítulo anterior, vários são os motivos que incentivam a criação de um modelo para o uso efetivo de QoS por aplicações, multimídia em redes IP, onde o Terminal de colaboração visual H.323 possa requisitar certo nível de QoS para a rede, de forma automática, mas onde o controle do QoS permanece com as entidades da rede.

Recapitulando rapidamente estes pontos, vimos que não é apropriado deixar o controle e o gerenciamento dos recursos de rede com o Terminal H.323 e por isso um mecanismo de QoS baseado em políticas é um componente importante do sistema.

O segundo ponto a ser lembrado é que atualmente nenhuma aplicação de colaboração H.323, implementa o modelo de provisionamento de QoS proposto em [ITU 98]. Além disso, os movimentos de permitir ao terminal H.323 requisitar QoS da rede, não vão além de implementar a marcação de pacotes no campo ToS do pacote IP, para prover qualidade de serviço baseado no modelo DiffServ. Este modelo, entretanto, continua deixando o controle de decidir sobre a utilização de QoS com o terminal. Vale a pena lembrar que nesta abordagem, quando o terminal marca os pacotes IP, com um certo valor de DSCP, o roteador de borda do domínio estará realizando algum tipo de “Traffic Shapping” e caso este verificar em sua lista de controle de acesso que este fluxo não possui nenhuma permissão para receber tratamento diferenciado, ele irá reclassificar o pacote, marcando-o com o valor ‘000000’, no campo ToS do pacote IP, não provendo nenhum tipo de QoS para este fluxo de comunicação e principalmente, não retornando nenhum tipo de alerta para o terminal que está originando a chamada. Desta forma, o terminal pensa estar enviando tráfego com QoS, mas a rede não trata este tráfego com o QoS requisitado.

Além disso, vimos que a utilização do RSVP/Intserv, além de apresentar problemas de escalabilidade, já discutidos, pode prejudicar a percepção geral dos usuários em relação à qualidade de serviço, por que a negociação de QoS proposta para um sistema H.323 usando RSVP, acontece bem no meio dos procedimentos de abertura dos canais de sinalização, o que além de introduzir atraso, pode levar a aplicação a um tipo de comportamento não apropriado, onde após o terminal sinalizar uma chamada entrante e o usuário atender esta chamada, a mesma poderá ser encerrada involuntariamente, por que a rede não dispõe dos recursos de QoS solicitados.

Todos estes motivos abrem espaço para uma abordagem diferente: a implementação de um terminal de colaboração visual que tenha a capacidade de sobrepor estes problemas e introduzir um modelo alternativo para a requisição de QoS da rede, na realização de sessões de colaboração visual.

Este sistema está baseado em um terminal (e outras entidades), capaz de sinalizar para a rede as suas necessidades de QoS e receber um “feedback” da rede a respeito do sucesso ou falha da alocação dos recursos que irão prover o nível de QoS requisitado. Esta funcionalidade permite que o terminal que está originando a chamada possa decidir se vai continuar ou não a chamada, caso a reserva de QoS falhar.

O trabalho apresenta esta proposta alternativa, de forma a preencher a lacuna existente, e mostra uma abordagem onde um terminal H.323 pode realizar sessões de colaboração visual, com níveis de qualidade de serviço adequadas às necessidades dos usuários, ao mesmo tempo em que se beneficia de um modelo de QoS baseado em

políticas, responsável por manter o controle sobre os recursos da rede com as entidades de rede, isto é, com o provedor de serviços.

As próximas sessões explicam a arquitetura do modelo proposto, bem como estabelecem um modelo de operação e explicam como o protótipo de um terminal H.323, implementa estes conceitos.

4.2 Arquitetura do ambiente

A abordagem proposta neste texto, está baseada em um terminal H.323, capaz de sinalizar para a rede as necessidades de QoS da sessão de colaboração visual que o terminal pretende realizar. Uma visão geral dos componentes de nossa proposta de terminal é vista na figura 4.1.

A grande diferença do terminal aqui apresentado em relação aos terminais H.323 presentes no mercado é a incorporação de uma camada responsável pela sinalização de QoS em benefício do terminal. Esta camada, será denominada EPQoSE, para preservar a nomenclatura que vem sendo proposta em [BUC 2000].

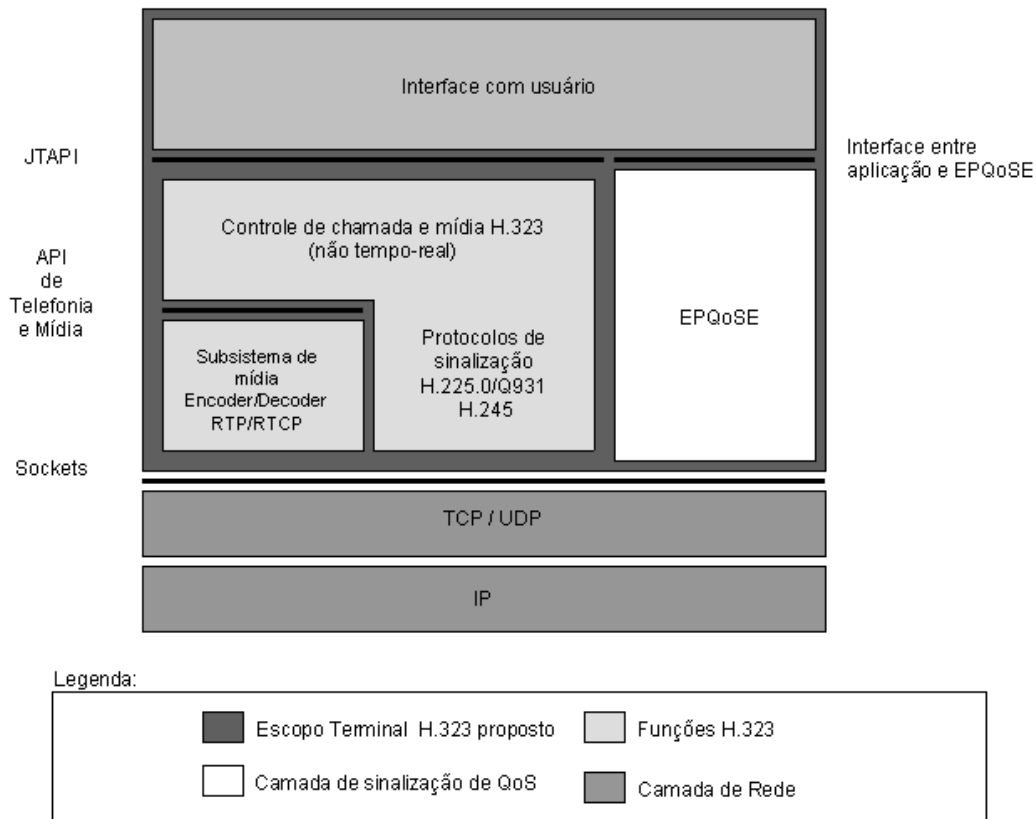


FIGURA 4.1 –Arquitetura do terminal H.323 proposto

As funções básicas a serem desempenhadas pelo EPQoSE, são (1) interagir com o terminal H.323 de forma a interpretar os parâmetros de QoS da aplicação (2) sinalizar para as entidades responsáveis por conceder recursos da rede os parâmetros de QoS do terminal H.323. Estas entidades podem ser o QoSM ou o QoSPE, conforme proposto em [BUC 2000]. (3) Receber resposta sobre o sucesso ou falha de alocação de QoS dentro do domínio. (4) Alertar a aplicação (interface com o usuário) sobre o sucesso ou falha de alocação de QoS, para que o usuário possa tomar a decisão de continuar com a

chamada ou interromper com os procedimentos de estabelecimento da chamada, se verificar que os recursos disponíveis não são suficientes para atender à sua demanda. (5) No caso da alocação de QoS ter sucesso, quando a sessão encerrar, o terminal deverá indicar o encerramento para que a rede possa desalocar os recursos de rede e tarifar o usuário adequadamente. (6) Receber o indicativo da rede, que sua sinalização de encerramento da sessão foi recebida com sucesso.

No caso do terminal decidir por realizar a chamada, os procedimentos de sinalização para interação com o Gatekeeper (H.225.0-RAS), estabelecimento de chamada (H.225.0-Q931), determinação de capacidades e determinação mestre/escravo, a abertura de canais para transmissão de mídia (H.245) e o envio de mídias (RTP/UDP), explicadas no capítulo 2 deste texto, serão realizadas, através das funções H.323 do terminal.

4.2.1 O módulo EPQoSE

Como os requisitos de QoS da aplicação dependem da performance da rede que suporta o tráfego IP, estas necessidades devem ser traduzidas adequadamente para parâmetros QoS, que possam ser compreendidos pelo sistema de gerenciamento de QoS e em última instância pelos dispositivos de rede. Por isso, estas necessidades foram divididas em duas partes: o descritor de QoS e o descritor de Tráfego, conforme explicado anteriormente no item 3.1. Estas duas estruturas de informação fazem parte do EPQoSE. Os parâmetros escolhidos para fazerem parte do Descritor de Tráfego, bem como pelo Descritor de QoS, permitem a compatibilidade de uso com o protocolo RSVP, mas permite também introduzir algumas das boas idéias discutidas em [BUC 2000]

Além destas estruturas de informação, um terceiro módulo de software capaz de realizar a negociação de QoS com as entidades de rede foi também incluído no EPQoSE. Este módulo é chamado de Negociador de QoS (QoS Broker).

4.2.1.1 Descritor de QoS

O descritor de QoS, deve especificar as necessidades de QoS da aplicação H.323 em termos de atraso fim-a-fim, jitter e perda de pacotes.

Como definido em [BUC 2000] diferentes aplicações vão demandar que um ou mais destes parâmetros sejam especificados e controlados adequadamente. O formato escolhido permite que a aplicação sinalize sob demanda os parâmetros, ou utilize parâmetros especificados anteriormente através de um contrato de acordo de nível de serviço. O descritor de QoS é composto dos seguintes campos:

- DefaultMaxDelay – Este parâmetro deve receber um valor verdadeiro, para especificar que o valor default de atraso máximo deve ser utilizado. Este valor é especificado por um acordo estabelecido anteriormente entre as entidades envolvidas. Quando este valor é instanciado, o valor de MaxDelay deve ser configurado como zero (0).
- MaxDelay – Quando o valor de DefaultMaxDelay é configurado em “falso”, o valor de MaxDelay deve ser usado para especificar o valor numérico de atraso máximo.
- DefaultMaxDelayVariation – Este parâmetro deve receber um valor verdadeiro, para especificar que o valor default de variação máxima de atraso deve ser utilizado. Este valor é especificado por um acordo estabelecido anteriormente entre as entidades envolvidas. Quando este valor

é instanciado, o valor de MaxDelayVariation deve ser configurado como zero (0).

- MaxDelayVariation – Quando o valor de DefaultMaxDelayVariation é configurado em “falso”, o valor de MaxDelayVariation deve ser usado para especificar o valor numérico de variação máxima do atraso.
- DefaultMaxPacketLoss – Este parâmetro deve receber um valor verdadeiro, para especificar que o valor default de perda máxima de pacotes deve ser utilizado. Este valor é especificado por um acordo estabelecido anteriormente entre as entidades envolvidas. Quando este valor é instanciado, o valor de MaxPacketLoss deve ser configurado como zero (0).
- MaxPacketLoss – Quando o valor de DefaultMaxPacketLoss é configurado em “falso”, o valor de MaxPacketLoss deve ser usado para especificar o valor numérico de perda máxima de pacotes.
- MaxBandwith – Quando o desejo da aplicação é sobre a quantidade de largura de banda disponibilizada, este parâmetro deve ser usado para especificar o valor numérico de largura de banda requisitada.

4.2.1.2 *Descritor de Tráfego*

O descritor de tráfego deve refletir a visão do terminal H.323, sobre o tráfego a ser gerado por ele, isto é, é uma descrição do tráfego a receber tratamento com QoS. Este tráfego deve ser sinalizado para o QoSM para a realização de controle de recursos, e controle de admissão. Ele representa um comprometimento do terminal sobre as características do tráfego a ser enviado para a rede e por isso, o nível de QoS só será mantido (garantido) pela rede se o fluxo gerado estiver de acordo com o descritor de tráfego informado pela aplicação. O descritor de tráfego é formado pelos seguintes parâmetros:

- TokenBucketRate – Este valor é estabelecido pela origem e representa juntamente com o parâmetro TokenBucketSize a visão que o terminal possui do tráfego gerado. Este parâmetro especifica a taxa em bytes por segundo que o terminal se compromete a enviar dados para a rede. Caso o terminal enviar dados para a rede, a uma taxa maior, a rede não estará apta a garantir o nível de QoS requerido.
- TokenBucketSize – Este valor é estabelecido pela origem e representa juntamente com o parâmetro TokenBucketRate a visão que o terminal possui do tráfego gerado. Este parâmetro especifica em bytes o tamanho, máximo a ser utilizado pelo algoritmo de token bucket, para o tamanho do bucket.
- PeakDataRate – É a taxa máxima que o terminal está planejando enviar dados. Juntamente com o parâmetro MaximumPacketSize, o tamanho máximo de uma rajada de dados que poderá ser enviada para a rede.
- MaxPacketSize – Especifica o tamanho máximo dos datagramas IP que devem ser enviados pela aplicação.

4.2.1.3 *Negociador de QoS*

Uma vez que as necessidades de QoS e o descritivo das características dos fluxos foram informados pela aplicação, o EPQoSE deverá iniciar um processo para a negociação de QoS com o QoSM. Esta tarefa deverá ser realizada pelo negociador de

QoS que vai informar ao QoSM todas as necessidades de QoS e características do tráfego.

Da mesma forma, o negociador de QoS, deverá receber do QoSM informações a respeito da alocação de QoS nos recursos da rede. A rede poderá informar que a requisição foi aceita, ou informar que apenas o nível de serviço “default” poderá ser utilizado.

A comunicação entre o EPQoSE e o QoSM poderá utilizar qualquer protocolo capaz de sinalizar estas necessidades de QoS entre as entidades envolvidas. Pode-se por exemplo utilizar SNMP ou RSVP.

4.2.2 Representação do modelo

A seguir representamos como o modelo foi projetado para funcionar, bem como o relacionamento dos componentes do EPQoSE e entre o EPQoSE e o terminal H.323. Para esta representação utilizamos a notação UML versão 1.4, ilustrado na figura 4.2. Esta figura ilustra as classes principais que compõe o ambiente de colaboração visual H.323 desenvolvido e os componentes do EPQoSE.

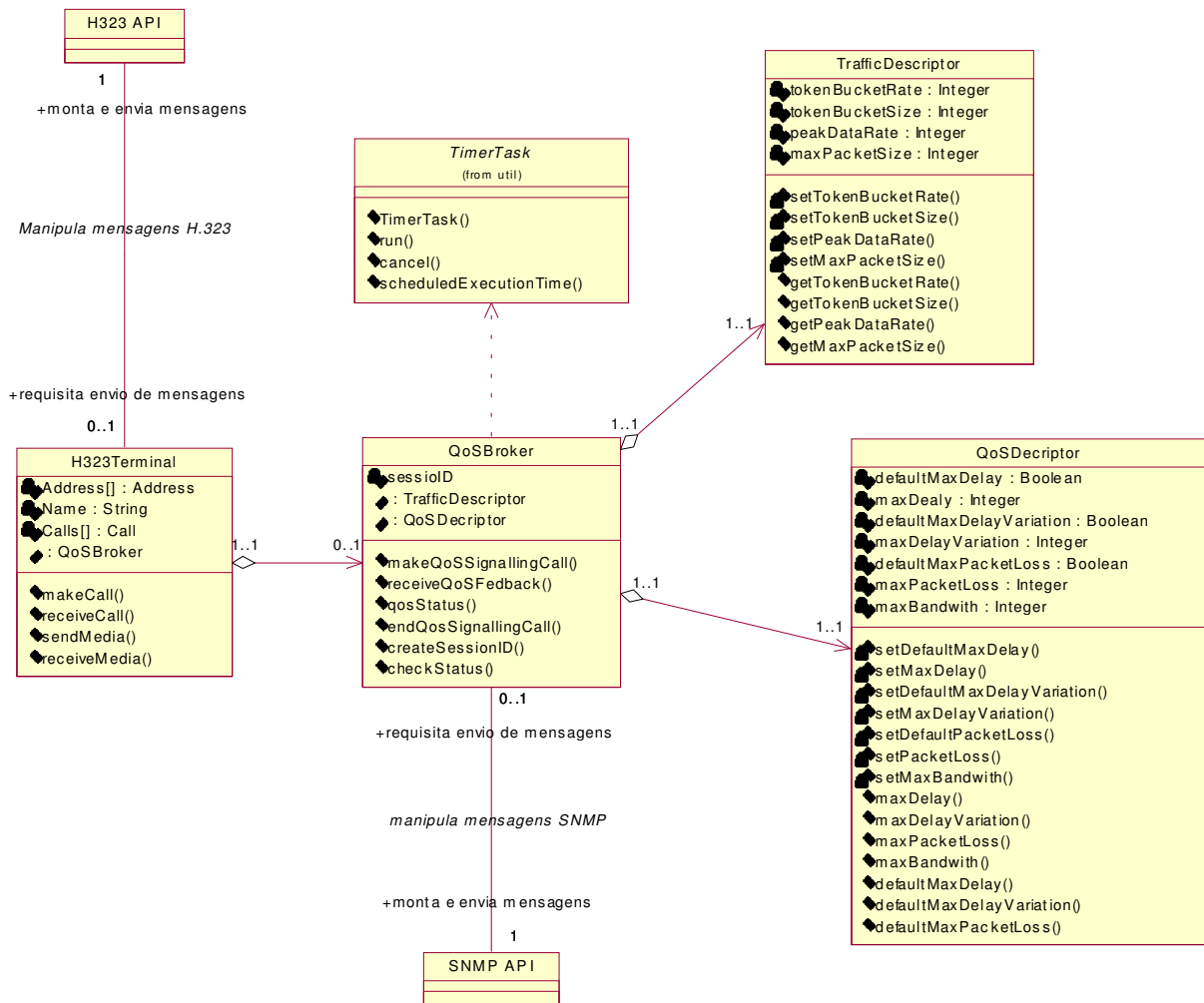


FIGURA 4.2 – Diagrama de classes do EPQoSE

O EPQoSE é dividido em três classes principais: QoSDescriptor, TrafficDescriptor e QoSBroker, sendo que esta última classe utiliza, uma API SNMP para se comunicar com as entidades externas de gerenciamento de QoS como por exemplo o QoSM. A API SNMP também é utilizada pelo QoSBroker para receber notificações do QoSM sobre o sucesso ou falha de alocação de QoS. Como a API é utilizada somente para formatar as mensagens SNMP, não nos interessa neste texto modelar a sua composição e comportamento. O QoSBroker, também utiliza um timer para marcar as funções de expiração de tempo necessárias.

A classe QoSDescriptor é utilizada para especificar as necessidades de QoS do terminal, conforme especificado em 4.2.1.1. Através desta classe é que os requisitos de QoS do terminal serão informados para o QoSBroker, que irá em última instância, comunicar estes parâmetros com outras entidades de rede como o QoSM.

A classe TrafficDescriptor é utilizada para armazenar e manipular os parâmetros de tráfego a ser gerado pelo terminal H.323, conforme especificado em 4.2.1.2. Através desta classe é que os parâmetros que descrevem a característica do tráfego gerado pelo terminal serão informados para o QoSBroker, que irá em última instância, comunicar estes parâmetros com outras entidades de rede como o QoSM.

Estas duas classes, não apresentam grande complexidade e servem apenas para que o terminal H.323 possa especificar as suas necessidades de QoS e descrever as características do tráfego a ser gerado pelo terminal. Os parâmetros definidos por estas classes podem ser direcionados para as entidades de QoS da rede, através de protocolos como o SNMP ou o RSVP. No caso do terminal descrito neste texto usamos o SNMP, conforme será explicado mais adiante.

A classe QoSBroker, é a responsável por implementar a comunicação do terminal com as entidades de QoS da rede e ao mesmo tempo informar o terminal H.323 sobre as respostas da rede em relação ao sucesso ou falha da alocação de QoS pela rede, na fase inicial de comunicação entre terminais H.323, bem como por informar as entidades de QoS da rede sobre o término de uma sessão H.323 e assegurar que estas entidades receberam a notificação de encerramento da sessão, no final da comunicação H.323.

No caso da aplicação, descrita neste texto, utilizamos uma API SNMP, para a comunicação com as entidades de QoS de rede. Desta forma, é responsabilidade da classe QoSBroker, recuperar os valores dos parâmetros do descritor de QoS e do descritor de tráfego das classes QoSDescriptor e TrafficDescriptor, respectivamente e formatar as mensagens SNMP disponibilizadas pela API SNMP, para enviar as mensagens de sinalização de requisição de QoS para as entidades de QoS da rede, como o QoSM.

Para que este mecanismo de comunicação seja possível de ser implementado, o QoSM deve implementar uma MIB SNMP, na qual estes valores podem ser modificados pelo terminal H.323. Esta MIB está especificada e pode ser encontrada no anexo 1 deste texto.

4.3 Funcionamento do modelo

Neste modelo, antes de se iniciar qualquer tipo de comunicação, deve ser configurado no terminal H.323, os parâmetros de QoS e descritor de tráfego, o terminal deseja utilizar. Caso nenhum parâmetro for configurado, o terminal irá realizar a chamada sem QoS.

Uma vez configurados os valores, o terminal poderá iniciar a chamada da mesma forma como os terminais fazem atualmente. Caso o terminal esteja configurado para se

comunicar com um gatekeeper esta comunicação também é realizada, da mesma forma como especificado em [ITU 98].

Caso o terminal tenha sido configurado para requisitar da rede algum nível de QoS superior ao ‘best effort’, o EPQoSE deverá iniciar um processo para a negociação de QoS com o QoSM.

Durante a fase de negociação com o QoSM, este irá informar se existem recursos suficientes para que a rede possa atender a requisição de QoS. Conforme esta resposta, o terminal poderá decidir em continuar com a chamada no nível ‘default’, ou optar por não realizar a chamada. O nível ‘default’, para a maioria dos casos será o tradicional nível ‘best effort’ oferecido atualmente pela Internet. Entretanto, para alguns casos, poderá existir um acordo de nível de serviço, com o provedor, especificando em um contrato, que o nível ‘default’ é um nível superior ao ‘best effort’.

Por exemplo, supondo que um provedor de serviço ofereça diferentes níveis de serviços para seus clientes, baseado no modelo Olímpico: ouro, prata, bronze, onde todas estas opções representam níveis de serviço superiores ao ‘best effort’ oferecido normalmente.

Este provedor de serviços possui um determinado cliente x, que utiliza intensamente recursos de rede para acesso a Internet e realização de downloads, mas eventualmente utiliza a conexão para a realização de conferências através da Internet. Como estas conferências são importantes para o seu negócio, ele necessita que este serviço tenha um nível de qualidade, superior ao tradicional ‘best effort’.

Por esta razão este cliente, estabelece com o provedor, um contrato de fornecimento de serviços de acesso a rede, onde especifica que seu tráfego TCP e UDP deve ser tratado de acordo com o nível de serviço bronze, em qualquer momento. Mas para a realização das conferências, poderá ser requisitado da rede um nível de serviço, prata. Desta forma, o nível de serviço ‘default’ estabelecido em contrato é bronze tanto para tráfego TCP, como para tráfego UDP, mas o provedor de serviço possui cadastrado em seu servidor de políticas, que o usuário x, poderá requisitar um nível de serviço prata para tráfego UDP a qualquer momento. Esta necessidade deve ser sinalizada para o provedor dinamicamente, no momento que o cliente deseja usar este ‘upgrade’ de serviço.

No momento em que este cliente decidir realizar a conferência ele poderá ou não indicar uma necessidade para requisitar tráfego prata. Se o usuário requisitar este upgrade de serviço para tráfego prata ele deverá indicar esta necessidade para o provedor, através dos mecanismos de negociação realizados pelo negociador de largura de banda. Caso o provedor tenha os recursos disponíveis ele poderá aceitar esta requisição e tarifar o usuário através deste novo nível de serviço, durante o tempo utilizado. Caso não existirem recursos suficientes para que o provedor possa oferecer este nível de serviço, ele deverá obrigatoriamente oferecer o nível de serviço ‘default’ (bronze) para este cliente, que continua sendo superior ao ‘best effort’.

4.3.1 Fase de sinalização de QoS

Para que o modelo possa funcionar conforme explicado até agora, uma nova fase de comunicação deverá fazer parte da comunicação entre o terminal H.323 e outras entidades funcionais como o QoSM.

Estes procedimentos devem ser inseridos antes do terminal enviar a mensagem Setup para o destino, para impedir que a sessão seja encerrada logo após o terminal destino ter atendido a chamada, devido à escassez de recursos, como descrito na sessão 3.7. Uma visão geral do processo pode ser visualizada na figura 4.3. Nesta figura é

inserida a fase de sinalização de QoS e logo após esta fase ser encerrada é que o terminal irá iniciar os procedimentos de estabelecimento de chamada H.323.

Para que este processo possa ser realizado, é necessário que o terminal H.323 esteja apto a sinalizar suas necessidades de QoS para a rede. Estas são as características adicionadas ao terminal H.323 aqui proposto, através do módulo EPQoSE.

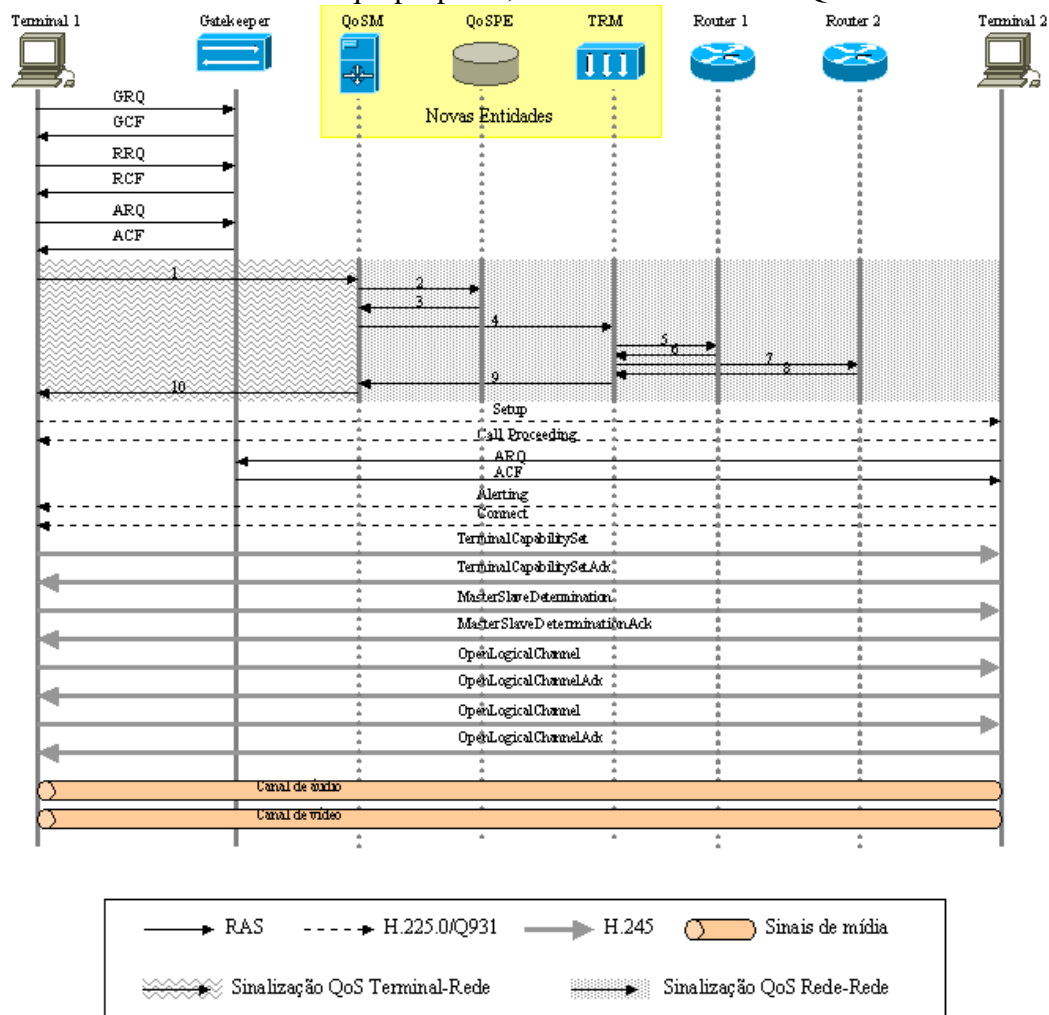


FIGURA 4.3 – Fase de sinalização de QoS

Como pode ser visto, existem diversas mensagens que compõem a fase de sinalização das necessidades de QoS e basicamente elas podem ser divididas em mensagens que transitam entre os terminais H.323 e a rede (sinalização terminal-rede) e mensagens que transitam apenas entre os elementos de rede da solução (sinalização rede-rede). A proposta do terminal H.323 discutido neste texto contempla as mensagens terminal-rede, pois diz respeito ao mecanismo de requisição de QoS do terminal para a rede. A comunicação entre os dispositivos de rede, incluindo o QoS SM, QoS SPE, TRM e roteadores podem ser realizadas através de protocolos já existentes como SNMP / COPS ou ainda através de configuração via CLI através de sessões de Telnet ou SSH e não serão discutidas em detalhes no texto. Algumas propostas para esta comunicação (sinalização rede-rede) podem ser encontradas em [MAH 2000] e [ALM 2003].

A seguir descrevemos passo-a-passo, como se dará o processo de sinalização de QoS entre o terminal e os dispositivos de rede para a inicialização de chamadas e o encerramento de chamadas. A próxima sessão detalha como o terminal foi implementado para realizar as funções descritas genericamente até aqui.

Tomando-se por base a ilustração da figura 4.3, os passos para obtenção de QoS são os seguintes:

1. O Terminal envia os parâmetros de QoS e o descritor de tráfego ao QoSM.
2. O QoSM consulta o servidor de políticas (QoSPE), para verificar se existe alguma política, que permita que o terminal 1 se comunique com o terminal 2 com nível de QoS especificado pelos parâmetros de QoS e descritor de tráfego, enviados pelo terminal.
3. O servidor de políticas responde ao QoSM, sobre a existência ou não da política. Permitindo ou não que a comunicação continue.
4. Caso a resposta do QoSPE for positiva, o QoSM contacta o TRM com os parâmetros de QoS e descritor de tráfego para o fluxo, a ser configurado nos dispositivos de rede. Caso contrário, os passos de 5 a 9 não serão executados.
5. O TRM é o responsável por configurar corretamente todos os dispositivos de rede do domínio, sejam eles switches, roteadores ou firewalls. O TRM deve estar apto a transformar os parâmetros de QoS e descritor de tráfego informados pelo terminal, em uma sintaxe compreendida por estes dispositivos de rede. Para tal, o TRM poderá se comunicar com estes dispositivos, através de SNMP, COPS, sessões de telnet ou outra forma de comunicação, instruindo o dispositivo para configuração apropriada dos parâmetros de QoS concedidos para esta comunicação. O TRM faz as configurações para todos os dispositivos de rede do domínio.
6. O dispositivo deve responder ao TRM sobre o sucesso ou falha da configuração realizada.
7. Idem ao passo 5.
8. Idem ao passo 6.
9. Depois que todos os dispositivos foram configurados o TRM deve informar ao QoSM sobre o sucesso ou falha na configuração dos dispositivos. Esta fase é muito importante, porque qualquer falha na configuração dos dispositivos poderá resultar em um nível de QoS abaixo do esperado pelo terminal e poderá comprometer a comunicação.
10. O QoSM irá indicar para o terminal o estado de sucesso ou falha da alocação de QoS solicitada pelo terminal. Esta resposta permitirá ao terminal saber, qual o nível de QoS será concedido para esta sessão e reagir apropriadamente.

Uma vez configurados os dispositivos de rede, a comunicação irá prosseguir normalmente, até que um dos terminais resolva desligar. Quando isto acontecer, aquele terminal que iniciou a chamada, terminal 1, neste caso, deverá indicar o encerramento da chamada para o QoSM, para que este possa desalocar os recursos de rede até então alocados para esta sessão, disponibilizando estes recursos para outras sessões.

4.4 Implementação do ambiente

A implementação do ambiente H.323 com o módulo EPQoSE foi realizada em Java. A escolha da linguagem foi feita por diversos fatores. Conforme apontado em [BHA 2000], a linguagem Java vêm sendo cada vez mais utilizada no desenvolvimento de aplicações portáteis que rodam em cima da Internet, da rede de telefonia, de redes sem fio e de redes inteligentes. Uma prova disso é o desenvolvimento do JAIN [SUN 2002]. Assim, usando esta linguagem se torna mais fácil adaptar o Terminal para rodar em diferentes plataformas.

Outros fatores, também sustentam a escolha de se implementar o ambiente na linguagem Java. Um destes fatores foi a existência de APIs de comunicação H.323 e SNMP que pudessem ser utilizadas, de forma a facilitar o trabalho. Como o objetivo final é construir um ambiente H.323 (terminal) com capacidade para se comunicar com entidades externas (QoSM), através do modelo proposto neste texto, e não a implementação das pilhas de protocolos, a existência de pilhas de protocolos H.323 e SNMP em Java, facilitam bastante a tarefa de construção do ambiente.

Para implementar as funções H.323, conforme ilustradas na figura 4.1 e descritas no capítulo 2 deste texto, utilizamos uma pilha do protocolo H.323 desenvolvida pelo laboratório de desenvolvimento de tecnologias emergentes da IBM, o Alphaworks. Esta pilha de protocolo, chamada de engine J323, é um software implementado em Java, que desenvolve as funções de controle de chamada e controle de mídia para um terminal H.323 de forma automática. Esta “engine” implementa as funções H.323 através da API de telefonia Java (JTAPI) permitindo que os desenvolvedores possam integrar as funcionalidades de um terminal H.323 em suas aplicações [DUP 2001].

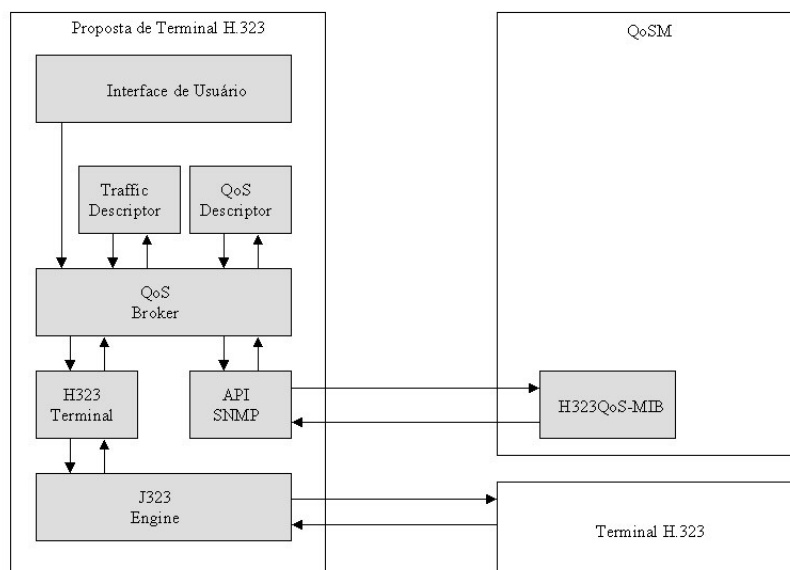


FIGURA 4.4 – Componentes introduzidos no modelo proposto

Para a comunicação entre o EPQoSE e o QoSM, optamos pela utilização do protocolo SNMP. Desta forma, as funções de negociação proporcionadas pelo módulo EPQoSE foram implementadas usando uma API SNMP da AdvetNet. Esta API é desenvolvida na linguagem Java e representa uma das mais completas implementações do protocolo SNMP disponíveis [ADV 2002].

Como pode ser visualizado na figura 4.4, o correto funcionamento do modelo implica na comunicação do terminal com o QoSM. Esta comunicação é realizada através de SNMP e por isso o QoSM deve implementar um agente SNMP e utilizar uma MIB, capaz de armazenar as informações de requisições de QoS dos terminais H.323 (EPQoSE). Estes detalhes são explicados mais adiante na sessão 4.4.5 e a MIB é apresentada no anexo 1 deste texto.

A seguir descreveremos as características dos principais módulos que compõe o ambiente de colaboração visual H.323 capaz de sinalizar suas necessidades de QoS para a rede, adicionando Qualidade de Serviço à comunicação H.323.

4.4.1 O QoSM

Para realizar o papel de QoSM, aproveitamos o ambiente de gerencia QAME (QoS-Aware Management Environment) [GRA 2001], já existente, que é responsável pela gerência da rede metropolitana Metropoa [PRO 2002]. Assim, nosso trabalho define apenas as adições necessárias a este ambiente, conforme ilustrado na figura 4.4 (H323QoS-MIB).

Este ambiente – o QAME – implementa um sistema de gerenciamento de rede baseado em políticas, conforme ilustrado na figura 3.7. O ambiente implementa um PDP, que é responsável por traduzir as políticas definidas pelo administrador em ações de configuração nos dispositivos da rede. Especificamente este PDP suporta políticas de QoS e traduz as políticas para ações em roteadores CBQ [RIS 2001]. O sistema é composto pelos seguintes componentes: NMS, PR, PDP e PEP.

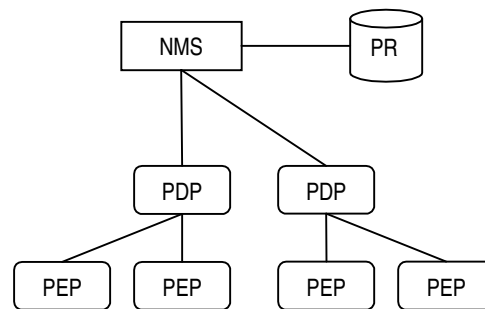


FIGURA 4.5 – Arquitetura PBNM padrão

O NMS é utilizado pelo administrador de rede para criar, editar, e remover políticas de um repositório, além de executar todas as outras tarefas de gerência convencionais. É também através da NMS que o administrador coordena o uso das políticas do repositório, aplicando certas políticas do repositório e editando ou removendo outras.

Funcionalmente, o repositório de políticas, é uma base de dados que armazena apenas as políticas criadas pelo administrador. Outras informações de gerência que não as políticas devem ser armazenadas em bases diferentes do repositório. Fisicamente,

entretanto, as políticas e outras informações de gerência podem se encontrar em um mesmo dispositivo, ou até mesmo em uma mesma base.

Os PDPs (*Policy Decision Points*) são os elementos responsáveis por traduzir as políticas definidas pelo administrador em ações de configuração e tomada de decisão nos dispositivos. Na verdade, as ações de configuração não visam apenas configurar o comportamento dos dispositivos como um todo, mas sim objetivam moldar o comportamento de elementos internos aos dispositivos: os PEPs (*Policy Enforcement Points*). Um PEP é um elemento onde as políticas deverão ser respeitadas, são os dispositivos de rede que devem ser configurados. Os PEPs, diferentemente dos PDPs, são incapazes de compreender as políticas de redes, e portanto precisam ser regidos pelos PDPs, estes sim capazes de entender os objetivos expressos nas políticas.

Internamente, o PDP do sistema é subdividido em duas partes, sendo uma parte de genérica de controle e outra parte específica que atua junto aos PEPs. A parte de controle interage com o sistema de gerenciamento recebendo URLs das políticas que devem ser aplicadas e fazendo o *download* das mesmas a partir do repositório de políticas. A parte genérica de controle, após receber uma política, também verifica os momentos corretos para ativação e desativação das políticas de acordo com as informações de agendamento de cada política. A parte genérica de controle é idêntica para todos os PDPs independentemente do tipo de política suportada por cada PDP.

A segunda parte, esta específica, é diferente para cada PDP desenvolvido. Tal parte é responsável por traduzir a política a ser aplicada e fazer a comunicação com o PEP associado usando um protocolo de gerência particular. Como PEPs diferentes exigem protocolos de comunicação diferentes, a segunda parte de um PDP é também diferente. A figura 4.6 exemplifica tal situação, onde PG é uma sigla para ‘parte genérica’ e PE é uma sigla para ‘parte específica’.

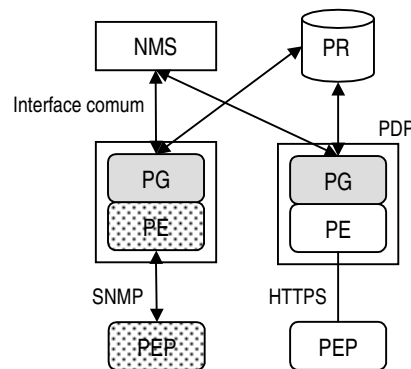


FIGURA 4.6 – Componentes do PDP

É importante notar que com esta arquitetura, a parte específica do PDP é unicamente responsável pelos serviços de tradução, já que o suporte ao agendamento, transferência e ativação/desativação de políticas já são executados pela parte genérica. É também importante verificar que a parte genérica pode controlar a ativação/desativação de qualquer tipo de políticas (e.g. QoS, segurança, multicast, etc.) pois as únicas informações acessadas são aquelas relativas ao agendamento. As informações específicas de cada tipo de política são avaliadas na parte específica do PDP.

A parte específica do PDP deve implementar não somente o suporte a um tipo de políticas, mas também conhecer o protocolo e tecnologia do PEP associado. Por exemplo, para políticas de QoS, a parte específica deve ter conhecimento dos parâmetros de QoS definidos internamente a cada política. Entretanto, para uma mesma

política, os parâmetros podem ser traduzidos para ações diferentes, caso os PEPs alvos implementarem tecnologias para fornecimento de QoS diferentes.

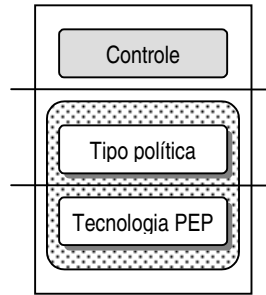


FIGURA 4.7 – Componentes da parte específica do PDP

A implementação do PDP aqui descrito, foi realizada através de um conjunto de classes Java e uma interface que permite a comunicação entre as partes genérica de controle e específica da arquitetura do PDP.

Conforme mostrado rapidamente, o QAME é composto por um conjunto de componentes que formam o sistema de gerenciamento baseado em políticas. Neste sentido usamos o QAME como se fosse o QoSM, o QoSPE e o TRM, como segue:

- As funções desempenhadas pelo QoSM, são realizadas pela parte Genérica do PDP, exceto a comunicação com o terminal H.323, via SNMP que teve de ser implementada na parte específica do PDP, para manter a consistência com a estrutura proposta em [ALM 2003].
- As funções do QoSPE são implementadas pelo PR (figura 4.6).
- As funções do TRM são implementadas na parte específica do PDP.

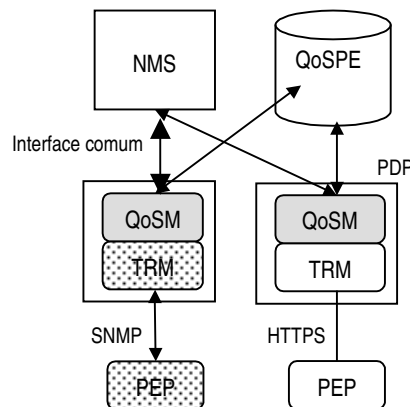


FIGURA 4.8 – Mapeamento entre entidades funcionais do sistema QAME para entidades funcionais de QoS H.323

4.4.2 Necessidades de implementação no QoSM

Para interagir com o Terminal H.323 proposto, o QoSM deverá estar preparado para (1) receber as mensagens de requisição de QoS do terminal (EPQoSE), (2) interpretar estas requisições adequadamente (3) verificar se as políticas da organização permitem que a requisição seja concedida (4) caso a requisição for concedida, comunicar ao TRM os parâmetros de descritor de tráfego e descritor de QoS para serem

configurados nos dispositivos de rede. (5) sinalizar o estado da requisição ao terminal H.323.

Esta sessão descreve como o QoSM pode receber as informações do EPQoSE. A sessão também dá uma breve explicação das ações a serem tomadas pelo QoSM sempre que alguma informação for passada pelo EPQoSE.

Uma vez que optamos pela utilização do SNMP, o QoSM deve implementar um agente SNMP e utilizar uma MIB, capaz de armazenar as informações de requisições de QoS dos terminais H.323 (EPQoSE), por isso definimos uma MIB SNMP chamada de H323QoS-MIB, encarregada de armazenar todas as requisições de QoS realizadas para o QoSM. A definição desta MIB pode ser vista no Anexo 1 deste texto e a figura 4.4 representa a idéia.

Pelo fato de ser possível que diversos terminais H.323 façam solicitações de QoS ao mesmo tempo, a definição da MIB utiliza tabelas, onde cada linha de uma tabela representa uma certa requisição de QoS em específico identificados por um identificador (ID) e onde cada tabela armazena uma parte da informação completa da requisição de QoS de uma sessão. A informação completa sobre uma requisição de QoS é identificada pelo objeto ID de cada tabela e pelos endereços IP da origem e destino.

Assim, uma requisição de QoS completa é formada por três tipos de informação: informações de controle, informações sobre o descritor de QoS e informações sobre o descritor de tráfego. As informações de controle ficam armazenadas na tabela “qosControlTable”, as informações sobre o descritor de QoS ficam armazenados na tabela “qosDescriptorTable” e as informações sobre o descritor de tráfego ficam armazenadas na tabela “trafficDescriptorTable” conforme visto na figura 4.9.

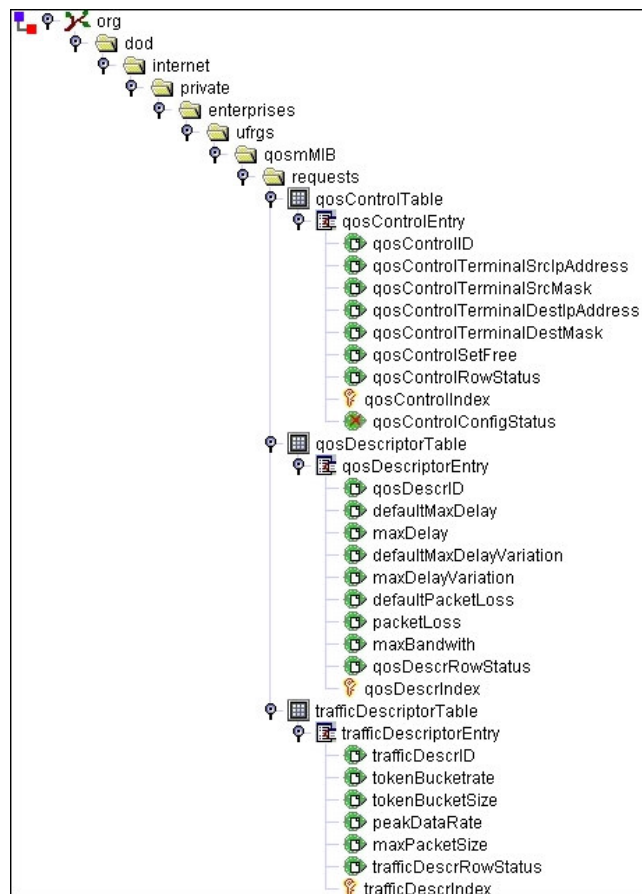


FIGURA 4.9 – H323QoS-MIB

4.4.2.1 *Processo de Alocação de QoS*

Quando um terminal inicia a requisição de QoS para o QoSM ele deverá incluir uma nova linha em cada tabela que compõe a MIB. Estas três linhas terão um mesmo número identificador (ID) armazenados nos seguintes objetos: qosControlID, qosDescrID e trafficDescrID e formam a requisição completa de QoS para esta sessão.

Quando estas tabelas passarem para o estado active (1), o que pode ser verificado através dos objetos RowStatus presentes em cada tabela, o QoSM deverá setar o valor do objeto qosControlConfigStatus para o estado requested (1).

A partir daí, o QoSM deverá executar as funções descritas anteriormente, na sessão 3.6.1, como consultar o servidor de políticas, e avisar ao TRM que novas políticas devem ser configuradas, receber notificação do TRM sobre o sucesso ou falha da alocação de QoS nos dispositivos intermediários, etc.

Caso a requisição não seja aceita pelo servidor de políticas, o valor do objeto qosControlConfigStatus deve ser configurado para o estado failed (4). Caso a requisição tenha sido aceita pelo servidor de políticas, o QoSM deverá indicar ao TRM os parâmetros de QoS e descritor de tráfego, para que este inicie a configuração dos dispositivos de rede do domínio. Se a alocação de recursos ocorreu em todos os dispositivos sem nenhum problema, o QoSM deverá setar o valor do objeto qosControlConfigStatus para o estado success (3). Se houve qualquer problema durante o processo de configuração dos dispositivos ou caso as políticas não tenham sido implementadas em pelo menos um dispositivo do domínio, o valor do objeto qosControlConfigStatus deve ser configurado para o estado failed (4).

A partir de consultas ao objeto qosControlConfigStatus, o terminal H.323 poderá decidir em continuar com a sessão ou encerrar a sessão.

4.4.2.2 *Processo de desalocação de QoS*

Quando o Terminal H.323 (EPQoSE), sinalizar para o QoSM que a sessão encerrou, ele fará isso configurando o valor do objeto qosControlSetFree em TRUE. Neste momento, o QoSM deve setar qosControlConfigStatus para o estado setfree (5), desta forma o terminal H.323 poderá saber que sua requisição foi recebida pelo QoSM com sucesso. Então, o QoSM deverá avisar ao TRM para reconfigurar os dispositivos de rede de forma a liberar os recursos previamente alocados para esta sessão. Quando o TRM avisar o QoSM sobre a liberação dos recursos, o QoSM deve setar o estado do objeto qosControlRowStatus, qosDescrRowStatus e TrafficDescrRowStatus para destroy (6) e finalmente o Terminal H.323 que requisitou a chamada deverá remover todas as linhas relacionadas a esta requisição das respectivas tabelas.

4.4.3 *A classe H323Terminal*

A classe denominada H323Terminal, fica responsável por inicializar todas as configurações da API J323 bem como informar ao TrafficDescriptor e ao QoSDescriptor as necessidades de QoS da aplicação. Além disso, qualquer tipo de implementação adicional referente às funções H.323 que são desempenhadas pela “engine” J323 será implementada nesta classe.

Como a “engine” J323 é uma implementação da API de Telefonia Java, ela esconde todos os detalhes de implementação H.323 do desenvolvedor, por isso, apenas interfaces de alto nível são disponibilizadas por este módulo de software.

A classe `H323Terminal` agrupa todas as funcionalidades implementadas pelo `J323` e usa estas facilidades para estabelecer a lista de endereços do terminal `H.323`, o nome do terminal `H.323`, a lista de chamadas ativas que o terminal está participando entre outras informações necessárias para que o terminal possa se comunicar.

Esta classe também é a responsável por iniciar uma chamada e enviar os sinais de mídia ao terminal com o qual está se comunicando, bem como deve ser capaz de receber os sinais de mídia do terminal remoto e apresenta-las ao usuário.

Caso o usuário sinalizar ao `H323Terminal` que deseja realizar uma chamada com QoS, o valor da variável `qosEnabled` deverá ser configurado em `TRUE`, o que significa que o Terminal deverá se comunicar com o QoSM através da fase de sinalização de QoS descrita na sessão 4.3.1. Caso o valor desta variável for `FALSE`, o `H323Terminal` irá realizar uma chamada `H.323` convencional sem qualquer comunicação com o QoSM. Neste caso, os procedimentos de conexão descritos no capítulo 2, são seguidos.

Assim, quando o `H323Terminal` é instruído para fazer uma chamada com QoS ele instancia o módulo `QoSBroker` e informa os valores de descritor de QoS e descritor de tráfego a serem usados para esta sessão. A partir daí o `QoSBroker` deverá se comunicar com o QoSM para negociar o nível de QoS requerido.

Após a fase de sinalização de QoS se encerrar, o `H323Terminal` faz uma chamada através do método `makeCall()`. Este método é responsável por realizar toda a fase de sinalização `H.225.0-Q931` e `H.245` explicados no capítulo 2. Quando existir um gatekeeper envolvido na comunicação, este método irá realizar a comunicação `H.225.0-RAS` com o respectivo gatekeeper. Após a chamada ser estabelecida e a negociação `H.245` ter sido realizada, o `H323Terminal` deve iniciar o envio de seus sinais de mídia através do método `sendMedia()`.

Os métodos `receiveCall()` e `receiveMedia()` são responsáveis por manipular as chamadas entrantes, bem como, a entrega dos sinais de áudio e vídeo ao usuário da aplicação, respectivamente. O método `receiveCall()` está permanentemente escutando por possíveis chamadas e assim que aceitar uma chamada, aciona o método `receiveMedia()`, para que este comece a manipular os sinais que estão chegando e o método `sendMedia()` para enviar os sinais de áudio e vídeo.

4.4.4 A classe *QoSDescriptor*

A classe `QoSDescriptor` é bastante simples e implementa a estrutura de dados descrita no item 4.2.1.1. A sua tarefa é representar o descritor de QoS que a aplicação está requisitando ao QoSM.

Os métodos `set`, da classe servem para configurar os novos valores para as variáveis `defaultMaxDelay`, `maxDelay`, `defaultMaxDelayVariation`, `maxDelayVariation`, `defaultMaxPacketLoss`, `maxPacketLoss` e `maxBandwith`. Todos os outros métodos servem para recuperar o valor atual das variáveis.

É importante salientar que o método deve realizar a checagem de valores para a entrada dos dados, conforme descrito no item 4.2.1.1. Isto é, se o valor de `defaultMaxDelay` é configurado em `TRUE`, o valor de `maxDelay` deve ser 0, e assim por diante.

4.4.5 A classe *TrafficDescriptor*

A classe *TrafficDescriptor* é bastante simples e implementa a estrutura de dados descrita no item 4.2.1.2. A sua tarefa é representar o descritor de tráfego que a aplicação está requisitando ao QoSM.

Os métodos *set*, da classe servem para configurar os novos valores para as variáveis *tokenBucketRate*, *tokenBucketSize*, *peakDataRate* e *maxPacketSize*. Todos os outros métodos servem para recuperar o valor atual das variáveis.

4.4.6 A classe *QosBroker*

É nesta classe que reside toda a inteligência do EPQoSSE. Esta classe é composta pelos descritores de QoS e tráfego. Ela utiliza um timer para funções que requerem controle de tempo e uma API SNMP para executar a comunicação com o QoSM.

Quando instanciada esta classe deverá gerar um número aleatório, para identificar a requisição de QoS. Este número é gerado pelo método *generateSessionID()* e armazenado na variável *CurrentSessionID*. Este número, juntamente com os endereços de origem e destino, identifica unicamente uma requisição de QoS para o QoSM.

O método *qosStatusCheck()*, é responsável por verificar periodicamente o estado atual de uma requisição de QoS em específico. A aplicação faz isso através de *pollings* periódicos na variável *qosControlConfigStatus* da MIB H323QoS-MIB, residente no QoSM. A descrição completa desta MIB é apresentada no anexo 1 deste texto.

Foram definidos seis possíveis estados para uma requisição de QoS: *notStarted* (0), *requested* (1), *inProgress* (2), *success* (3), *failed* (4) e *setfree* (5). O valor inicial desta variável é zero (0), indicando que a requisição ainda não foi realizada. Logo que, o terminal requisitar os parâmetros de QoS para o QoSM o estado deve ser *requested* (1), pois a configuração dos dispositivos ainda não iniciou. Assim que o QoSM iniciar a comunicação com o TRM este estado deve passar para *inProgress* (1), indicando que a configuração está sendo realizada. Se a configuração de todos os dispositivos tiver êxito, o estado deve passar para *success* (3) e caso a configuração em pelo menos um dispositivo falhar o estado deve ser *failed* (4). Quando uma linha da tabela possuir o valor deste objeto no estado *setfree* (5), significa que os recursos de rede podem ser desalocados para esta sessão.

4.4.6.1 Processo de Alocação de QoS

O método *makeQosSignallingCall()*, é o responsável por sinalizar ao QoSM as necessidades de QoS do terminal, bem como o descritor de tráfego. Para isso, ele deve executar três passos: (1) inserir uma nova linha na Tabela “*qosControlTable*”, uma nova linha na Tabela “*qosDescriptorTable*” e uma nova linha na tabela “*trafficDescriptorTable*” da MIB H323QoS -MIB. (2) Configurar apropriadamente os objetos destas três tabelas, de forma a requisitar um determinado nível de QoS para esta sessão H.323. Esta ação é realizada, enviando comandos *snmp SET*, para cada um dos objetos da MIB. (3) Verificar se a requisição de QoS foi devidamente atendida ou se aconteceu alguma falha na alocação dos recursos de rede. O método *makeQosSignallingCall()* implementa estas três ações da seguinte forma:

1. Para inserir uma nova linha nas tabelas “*qosControlTable*”, “*qosDescriptorTable*” e “*trafficDescriptorTable*” ele utiliza o comando *addRow(boolean Status, String[] oidlist, String[] values)* da API SNMP.

2. Para configurar apropriadamente os objetos das tabelas, este método usa o comando *SET*, da API snmp. Como as novas linhas acabaram de ser inseridas na tabela, deve-se agora configurar os objetos de cada uma das tabelas definidas na MIB. A tabela “qosControlTable” irá cuidar das informações genéricas de uma requisição de QoS e deve ser configurada com os seguintes valores:

TABELA 4-1 - Variável da classe QoSBroker a ser configurado na tabela qosControlTable

Variáveis da Classe QoSBroker	Objeto da Tabela qosControlTable
CurrentSessionID	qosControlID

TABELA 4-2 - Endereços IP e máscara de sub-rede a serem configurados na tabela qosControlTable

Endereços IP e máscara de sub-rede	Objeto da Tabela qosControlTable
Endereço IP da Origem	qosControlTerminalSrcIpAddress
Máscara de sub-rede da Origem	qosControlTerminalSrcMask
Endereço IP do Destino	qosControlTerminalDestIpAddress
Máscara de sub-rede do Destino	qosControlTerminalDestMask

TABELA 4-3 - Valores adicionais a serem configurados na tabela qosControlTable

Valor	Objeto da Tabela qosControlTable
0 (valor zero) = FALSE	qosControlSetFree

A linha da tabela QoSDescriptorTable conterá os valores de descritor de QoS para esta sessão, e deve ser preenchida da seguinte forma:

TABELA 4-4 - Variável da classe QoSBroker a ser configurado na tabela qosDescriptorTable

Variáveis da Classe QoSBroker	Objeto da Tabela qosDescriptorTable
CurrentSessionID	qosDescrID

TABELA 4-5 - Variável da classe QoSDescriptor a ser configurado na tabela qosDescriptorTable

Variável da Classe QoSDescriptor	Objeto da Tabela qosDescriptorTable
DefaultMaxDelay	defaultMaxDelay
MaxDelay	maxDelay
DefaultMaxDelayVariation	defaultMaxDelayVariation
MaxDelayVariation	maxDelayVariation
DefaultMaxPacketLoss	defaultPacketLoss
MaxPacketLoss	packetLoss
MaxBandwith	maxBandwith

A linha da tabela TrafficDescriptorTable conterá os valores de descritor de tráfego para esta sessão, e deve ser preenchida da seguinte forma:

TABELA 4-6 - Variável da classe QoSBroker a ser configurado na tabela trafficDescriptorTable

Variáveis da Classe TrafficDescriptor	Objeto da Tabela trafficDescriptorTable
CurrentSessionID	TrafficDescrID

TABELA 4-7 - Variável da classe TrafficDescriptor a ser configurado na tabela trafficDescriptorTable

Variáveis da Classe TrafficDescriptor	Objeto da Tabela trafficDescriptorTable
TokenBucketRate	tokenBucketRate
TokenBucketSize	tokenBucketSize
PeakDataRate	peakDataRate
MaxPacketSize	maxPacketSize

- Após inserir estes valores na MIB do QoSM, o terminal deve verificar periodicamente em intervalos de tempo curtos (a cada 30ms no máximo) o estado da sua requisição. Isto é realizado através do comando *GET*, da API SNMP. Este procedimento é realizado até que o objeto qosControlConfigStatus esteja com o valor igual a success (3) ou failed (4).

4.4.6.2 Processo de desalocação de QoS

Quando algum dos terminais envolvidos na sessão resolver encerrar a sessão, o terminal que requisitou QoS ao QoSM será o responsável por sinalizar ao QoSM que os recursos alocados até então podem ser liberados para outra sessão.

O terminal que requisitou a chamada faz isso através de um comando *SET* da API SNMP. Neste caso o objeto qosControlSetFree deverá ser configurado em TRUE:

TABELA 4-8 - Valores adicionais a serem configurados na tabela trafficDescriptorTable

Valor	Objeto da Tabela trafficDescriptorTable
1 (valor UM) = TRUE	qosControlSetFree

Depois disso, o QoSBroker deverá ser o responsável por retirar das tabelas da MIB, as linhas referentes a esta alocação de QoS. Neste caso ele deverá retirar das tabelas “qosControlTable”, “qosDescriptorTable” e “trafficDescriptorTable”, todas as linhas que possuem ID igual a variável currentSessionID. Ele faz isso utilizando o comando *table.deleteRow(rowStatus.indexvalue)* da API SNMP.

5 Comentários e considerações finais

O protocolo IP e a Internet estão cada vez mais sendo utilizados para o transporte de informações, em qualquer parte do mundo. Antes não se pensava no transporte de sinais de tempo real na Internet, hoje, entretanto, com o avanço das tecnologias já é possível transportar sinais de tempo real na Internet. Entretanto, como colocado por [WYD 2002], o nível de QoS das redes de pacotes “best-effort” de hoje ainda é limitado para as necessidades demandadas por aplicações multimídia e/ou interativas, que possuem requisitos especiais de atraso e largura de banda. Porém, segundo este mesmo autor as pesquisas em andamento na área, sugerem que dentro de pouco tempo, com pequenas modificações nos dispositivos de rede (links) e nas aplicações (fontes), uma abordagem elegante e simples para adoção de QoS poderá ser obtida.

Ainda segundo [WYD 2002], deve existir um tipo de negociação entre a fonte dos sinais (aplicação) e a rede (roteadores), para a obtenção do grau adequado de QoS requerido pela aplicação. Neste sentido, a dificuldade maior está em construir algoritmos capazes de negociar a largura de banda necessária com as aplicações de forma rápida (para minimizar a percepção do atraso), chegando a um equilíbrio entre as necessidades da aplicação e os recursos existentes na rede. Conforme descrito, [WYD 2002] também aponta que um dos principais desafios é construir um sistema de controle, robusto e rápido o suficiente, capaz de negociar rapidamente os parâmetros de QoS com a aplicação e com a rede.

5.1 Contribuições

Nosso trabalho apresenta uma abordagem simples nesta direção, onde propomos um modelo para o desenvolvimento de um ambiente de colaboração visual baseado no padrão H.323, capaz de negociar com entidades de rede um certo nível de qualidade de serviço, adequado às necessidades do usuário. O modelo completo deve ser formado por um terminal H.323 que implemente a funcionalidade de requisitar QoS para a rede, bem como por outras entidades de rede, responsáveis por controlar as requisições de QoS. Estas entidades são: o QoSM, o QoSPE e o TRM, também explicadas no texto. Nesta abordagem é essencial que exista uma forma através da qual o terminal H.323 se comunique com o QoSM para requisitar o nível de QoS desejado. Esta comunicação entre o terminal H.323 e o QoSM foi o tema central apresentado no trabalho.

Os principais componentes do modelo são o terminal H.323, que implementa um componente adicional chamado de EPQoSE, responsável por emitir requisições de QoS em benefício do terminal e o QoSM, entidade de rede responsável por controlar as requisições de QoS e interagir com outras entidades de rede para alocação de QoS na rede.

No modelo proposto, o EPQoSE é formado por três módulos principais: o descritor de QoS, o descritor de Tráfego e o Negociador de QoS (QoS Broker). A comunicação entre o EPQoSE e o QoSM foi realizada através do protocolo SNMP, por isso, o QoSM implementa um agente SNMP e uma MIB, que também foi definida e apresentada como parte do trabalho.

Vimos que o negociador de QoS implementa as funcionalidades mais complexas do EPQoSE por que é responsável por toda a comunicação com o QoSM. Para realizar a comunicação usamos uma API SNMP, embora pudéssemos ter utilizado outro protocolo como o RSVP. A escolha pelo SNMP é explicada pela maturidade do protocolo e pela

existência de uma API de fácil manipulação, já que o intuito principal foi o de desenvolver o modelo proposto. Desta forma, sempre que o terminal H.323 deseja realizar uma chamada com QoS, o negociador de QoS deve se comunicar com o QoSM através do SNMP e configurar alguns valores na MIB implementada pelo QoSM. A partir dos valores configurados na MIB, o QoSM irá acionar as funções de configuração do QoS requisitado e retornar o resultado para o terminal. Este retorno foi implementado através da modificação do estado de um objeto na MIB do QoSM o qual o EPQoSE deve consultar periodicamente.

Vimos que o modelo apresentado levou em consideração os seguintes pontos principais:

- Atualmente o modelo DiffServ é o mecanismo preferencial para prover QoS no centro (backbone) das redes IP.
- A necessidade de incluir entidades de controle de QoS como um sistema de gerenciamento baseado em Políticas.
- A necessidade de utilizar mecanismos de QoS informados através dos acordos pré-estabelecidos com o provedor de serviço ou administrador de rede e através de sinalização sob-demanda para o QoSM.

Este modelo possui uma série de vantagens e por isso acreditamos ser o caminho a ser seguido, para o controle de QoS em sistemas de colaboração visual baseado no H.323. Algumas das vantagens do modelo são:

- Assegura que o controle sobre o QoS permanece com entidades de rede e não com o terminal, o que permite que políticas organizacionais sejam respeitadas e seguidas.
- Possibilita que formas novas de cobrança do usuário sejam estabelecidas, cobrando o usuário pela quantidade de informações trafegadas e não apenas pela contratação do serviço com base em taxas fixas.
- É mais flexível, pois permite requisitar QoS de forma provisionada e/ou sinalizada.
- Oferece um mecanismo para assegurar QoS fim-a-fim entre os terminais H.323 em um mesmo domínio.

5.2 Limitações e trabalhos futuros

Embora o modelo apresentado no trabalho, possibilite o melhoramento uma série de questões relacionadas à requisição de QoS por sistemas H.323 ele ainda carrega consigo algumas limitações. Podemos listar três principais limitações do modelo:

- Não utiliza endereços de transporte para estabelecer as políticas de QoS.
- Introduce um certo atraso na fase de estabelecimento de chamada.
- Alcance intra-domínio.

O primeiro ponto é causado por que inserimos a fase de sinalização de QoS antes dos procedimentos definidos no H.323, para o estabelecimento de chamadas, local onde o endereço de transporte (porta) para estabelecimento dos canais de mídia são desconhecidos. Na verdade esta é uma discussão bem interessante por que, ao mesmo tempo em que não é adequado a existência de uma fase de negociação de QoS no meio dos procedimentos de abertura dos canais lógicos, pelos motivos explicados no texto, não se consegue estabelecer políticas de QoS mais específicas – baseados no endereço de transporte – quando a fase de negociação de QoS é realizada antes dos

procedimentos de estabelecimento de chamadas, o que impede a diferenciação dos níveis de QoS entre os diferentes tipos de mídia.

O segundo ponto, embora listado como uma limitação acontece de qualquer maneira, seja em uma fase anterior ao estabelecimento da chamada seja no meio dos procedimentos de estabelecimento da chamada, por que sempre que existirem procedimentos de sinalização de QoS eles irão tomar tempo dos dispositivos e da rede. Entretanto, acreditamos que quando introduzidos antes dos procedimentos de estabelecimento de chamadas, a qualidade percebida pelos usuários é maior do que quando estes procedimentos acontecem no meio da sinalização de chamadas.

O terceiro ponto, diz respeito ao alcance da solução. O modelo proposto neste texto não foi prevista a sinalização entre diferentes domínios. Para estas situações, recomenda-se que acordos bilaterais como o modelo de interconexão de tráfego utilizado atualmente pelas companhias de telefonia, seja utilizado.

Todas estas limitações podem ser resolvidas em trabalhos futuros para o melhoramento do modelo proposto. Para a resolução do primeiro ponto, podemos por exemplo, pensar em um modelo onde o QoSM esteja realmente implementado dentro de um gatekeeper e onde as comunicações aconteçam no modelo roteado pelo gatekeeper. Assim, no momento que o QoSM recebesse a mensagem *OpenLogicalChannelAck* do destino ele iniciaria a comunicação com o TRM para instalar as políticas de QoS nos roteadores do domínio.

Para reduzir o atraso na fase de negociação de QoS, a solicitação de QoS deve ocorrer como parte dos procedimentos de comunicação com o gatekeeper (H.225.0-RAS). Neste caso, o QoSM deve estar localizado funcionalmente dentro do gatekeeper. Este é o modelo que está sendo trabalhado em [BUC 2000], mas ainda longe de ser um consenso.

A questão da comunicação entre diferentes domínios, não é tão complexa de ser desenvolvida, mas apresenta algumas complicações a serem estudadas em detalhes, como: uma vez que o terminal sinalizou para o QoSM de seu domínio as suas necessidades de QoS e o QoSM verifica que o domínio possui recursos suficientes, ele já envia uma resposta OK para o terminal ou espera até que todos os domínios contactados respondam? Como fica a introdução de atraso, pois quanto mais domínios envolvidos maior o atraso envolvido? Questões como estas devem ser estudadas para estabelecer uma forma eficiente de comunicação inter-domínios.

5.3 Tabela Resumo

Neste sentido, esperamos que os argumentos apresentados bem como o modelo proposto tenha contribuído positivamente no sentido de encontrar uma maneira equilibrada para que sistemas H.323 possam usufruir mecanismos de QoS oferecidos pela rede para entregar seus sinais com qualidade aos usuários finais.

Para facilitar a visão do leitor entre os dois modelos (modelo atual H.323 e esta nova proposta), elaboramos a tabela abaixo, que resume de forma simples as características principais de ambos os modelos.

TABELA 5-1 – Tabela resumo de características

Características Principais	Solução	
	Recomendação H.323	Proposta Apresentada
Permite estabelecimento do canal H.225.0-RAS com QoS	Não	Não
Permite estabelecimento do canal H.225.0-Q.931 com QoS	Não	Sim
Permite estabelecimento do canal H.245 com QoS	Não	Sim
Permite estabelecimento dos canais de mídia com QoS	Sim	Sim
Utiliza QoS em nível de Aplicação	Sim (RTP/RTCP)	Sim (RTP/RTCP)
Utiliza QoS em nível de Rede	Sim	Sim
Mecanismo principal de QoS utilizado	IntServ	DiffServ, mas pode utilizar facilmente outros mecanismos.
Controle dos recursos de rede	Não	Sim
Modelo de requisição de QoS	Sinalizado	Sinalizado/Provisionado
Introdução de atraso nos procedimentos de sinalização da chamada	Sim	Sim
Verificação dos recurso de rede antes de proceder com a chamada	Não	Sim
Necessidade de entidades de controles adicionais	Não	Sim
Necessidade de implementação adicional nos terminais	Sim (Agente RSVP)	Sim (EPQoSE)
Políticas de QoS estabelecidas no nível de endereços de transporte	Sim	Não
Alcance da solução de QoS	Intradomínio/ Interdomínio	Intradomínio.

Conforme se pode visualizar, acima, mesmo que o modelo apresentado possua algumas limitações, acreditamos que as contribuições apresentadas superam estas limitações que podem ser resolvidas através do melhoramento do modelo conforme explicado brevemente.

Anexo

Proposta de Base de Informação de Gerenciamento (MIB)
para a interação entre o EPQoSE o QoSM

1. Introdução

Para tornar possível a sinalização de necessidades de QoS de uma aplicação H.323, via SNMP, é necessário que as entidades responsáveis por controlar e gerenciar as requisições de QoS estejam aptas a receber informações passadas por um terminal H.323.

Conforme definido em [BUC 2000] o QoSM é a entidade responsável por receber as requisições de QoS de um terminal H.323 e tratar adequadamente a requisição de QoS.

Este documento define a versão inicial para uma base de informação de gerenciamento (MIB), conforme especificado em [ROS 90], para utilização por entidades como o QoSM, responsáveis por receber e manipular requisições de QoS para sessões de colaboração visual.

2. Objetivos da MIB

Esta MIB foi definida para permitir que terminais H.323 que implementem um EPQoSE através do protocolo SNMP, possam se comunicar com entidades como o QoSM, para expressar suas necessidades de QoS. A partir das informações contidas nesta MIB, o QoSM poderá acionar outras entidades como o QoSEP e o TRM para realizar as reservas de recursos necessárias e prover qualidade de serviço à comunicação de colaboração visual.

Entretanto esta MIB não espera definir nenhum tipo de comunicação entre o QoSM e outras entidades de rede como QoSPE ou TRM.

3. Grupo de objetos

Como o único objetivo da MIB é proporcionar um meio de comunicação entre o terminal H.323 e o QoSM, foi definido apenas um grupo de objetos para esta MIB, o grupo de objetos requests, onde ficam armazenados todos os dados sobre as requisições de QoS.

Dentro deste grupo existem três tipos de informações armazenadas: informações de controle, informações sobre o descritor de QoS e informações sobre o descritor de tráfego. As informações de controle ficam armazenadas na tabela "qosControlTable", as informações sobre o descritor de QoS ficam armazenadas na tabela "qosDescriptorTable" e as informações sobre o descritor de tráfego ficam armazenadas na tabela "trafficDescriptorTable".

Todas as linhas correspondentes destas tabelas representam a informação completa de uma e apenas uma requisição de QoS. A correspondência entre as linhas são realizadas pelo ID de cada tabela.

4. Definições dos objetos

```

H323QoS-MIB DEFINITIONS ::= BEGIN
    IMPORTS
        RowStatus
            FROM SNMPv2-TC
        enterprises, MODULE-IDENTITY, OBJECT-TYPE, Integer32,
        Gauge32, IpAddress
            FROM SNMPv2-SMI;

    ufrgs MODULE-IDENTITY
        LAST-UPDATED          "200202231720Z"
        ORGANIZATION          "Organization"
        CONTACT-INFO          "Contact-info"
        DESCRIPTION           "Description"
        REVISION               "200202231720Z"
        DESCRIPTION           ""
        ::= { enterprises 12619 }

    org OBJECT IDENTIFIER
        ::= { iso 3 }

    dod OBJECT IDENTIFIER
        ::= { org 6 }

    internet OBJECT IDENTIFIER
        ::= { dod 1 }

    private OBJECT IDENTIFIER
        ::= { internet 4 }

    enterprises OBJECT IDENTIFIER
        ::= { private 1 }

    qosmMIB OBJECT IDENTIFIER
        ::= { ufrgs 1 }

```

4.1 A tabela qosControlTable

Em cada linha desta tabela ficam armazenadas as informações genéricas de controle, referente a uma requisição de QoS. Aqui ficam armazenados os endereços IP, bem como a máscara de sub-rede da origem e do destino.

A tabela possui um objeto para a identificação da requisição de QoS, que será relacionada com as linhas das outras tabelas. Este objeto é o objeto qosControlID.

Além disso, a tabela possui um campo para sinalização de liberação de recursos de rede relativos a uma requisição de QoS. Este objeto é o qosControlSetFree.

Para que o terminal H.323 possa acompanhar o estado de cada requisição, foi definido o objeto qosControlConfigStatus, responsável por armazenar o estado atual da requisição de QoS.

```

qosControlTable    OBJECT-TYPE
    SYNTAX          SEQUENCE OF QosControlEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION     "Description"
    ::= { requests 1 }

qosControlEntry    OBJECT-TYPE
    SYNTAX          QosControlEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION     "Row Description"
    INDEX          { qosControlIndex }
    ::= { qosControlTable 1 }

QosControlEntry ::= SEQUENCE {
    qosControlID      Integer32,
    qosControlTerminalSrcIpAddress  IpAddress,
    qosControlTerminalSrcMask       IpAddress,
    qosControlTerminalDestIpAddress IpAddress,
    qosControlTerminalDestMask      IpAddress,
    qosControlSetFree  INTEGER,
    qosControlRowStatus  RowStatus,
    qosControlIndex  Integer32,
    qosControlConfigStatus  INTEGER
}

qosControlID      OBJECT-TYPE
    SYNTAX          Integer32
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION     "valor de ID desta entrada na table. Deve
corresponder a um valor de na coluna trafficDescrID e qosDescrID."
    ::= { qosControlEntry 1 }

qosControlTerminalSrcIpAddress    OBJECT-TYPE
    SYNTAX          IpAddress
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION     "Endereço IP do terminal H.323 que está
solicitando suas necessidades de QoS da rede."
    ::= { qosControlEntry 2 }

qosControlTerminalSrcMask         OBJECT-TYPE
    SYNTAX          IpAddress
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION     "Máscara de sub-rede do terminal H.323
que está requisitando a sessão de QoS para o QoSM."
    ::= { qosControlEntry 3 }

qosControlTerminalDestIpAddress    OBJECT-TYPE
    SYNTAX          IpAddress
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION     "Endereço IP do terminal H.323 destino
para a sessão."

```



```

 ::= { qosControlEntry 4 }

qosControlTerminalDestMask OBJECT-TYPE
    SYNTAX                IPAddress
    MAX-ACCESS             read-create
    STATUS                  current
    DESCRIPTION             "Máscara de sub-rede do terminal H.323
destino para a sessão."
 ::= { qosControlEntry 5 }

qosControlSetFree OBJECT-TYPE
    SYNTAX                INTEGER { false ( 0 ) , true ( 1 ) }
}
    MAX-ACCESS             read-create
    STATUS                  current
    DESCRIPTION             "Quando possuir valor TRUE, este objeto
indica que os recursos alocados para esta sessão podem ser
desalocados.
Durante o andamento da sessão o valor do objeto deverá ser igual a
FALSE."
 ::= { qosControlEntry 6 }

qosControlRowStatus OBJECT-TYPE
    SYNTAX                RowStatus { active ( 1 ) ,
notInService ( 2 ) , notReady ( 3 ) , createAndGo ( 4 ) ,
createAndWait ( 5 ) , destroy ( 6 ) }
    MAX-ACCESS             read-create
    STATUS                  current
    DESCRIPTION             "Column Description"
 ::= { qosControlEntry 7 }

qosControlIndex OBJECT-TYPE
    SYNTAX                Integer32 ( -2147483648 ..
2147483647 )
    MAX-ACCESS             not-accessible
    STATUS                  current
    DESCRIPTION             "Column Description"
 ::= { qosControlEntry 8 }

qosControlConfigStatus OBJECT-TYPE
    SYNTAX                INTEGER { notStarted ( 0 ) ,
requested ( 1 ) , inProgress ( 2 ) , success ( 3 ) , failed ( 4 ) ,
setfree ( 5 ) }
    MAX-ACCESS             read-only
    STATUS                  current
    DESCRIPTION             "O Estado de processamento da requisição.
O valor inicial desta variável é zero (0), indicando que a requisição
ainda não foi realizada. Logo que, o terminal requisita os parâmetros
de QoS para o QoSM o estado deve ser requested (1), pois a
configuração dos dispositivos ainda não iniciou.
Assim que o QoSM iniciar a configuração dos dispositivos este estado
deve passar para inprogress (1), indicando que a configuração está
sendo realizada.
Se a configuração de todos os dispositivos tiver êxito, o estado deve
passar para success (3) e caso a configuração em pelo menos um
dispositivo falhar o estado deve ser failed (4).

```

Quando o objeto qosDescrSetFree for true (1), o estado deste objeto pode passar para setfree (5). Quando a linha estiver com este objeto no estado setfree (5), a respectiva linha poderá ser eliminada da tabela."

```
::= { qosControlEntry 9 }
```

4.2 A tabela qosDescriptorTable

Em cada linha desta tabela ficam armazenadas as informações que descrevem o nível de QoS requisitado pela aplicação. Aqui ficam armazenados as informações de requisição sobre o atraso, variação no atraso, perda de pacotes e largura de banda, requisitadas pela aplicação.

A tabela possui um objeto para a identificação da requisição de QoS, que será relacionada com as linhas das outras tabelas. Este objeto é o objeto qosDescrID, que deverá ser igual a uma entrada na tabela qosControlTable e na tabela trafficDescriptorTable.

```
qosDescriptorTable      OBJECT-TYPE
    SYNTAX               SEQUENCE OF QosDescriptorEntry
    MAX-ACCESS           not-accessible
    STATUS                current
    DESCRIPTION          "Tabela contendo informações sobre as
necessidades de QoS do Terminal H.323 que está se comunicando com o
QoSM."
    ::= { requests 2 }

qosDescriptorEntry      OBJECT-TYPE
    SYNTAX               QosDescriptorEntry
    MAX-ACCESS           not-accessible
    STATUS                current
    DESCRIPTION          "Informações sobre as necessidades de QoS de um
terminal H.323. Um objeto deste tipo é transiente, uma vez que ele
deixa de existir quando (ou tão logo) a requisição entra no estado
destroy (6)."
```

```
    INDEX                { qosDescrIndex }
    ::= { qosDescriptorTable 1 }
```

```
QosDescriptorEntry ::= SEQUENCE {
    qosDescrID Integer32,
    defaultMaxDelay INTEGER,
    maxDelay Integer32,
    defaultMaxDelayVariation INTEGER,
    maxDelayVariation Integer32,
    defaultPacketLoss INTEGER,
    packetLoss Integer32,
    maxBandwidth Gauge32,
    qosDescrRowStatus RowStatus,
    qosDescrIndex Integer32
}
```

```
qosDescrID OBJECT-TYPE
    SYNTAX               Integer32 ( -2147483648 ..
2147483647 )
    MAX-ACCESS           read-create
```

```

        STATUS                current
        DESCRIPTION            "valor de ID desta entrada na table. Deve
corresponder a um valor de na coluna trafficDescrID"
        ::= { qosDescriptorEntry 1 }

    defaultMaxDelay    OBJECT-TYPE
        SYNTAX            INTEGER { false ( 0 ) , true ( 1 ) }
    }

        MAX-ACCESS            read-create
        STATUS                current
        DESCRIPTION            "Este parâmetro deve receber um valor
verdadeiro, para especificar que o valor default de atraso máximo deve
ser utilizado. Este valor é especificado por um acordo estabelecido
anteriormente entre as entidades envolvidas.
Quando este valor é instanciado, o valor de MaxDelay deve ser
configurado como zero (0) e o valor de MaxBandwith pode ser
estabelecido."
        ::= { qosDescriptorEntry 2 }

    maxDelay            OBJECT-TYPE
        SYNTAX            Integer32 ( 0 .. 4294967295 )
        MAX-ACCESS            read-create
        STATUS                current
        DESCRIPTION            "Quando o valor de DefaultMaxDelay é
configurado em •falso• (0), o valor de MaxDelay deve ser usado para
especificar o valor numérico de atraso máximo."
        ::= { qosDescriptorEntry 3 }

    defaultMaxDelayVariation    OBJECT-TYPE
        SYNTAX            INTEGER { false ( 0 ) , true ( 1 ) }
    }

        MAX-ACCESS            read-create
        STATUS                current
        DESCRIPTION            "Este parâmetro deve receber um valor
verdadeiro, para especificar que o valor default de variação máxima de
atraso deve ser utilizado. Este valor é especificado por um acordo
estabelecido anteriormente entre as entidades envolvidas. Quando este
valor é instanciado, o valor de MaxDelayVariation deve ser configurado
como zero (0) e o valor de MaxBandwith pode ser configurado."
        ::= { qosDescriptorEntry 4 }

    maxDelayVariation    OBJECT-TYPE
        SYNTAX            Integer32 ( 0 .. 4294967295 )
        MAX-ACCESS            read-create
        STATUS                current
        DESCRIPTION            "Quando o valor de
DefaultMaxDelayVariation é configurado em •falso• (0), o valor de
MaxDelayVariation deve ser usado para especificar o valor numérico de
variação máxima do atraso."
        ::= { qosDescriptorEntry 5 }

    defaultPacketLoss    OBJECT-TYPE
        SYNTAX            INTEGER { false ( 0 ) , true ( 1 ) }
    }

        MAX-ACCESS            read-create
        STATUS                current

```

DESCRIPTION "Este parâmetro deve receber um valor verdadeiro (1), para especificar que o valor default de perda máxima de pacotes deve ser utilizado. Este valor é especificado por um acordo estabelecido anteriormente entre as entidades envolvidas. Quando este valor é instanciado, o valor de MaxPacketLoss deve ser configurado como zero (0) e o valor de MaxBandwith pode ser configurado."

```
::= { qosDescriptorEntry 6 }
```

packetLoss OBJECT-TYPE

```
SYNTAX Integer32 ( 0 .. 100 )
```

```
MAX-ACCESS read-create
```

```
STATUS current
```

DESCRIPTION "Quando o valor de DefaultPacketLoss é configurado em `false`, o valor de MaxPacketLoss deve ser usado para especificar o valor percentual relativo ao número máximo de pacotes descartados.

O QoS usa este valor para configurar nos dispositivos de rede, políticas para descarte de pacotes nas filas dos roteadores e switches."

```
::= { qosDescriptorEntry 7 }
```

maxBandwith OBJECT-TYPE

```
SYNTAX Gauge32 ( 0 .. 4294967295 )
```

```
MAX-ACCESS read-create
```

```
STATUS current
```

DESCRIPTION "Quando o desejo da aplicação é sobre a quantidade de largura de banda disponibilizada, este parâmetro deve ser usado para especificar o valor numérico de largura de banda requisitado.

A marcação deste parâmetro sobrepõe os parâmetros anteriores, isto é, os parâmetros anteriores deixarão de ser utilizados e por isso os valores de `defaultMaxDelay`, `defaultMaxDelayVariation` e `defaultPacketLoss`, devem estar configurados como `true` (1)."

```
::= { qosDescriptorEntry 8 }
```

qosDescrRowStatus OBJECT-TYPE

```
SYNTAX RowStatus { active ( 1 ) ,
```

```
notInService ( 2 ) , notReady ( 3 ) , createAndGo ( 4 ) ,
```

```
createAndWait ( 5 ) , destroy ( 6 ) }
```

```
MAX-ACCESS read-create
```

```
STATUS current
```

```
DESCRIPTION "Column Description"
```

```
::= { qosDescriptorEntry 9 }
```

qosDescrIndex OBJECT-TYPE

```
SYNTAX Integer32 ( -2147483648 .. 2147483647 )
```

```
MAX-ACCESS not-accessible
```

```
STATUS current
```

```
DESCRIPTION "Column Description"
```

```
::= { qosDescriptorEntry 11 }
```

4.3 A tabela trafficDescriptorTable

Em cada linha desta tabela ficam armazenadas as informações que descrevem o descritor de tráfego acordado entre a aplicação e a rede. Aqui ficam armazenadas as informações para ser utilizado para "Policing". São armazenadas informações sobre a taxa do algoritmo de tokenbucket utilizado, o tamanho do bucket a ser utilizado pelo algoritmo de token bucket a taxa máxima de pico a ser disparada pela aplicação e o tamanho máximo dos pacotes gerados pela aplicação.

A tabela possui um objeto para a identificação da requisição de QoS, que será relacionada com as linhas das outras tabelas. Este objeto é o objeto trafficDescrID, que deverá ser igual a uma entrada na tabela qosControlTable e na tabela qosDescriptorTable.

```

trafficDescriptorTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF TrafficDescriptorEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION     "Tabela contendo informações sobre o descritor
de tráfego das requisições de QoS feitas ao QoSM. Cada linha
especifica uma requisição de QoS, para um determinado terminal H.323."
    ::= { requests 3 }

```

```

trafficDescriptorEntry OBJECT-TYPE
    SYNTAX          TrafficDescriptorEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION     "Informações sobre o descritor de tráfego
relacionado a uma requisição de QoS. Um objeto deste tipo é
transiente, uma vez que ele deixa de existir quando (ou tão logo) a
requisição entra no estado destroy (6)."
```

```

    INDEX          { trafficDescrIndex }
    ::= { trafficDescriptorTable 1 }

```

```

TrafficDescriptorEntry ::= SEQUENCE {
    trafficDescrID Integer32,
    tokenBucketrate Integer32,
    tokenBucketSize Integer32,
    peakDataRate Gauge32,
    maxPacketSize Integer32,
    trafficDescrRowStatus RowStatus,
    trafficDescrIndex Integer32
}

```

```

trafficDescrID OBJECT-TYPE
    SYNTAX          Integer32
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION     "Valor de ID desta entrada na tabela.
Deve corresponder ao valo de uma coluna qosDescrID"
    ::= { trafficDescriptorEntry 1 }

```

```

tokenBucketrate OBJECT-TYPE

```

```

                SYNTAX                Integer32 ( -2147483648 ..
2147483647 )
                MAX-ACCESS            read-create
                STATUS                 current
                DESCRIPTION            "Este valor é estabelecido pela origem e
representa juntamente com o parâmetro TokenBucketSize a visão que o
terminal possui do tráfego gerado. Este parâmetro especifica a taxa em
bytes por segundo que o terminal se compromete a enviar dados para a
rede. Caso o terminal enviar dados para a rede, a uma taxa maior, a
rede não estará apta a garantir o nível de QoS requerido."
                ::= { trafficDescriptorEntry 2 }

                tokenBucketSize      OBJECT-TYPE
                SYNTAX                Integer32 ( -2147483648 ..
2147483647 )
                MAX-ACCESS            read-create
                STATUS                 current
                DESCRIPTION            "Este valor é estabelecido pela origem e
representa juntamente com o parâmetro TokenBucketRate a visão que o
terminal possui do tráfego gerado. Este parâmetro especifica em bytes
o tamanho, máximo a ser utilizado pelo algoritmo de token bucket, para
o tamanho do bucket."
                ::= { trafficDescriptorEntry 3 }

                peakDataRate          OBJECT-TYPE
                SYNTAX                Gauge32 ( 0 .. 4294967295 )
                MAX-ACCESS            read-create
                STATUS                 current
                DESCRIPTION            "É a taxa máxima que o terminal está
planejando enviar dados. Juntamente com o parâmetro MaximumPacketSize,
o tamanho máximo de uma rajada de dados que poderá ser enviada para a
rede."
                ::= { trafficDescriptorEntry 4 }

                maxPacketSize         OBJECT-TYPE
                SYNTAX                Integer32 ( 0 .. 65535 )
                MAX-ACCESS            read-create
                STATUS                 current
                DESCRIPTION            "Especifica o tamanho máximo dos
datagramas IP que devem ser enviados pela aplicação. A faixa de
Valores varia de 0 a 65535."
                ::= { trafficDescriptorEntry 5 }

                trafficDescrRowStatus OBJECT-TYPE
                SYNTAX                RowStatus { active ( 1 ) ,
notInService ( 2 ) , notReady ( 3 ) , createAndGo ( 4 ) ,
createAndWait ( 5 ) , destroy ( 6 ) }
                MAX-ACCESS            read-create
                STATUS                 current
                DESCRIPTION            "O estado atual da linha da tabela.

```

The different levels supported by Row Status are:

active (1) : The active status indicates that the conceptual row is available for use by the Managed device. A row can be added and deleted using this active status.

notInService (2) : The conceptual row that exists is Not in service as per this range.

notReady (3): This range indicates that the conceptual row exists but is not in an usable state by the managed device.

createAndGo (4) : As the name indicates, the conceptual row gets created on specifying this range from the management station and automatically becomes active, thus allowing to add any number of rows.

createAndWait (5): -Using this range will create a row from the Management station but would not be accessible by the managed device. The values can be set only if the row is made active.

destroy(6): The conceptual row can be removed from a table by specifying the respective instance and making the status as destroy."

```
 ::= { trafficDescriptorEntry 6 }
```

```

trafficDescrIndex OBJECT-TYPE
    SYNTAX          Integer32 ( -2147483648 ..
2147483647 )
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION     "Column Description"
    ::= { trafficDescriptorEntry 8 }

requests    OBJECT IDENTIFIER
    ::= { qosmMIB 4 }
```

END

Bibliografia

- [ADV 2002] ADVENTNET INC. **AdventNet SNMP API 4.0**. 2002. Disponível em: <<http://www.adventnet.com/products/snmpapi4/index.html>>. Acesso em: 08 out. 2002.
- [BHA 2000] BHAT, R.; GUPTA, R. Jain Protocol APIS. **IEEE Communications Maganize**, New York, v. 38, n. 1, Jan. 2000.
- [BLA 98] BLAKE, S. et al. **An Architecture for Differentiated Services**: RFC 2475. [S.l.]: IETF, 1998.
- [BRA 97] BRADEN, R. et al. **Resource Reservation Protocol (RSVP) – version 1 Functional Specification**: RFC 2205. [S.l.]: IETF, 1997.
- [BRA 94] BRADEN, R. et al. **Integrated Services in the Internet Architecture: an Overview**: RFC 1633. [S.l.]: IETF, 1994.
- [BSI 99] BRITISH STANDARD INSTITUTE. **Information Security Management – Part 2: Specification for information security management systems**: BS 7799-2:1999. [S.l.], 1999.
- [BUC 2000] BUCKLEY, Mike. **H.323 Annex N – End-to-End QOS control and Signaling in H.323 Systems**: TD 66. [S.l.]: ITU-T, 2000.
- [CAS 99] CASTELLS, Manuel. **A era da Informação: Economia, Sociedade e Cultura**. São Paulo: Paz e Terra, 1999. v.1.
- [CIS 2001] CISCO SYSTEMS. **Designing Service Provider Core Networks to Deliver Real-Time Services**. [S.l.]: 2001.
- [CIS 97] CISCO SYSTEMS. **Quality of Service (QoS) Networking**. [S.l.]: 1997.
- [COM 95] COMMER, D. E. **Internetworking with TCP/IP: Principles, Protocols and Architetures**. New Jersey: Prentice Hall, 1995.
- [DAV 99] DAVIS, A. W. **Integrated Collaboration: Driving Business Efficiency to the Next Millenium**. 1999. Disponível em: <<http://www.picturetel.com/pdf/intw01.pdf>> Acesso em: 02 maio 2001.
- [DUP 2001] DUPONCHEL, Y. **J.323 Engine**. 2001. Disponível em: <<http://www.alphaworks.ibm.com/tech/j323engine>> Acesso em: 08 jan. 2002.
- [EUR 2002a] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. **End-to-end Quality of Service in TIPHON systems: general aspects of Quality of Service (QoS)**: TR 101 329-1 V3.1.2. [S.l.], 2002.

- [EUR 2002b] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. **End-to-end Quality of Service in TIPHON systems**: definition of speech quality of service (QoS) classes: TS 101 329-2 V2.1.3. [S.l.], 2002.
- [EUR 2002c] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. **End-to-end Quality of Service in TIPHON systems**: signalling and control of end-to-end quality of service (QoS). TS 101 329-3 V2.1.2. [S.l.], 2002.
- [EUR 2002d] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. **End-to-end Quality of Service in TIPHON systems**: quality of service (QoS) measurement methodologies. TS 101 329-5 V1.1.2. [S.l.], 2002.
- [EUR 2002e] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. TR 101 329-6 V2.1.1. **End-to-end Quality of Service in TIPHON systems**: actual measurements of network and terminal characteristics and performance parameters in TIPHON networks and their influence on voice quality. TR 101 329-6 V2.1.1. [S.l.], 2002.
- [FIN 2002] FINEBERG, Victoria. A Practical Architecture for Implementing End-to-End QoS in an IP Network. **IEEE Communications Magazine**, [S.l.], v. 40, n. 1, p. 122 - 130, Jan. 2002.
- [FVC 2001] FIRST VIRTUAL COMMUNICATIONS. **Implementing IP-based Videoconferencing through a Firewall**. 2001. Disponível em: <http://www.fvc.com/eng/webconferencing/whitepaper_desc.htm>. Acesso em: 08 nov. 2001.
- [GLI 2000] GLITHO, R. H. Advanced Services Architectures for Internet Telephony: A Critical Overview. **IEEE Network**, Los Alamitos, v. 14, n. 4, p. 38-44, Aug. 2000.
- [GRA 2001] GRANVILLE, L.Z.; CECCON, M.B.; TAROUCO, L.M.; ALMEIDA, M. J.; CARISSIMI, A.S. An Approach for Integrated Management of Networks with Quality of Service Support Using QAME. In: INTERNATIONAL WORKSHOP ON DISTRIBUTED SYSTEMS: OPERATIONS & MANAGEMENT, 12., 2001, Nancy. **Proceedings...**[S.l.]: INRIA, 2001.
- [HAS 2000] HASSAN, M.; NAYANDORO, A.; ATIQUZZAMAN. Internet Telephony: Services, Technical Challenges, and Products. **IEEE Communications Magazine**, New York, v. 38, n. 4, p. 96-103, Apr. 2000.
- [HER 2000] HERSENT, Oliver; GURLE, David; PETIT, Jean-Pierre. **IP Telephony**: Packet based multimedia communications systems. Great Britain: Addison Wesley, 2000.
- [ITU 98] ITU-T. **H.323**: Audiovisual and Multimedia Systems. Infrastructure of audiovisual services – Systems and terminal equipment for audiovisual services. [S.l.]: 1998.

- [ITU 96] ITU-T. **H.225.0**: Transmission of non-telephone signals – Call signalling protocols and media fluxo packetization for packet based multimedia communication systems. [S.l.]: 1996.
- [ITU 97a] ITU-T. **H.225.0**: Audiovisual and multimedia systems – Infrastructure of audiovisual services – Communication procedures. [S.l.]: 1997.
- [ITU 97b] ITU-T. **T.120**: Terminal equipments and protocols for telematic services – Data protocols for multimidia conferencing. [S.l.]: 1997.
- [KAW 2000] KAWAMURA, R.; STADLER, R. Active Distributed Management for IP Networks. **IEEE Communications Magazine**, New York, v. 38, n. 4, p. 114-120, Apr. 2000.
- [KUM 2000] KUMAR, V.; KORPI, M.; SENGODAN, Senthil. **IP Telephony with H.323**: architectures for unified networks and integrated services. New York, NY: John Wiley & Sons, 2000.
- [LAL 97] LALIBERTE, D. **WWW Collaboration Projects**. Feb. 1997. Disponível em: <<http://www.hypernews.org/HyperNews/get/www/collaboration.html>>. Acesso em: 05 ago. 2001.
- [LAW 99] LAWRENCE, J. On the Road to Voice over IP. **Business Communications Review**, [S.l.]: p. 24, 1999.
- [LI 2000] LI, B.; HAMDI, M.; JIANG, D.; CAO, X.; HOU, Y. T. QoS-Enabled Voice Support in the Next-Generation Internet: Issues, Existing Approaches and Challenges. **IEEE Communications Magazine**, New York, v. 38, n. 4, p. 54-61, 2000.
- [LIU 2000] LIU, H.; MOUCHTARIS, P. Voice over IP signalling: H.323 and Beyond. **IEEE Communications Magazine**, New York, v. 38, n. 10, p. 142-148, 2000.
- [MAH 2000] MAHON, H.; BERNET, Y.; HERZOG, S. Requirements for a Policy Management System. [S.l.]: IETF, 2000.
- [MAS 2002] MASE, K.; TOYAMA, Y. End-to-End Measurement Based Admission Control. In: INTERNATIONAL CONFERENCE ON COMMUNICATIONS, 2002, New York. **Proceedings...** [S.l.]: INRIA, 2002.
- [MCD 94] MCDYSAN, D. E.; SPOHN, D. L. **ATM: theory and application**. New York: McGraw-Hill, 1994.
- [MCD 98] MCDYSAN, D. E.; SPOHN, D. L. **Hands-On ATM**. New York : McGraw-Hill, 1998.
- [MIC 2002] MICROSOFT CORPORATION. **How to Establish NetMeeting Connections Through a Firewall**. Oct. 2002. Disponível em:

<<http://support.microsoft.com/default.aspx?scid=KB;en-us;158623&>>.
Acesso em: 05 nov. 2002.

- [MIN 97] MINOLI, D.; GOLWAY, T.; SMITH, Norris. **Planning and Managing ATM Networks**. Greenwich: Manning Publications Company, 1997.
- [NGU 2001] NGUYEN, T.; YEGENOGLU, F.; SCIUTO, A.; SUBBARAYAN, Ravi. Voice over IP Service and Performance in Sattelite Networks. **IEEE Communications Magazine**, New York, v. 39, n. 3, p. 164-171, Mar. 2001.
- [NIC 99] NICHOLS, K.; JACOBSON, V.; ZHANG L. **A Two-bit Differentiated Architecture for the Internet**. [S.l.]: IETF, 1999.
- [NIC 98] NICHOLS, K.; BLAKE, S.; BAKER, F.; BLACK, D. **Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers: RFC 2474**. [S.l.]: IETF 1998.
- [ONE 2002] O'NEIL, T. M. **Quality of Experience and Quality of Service for IP Video Conferencing**. Disponível em: <http://www.polycom.com/common/pw_item_show_doc/0,1449,978,00.pdf> . Acesso em: 20 out. 2002.
- [ONV 95] ONVURAL, R. O. **Asynchronous Transfer Mode Networks: Performance Issues**. Boston: Artech House, 1995.
- [OPE 2001] OPERAX COMPANY. **IQ-Man™ in IP telephony systems**. [S. l.]: 2001.
- [PAR 2001] PARK, J.; BAEK, J.; HONG, J. W. Management of Service Level Agreements for Multimedia Internet Service Using a Utility Model. **IEEE Communications Magazine**, New York, v. 39, n. 5, p. 100-106, May 2001.
- [PLA 95] PLAGEMANN, Thomas; SÆTHRE, K. A.; GOEBEL, V. Application Requirements and QoS Negotiation in Multimedia Systems. In: WORKSHOP ON PROTOCOLS FOR MULTIMEDIA SYSTEMS, 2., 1995, Salzburg, Austria, **Proceedings...** [S. l.: s. n.], 1995.
- [PRO 2002] PROJETO METROPOA. Disponível em: <<http://www.metroboa.tcbe.br>>. Acesso em: out. 2002.
- [RAJ 99] RAJAHAME, J.; SENGODAN, S. **On QoS for H.323 Systems: Q.12-14/16 Rapporteur Meeting**. Document – APC – 1705. [S.l.]: ITU-T, 1999.
- [RAY 2000] RAYES, A.; SAGE, K. Integrated Management Architecture for IP-Based Networks. **IEEE Communications Magazine**, New York, v. 38, n.4, p. 48-53, Apr. 2000.
- [RIS 2001] RISSO, F. Decoupling bandwidth and delay properties in class based queuing. In: IEEE SYMPOSIUM ON COMPUTERS AND COMMUNICATIONS, 6., 2001, Hammamet-Tunisia, 2001. **Proceedings...** [S. l.: s. n.], 2001.

- [ROS 90] ROSE, M.; MCCLOGHRIE, K. **Structure and Identification of Management Information for TCP/IP-based Internets**: RFC 1155. [S.l.]: IETF 1990.
- [SBC 95] SBC PACIFIC BELL. **Pacific Bell Videoconferencing for learning**: introduction. 1995. Disponível em: <<http://www.kn.pacbell.com/wired/vidconf/intro.html>>. Acesso em: 20 ago. 2001.
- [SCH 99] SCHAPHORST, Richard. **Videoconferencing and Videotelephony technology and standards**. 2nd ed. Norwood: Artech House, 1999.
- [SCH 96a] SCHULZRINNE, H.; CASNER, S.; FREDERICK, R.; JACOBSON V. **RTP: A Transport Protocol for Real-Time Applications**: RFC 1889. [S. l.]: IETF 1996.
- [SCH 96b] SCHULZRINNE, H. **RTP Profile for Audio and Video Conferences with Minimal Control**: RFC 1890. [S. l.]: IETF 1996.
- [STA 95a] STALLINGS, W. **Handbook of Computer-Communications Standards**: The Open System Interconnection (OSI) Model and OSI related Standards. [S. l.]: Mcmillian Publishing Company, 1995. v.1.
- [STA 95b] STALLINGS, W. **ISDN and Broadband ISDN with Frame Relay and ATM**. 3rd ed. Upper Saddle River: Prentice Hall, 1995.
- [STA 97] STALLINGS, W. **Local e Metropolitan Area Networks**. 5th ed. Upper Saddle River: Prentice Hall, 1997.
- [STA 2001] STADER, R. **QoS provisioning for IP Telephony Networks by Advanced Bandwith Management**. A Thesis submitted to the University of Applied Science at Offenburg in partial fulfilment of de Requirements for the Degree. Feb. 2001.
- [SUN 2002] SUN MICROSYSTEMS. **The Jain API**: Overview. Disponível em: <<http://java.sun.com/products/jain/overview.html>>. Acesso em: 18 jul. 2002.
- [SVE 98] SVEYBI, K. E. **A Nova Riqueza das Organizações**: patrimônio de conhecimento. Campinas, SP: Campus, 1998.
- [TAN 96] TANENBAUM, A. S. **Computer Netwoks**. 3rd ed. New Jersey: Prentice Hall, 1996.
- [TAR 2002] TAROUÇO, L.; TAMUSIUNAS, F. QoS requirements for data collaboration using T.120. In: INTERNET2 FALL 2002 MEMBER MEETING, Los Angeles. Disponível em: <<http://www.internet2.edu/presentations/fall02/20021028-ITF-Tarouco.htm>>. Acesso em 03 dez. 2002.

- [TEI 2000] TEITELBAUM, B.; CHIMENTO, P. **QBone Bandwidth Broker Architecture**. June 2000. Disponível em: <<http://qbone.internet2.edu/bb/bboutline2.html>>. Acesso em: 02 out. 2002.
- [TOB 99] TOBIET, H.; LORENZ, P. Desktop Videoconferencing Performance Based on na ATM Metropolitan Area Network. **IEEE Communications Magazine**, New York, v. 37, n. 12, p. 112-116, Dec. 1999.
- [WYD 2002] WYDROWSKY, B.; ZUKERMAN, M. QoS in Best-Effort Networks. **IEEE Communications Magazine**, New York, v. 40, n. 12, p. 44-49. Dec. 2002.