UNIVERSIDADE SALVADOR – UNIFACS
PROGRAMA DE PÓS-GRADUAÇÃO EM REDES DE COMPUTADORES
MESTRADO PROFISSIONAL EM REDES DE COMPUTADORES

ANA LÚCIA LAGE PEREIRA

UMA PROPOSTA DE ARQUITETURA DE SUPORTE PARA A QUALIDADE DE SERVIÇO ORIENTADA PARA APLICAÇÕES DE TELEMEDICINA

Salvador
2008

ANA LÚCIA LAGE PEREIRA

UMA PROPOSTA DE ARQUITETURA DE SUPORTE PARA A QUALIDADE DE SERVIÇO ORIENTADA PARA APLICAÇÕES DE TELEMEDICINA

Dissertação apresentada ao Curso de Mestrado Profissional em Redes de Computadores, área de concentração em Redes de Computadores, Universidade Salvador – UNIFACS, como requisito parcial para obtenção do título de Mestre.

Orientador: Prof. Dr. Joberto Sérgio Barbosa Martins

Salvador
2008

FICHA CATALOGRÁFICA
(Elaborada pelo Sistema de Bibliotecas da Universidade Salvador - UNIFACS)

Pereira, Ana Lucia Lage
Uma proposta de arquitetura de suporte para a qualidade de serviço orientada para a aplicação de telemedicina/Ana Lucia Lage Pereira. - 2009.
111f.

Dissertação (Mestrado) - Universidade Salvador – UNIFACS. Mestrado em Sistemas de Computação, 2009.
Orientador: Prof. Joberto Sérgio Barbosa Martins

1.Redes IP. 2. Telemedicina 3. Gerência de redes I. Martins, Joberto Sérgio Barbosa, orient. II. Título.

CDD: 004.6068

TERMO DE APROVAÇÃO

ANA LÚCIA LAGE PEREIRA

UMA PROPOSTA DE ARQUITETURA DE SUPORTE PARA A QUALIDADE DE SERVIÇO ORIENTADA PARA APLICAÇÕES DE TELEMEDICINA

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre em Redes de Computadores, Universidade Salvador – UNIFACS, pela seguinte banca examinadora:

Prof. Dr. José Neuman de Souza _____
Docteur en Informatique, Université Pierre et Marie Curie, 1994, França; M.Sc. em Ciência da Computação, UFPB, 1989; B.Sc. em Eng. Civil, UFC, 1984.
Universidade Federal do Ceará – UFC.

Prof. Dr. Thomas Araújo Buck _____
Dr. rer. nat. em Informática, Universität Tübingen, 1995, Alemanha; M.Sc. em Eng. de Computação, UNICAMP, 1989; B.Sc. em Eng. Mecânica, UFBa, 1986.
Universidade Salvador – UNIFACS.

Prof. Dr. Joberto Sérgio Barbosa Martins – Orientador _____
Docteur en Informatique, Université Pierre et Marie Curie, 1986, França; M.Sc. em Eng. Eletrônica, Netherlands University Foundation, Holanda, 1979; B.Sc. em Eng. Eletrônica, UFPB, 1977.
Universidade Salvador – UNIFACS.

Salvador, 16 de dezembro de 2008

A Roberto, meu companheiro, meu maior incentivador.
A Laszlo, meu filho, minha maior alegria.

AGRADECIMENTOS

À minha família, por todo o apoio aos meus projetos e sonhos, e por estar sempre comigo, nos momentos felizes e nos momentos difíceis. Inclusive em enfrentar o problema de saúde que impactou o prazo de defesa dessa dissertação. Eu não conseguiria sem vocês.

Ao meu orientador Prof. Joberto Martins, por sua visão, facilitação, encaminhamento e encorajamento. Obrigado por me fazer acreditar no valor do que eu havia produzido e me incentivar a defender esse trabalho.

Ao Prof. Carlos Ferraz, por me acolher no projeto InfraVIDA e apoiar a publicação de artigos e a participação em congressos internacionais (WebMedia/LA-Web 2004 em Ribeirão Preto e IEEE IPOM 2004 em Pequim).

Aos meus professores e colegas no NUPERC (Núcleo de Pesquisa em Redes e Computação da UNIFACS), pela troca de idéias e de conhecimentos e pela convivência enriquecedora. Em especial a Jorge Oliveira, co-autor de dois artigos, com quem troquei muitas idéias e virei muitas horas e muito café em laboratório.

RESUMO

Essa dissertação propõe e avalia uma arquitetura de suporte de Qualidade de Serviço para aplicações de Telemedicina, com suporte ao uso concorrente de Vídeo sob Demanda e Videoconferência sobre uma infra-estrutura de rede IP. A arquitetura de Qualidade de Serviço (QoS - *Quality of Service*) considerada é o *DiffServ* (*Differentiated Services*) proposta pelo IETF (*Internet Engineering Task Force*). Em particular, este estudo propõe a utilização de um elemento gerenciador de políticas de QoS – o *Bandwidth Broker* – que recebe solicitações de alocação dinâmica de recursos das aplicações de Vídeo sob Demanda e Videoconferência, valida as solicitações contra contratos de níveis de serviço (SLAs - *Service Level Agreements*) acordados, e aloca recursos através de distribuição de configuração de mecanismos *DiffServ* em dispositivos de roteamento na rede. Esse trabalho especifica uma solução de integração da rede de vídeo digital a um elemento gerenciador de políticas ao nível de plano de controle – o *Bandwidth Broker*, de forma a permitir o gerenciamento dinâmico de recursos baseado em políticas. Especifica e prototipa, ainda, uma infra-estrutura de suporte para configurações, fazendo uso de plataformas Linux, e valida resultados de implementação de mecanismos de Qualidade de Serviço *DiffServ*. Para este fim são considerados diferentes cenários que simulam o uso das aplicações do Projeto InfraVIDA – um projeto de telemedicina que integra serviços de Telediagnóstico e Segunda Opinião Médica, baseados na Internet. Uma avaliação das tendências de evolução do campo de gerenciamento de Qualidade de Serviço em redes IP baseado em políticas complementa esse estudo.

Palavras-chave: Redes IP. Telemedicina. Aplicações Multimídia. Qualidade de Serviço. DiffServ. Gerenciamento Baseado em Políticas.

ABSTRACT

This master's thesis proposes and evaluates a Quality of Service framework for telemedicine environments, supporting concurrent applications such as Video on Demand and Videoconference over an IP network infrastructure. IETF Differentiated Services (*DiffServ)* architecture for IP Quality of Service (QoS) is considered. More specifically, this study proposes the utilization of a Bandwidth Broker – a network element responsible for IP QoS policy management, which receives dynamic allocation resources requests from Video on Demand and Videoconference applications, validates them against negotiated Service Level Agreements (SLAs) and allocates resources through configuration distribution of *DiffServ* mechanisms for routing network devices. This study specifies an integration solution at the control plane, between the digital video network and a policy management element – the Bandwidth Broker, to make the dynamic allocation resources policy-based management possible. It also specifies and prototypes a Linux infrastructure for configuration support and validates results of *DiffServ* Quality of Service mechanisms implementation. For this purpose, different scenarios are considered simulating the concurrence of applications used in the scope of InfraVIDA – a telemedicine project which integrates remote diagnosis and second medical opinion applications as well as a remote continuous education system for Health professionals, all based on the Internet. An evaluation of evolving tendencies in the field of IP Quality of Service Policy Based Management complements this study.

Keywords: IP networks. Telemedicine. Multimedia Applications. Quality of Service. DiffServ. Policy Management

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE TABELAS

LISTA DE SIGLAS

ABRA – Ambiente Brasileiro de Aprendizagem

ADPCM – *Adaptive Differential Pulse Code Modulation*

ASP – *Advanced Simple Profile*

AVC – *Advanced Video Coding*

BA – *Behavior Aggregate*

BB – *Bandwidth Broker*

BP– *baseline profile*

CAT – *Client-Accept*

CC – *Client-Close*

CBQ – *Class-Based Queueing*

CBWFQ – *Class-Based Weighted Fair Queueing*

CD-ROM – *Compact Disc Read-Only Memory*

CIR – *Commited Information Rate*

CE – *Congestion Experienced*

CNPq – Conselho Nacional do Desenvolvimento Científico e Tecnológico

COPS – *Common Open Policy Protocol*

COPS-PR – *Common Open Policy Protocol Provisioning*

CoS – *Class of Service*

CRUDE – *Collector for Real-time UDP Data Emitter*

CS-ACELP – *Conjugate Structure, Algebraic Code Excited Linear Prediction*

DCT – *Discrete Cosine Tranform*

DEC – *Decision*

DICOM – *Digital Imaging and Communications in Medicine*

DiffServ – Differentiated Services

DPCM – *Differential Pulse Code Modulation*

DRR – *Deficit Round Robin*

DSCP – *Differentiated Services Code Point*

DVD – *Digital Video Disc*

DVX – *Digital Video Express*

DynaVideo – *Dynamic Video Distribution Service*

EAD – Ensino à Distância

ECN – *Explicit Congestion Notification*

ECT – *ECN-Capable Transport*

FIFO – *First-In, First-Out*

FQ – *Fair Queueing*

G.114– Recomendação ITU-T de Atraso de Transmissão Fim-a-Fim

G.711, G.722, G.723, G.728, G.729 – Recomendações ITU-T de Codificação de Áudio

GK – *Gatekeeper*

GPS – *Global Position Sattelite*

GPS –*Generalized Processor Sharing*

GTVD-RNP – Grupo de Trabalho de Vídeo Digital da Rede Nacional de Pesquisa

H.323 – Recomendação ITU-T de Sistemas de Comunicação Multimídia em Redes de Pacotes

H.261, H.262, H.263, H.264 – Recomendações ITU-T de Codificação de Vídeo

HD DVD – *High-Definition Digital Video Disc*

HDTV – *High-Definition Television*

HiP – *high profile*

HTB – *Hierachical Token Bucket*

IEC – *International Eletrotecnical Commission*

IEEE – *Institute of Electrical and Electronic Engineers*

IEEE IPOM– *IEEE Workshop on IP Operations and Management*

IETF – *Internet Engineering Task Force*

InfraVIDA — Infra-Estrutura de Vídeo Digital para Aplicações de Tele-Saúde

IntServ – Integrated Services

IoD – *Image on Demand*

IP – *Internet Protocol*

ISDN – *Integrated Services Digital Network*

ISO – *International Organization for Standardization*

ITU-T – *International Telecommunication Union - Telecommunication Standardization Sector*

JAX-RPC – *Java APIs for XML-Based Remote Procedure Call*

JDBC – *JDBC Java Database Connectivity*

JWSD – *Java Web Services Developer Pack*

JPEG – *Joint Photographic Experts Group*

LD-CELP – *Low-Delay, Code Excited Linear Prediction*

LDAP – *Light-Weight Directory Access Protocol*

LE – *Lower Effort*

LLQ – *Low Latency Queueing*

MAC – *Media Access Control*

MCU – *Multipoint Control Unit*

MIB – *Management Information Base*

MOS – *Mean Opinion Score*

MP – *main profile*

MP3 – *MPEG-1 Audio Layer 3*

MPEG – *Motion Picture Experts Group*

MP-MLQ – *Multi-Pulse Maximum Likelihood Quantisa*

NTP – *Network Time Protocol*

PBM – *Policy Based Management*

PCM – *Pulse Code Modulation*

PDB – *Per-Domain Behavior*

PDP – *Policy Decision Point*

PEP – *Policy Enforcement Point*

PHB – *Per-Hop Behavior*

PIB – *Policy Information Base*

PIR – *Peak Information Rate*

PQ – *Priority Queueing*

PRC – *Provisioning Classe*

PRI – *Provisioning Instance*

PRID – *Provisioning Instance Identifier*

QoS – *Quality of Service*

RAA – *Resource Allocation Answer*

RADIUS – *Remote Authentication Dial In User Service*

RAP WG – *Resource Allocation Protocol Workgroup*

RAR – *Resource Allocation Request*

RDBMS – *Relational DataBase Management System*

RED – *Random Early Detection*

REQ – *Request*

RHP –Real Hospital Português

RNP – Rede Nacional de Pesquisa

RPC – *Remote Control Procedure*

RPT – *Report State*

RSVP – *Resource Reservation Protocol*

RTCP – *Real-Time Control Protocol*

RTP – *Real-Time Transport Protocol*

RTT – *Round Trip Time*

RUDE – *Real-time UDP Data Emitter*

SIBBS – *Simple Inter-Domain Bandwidth Broker Protocol*

SLA – *Service Level Agreement*

SLS – *Service Level Specification*

SMI – *Structure of Management Information*

SNMP – *Simple Network Management Protocol*

SOA – *Service Oriented Architecture*

SOAP – *Simple Object Access Protocol*

SQL – *Structured Query Language*

TC – *Traffic Class*

TCB – *Traffic Conditioning Behavior*

TCP - *Transport Control Protocol*

TCS – *Traffic Conditioning Specification*

ToS – *Type of Service*

UDP – *User Datagram Protocol*

UFBA – Universidade Federal da Bahia

UFPE – Universidade Federal de Pernambuco

UFPB – Universidade Federal da Paraíba

UFRN – Universidade Federal do Rio Grande do Norte

UNIFACS – Universidade Salvador

VCD – *Video Compact Disc*

VHS – *Video Home System*

VoD – *Video on Demand*

VoIP – *Voice over IP*

W3C – *World Wide Web Consortium*

WFQ – *Weighted Fair Queueing*

WRR – *Weighted Round Robin*

XML – *eXtensible Markup Language*

XP – *eXtended Profile*

SUMÁRIO

CAPÍTULO 1 INTRODUÇÃO

1.1 CONTEXTO

A Internet evolui em direção a uma rede multiserviços, permitindo o suporte a aplicações nos mais diversos segmentos e áreas de atividades, incluindo a área médica. A disponibilização do acesso à Internet, em escala cada vez maior no Brasil, levou o Ministério da Saúde a aprovar um projeto de pesquisa, que estudava meios de levar às localidades distantes e de mais baixo nível de recursos, a possibilidade de acesso remoto a centros médicos de primeira linha, que pudessem apoiar o diagnóstico e a orientação do tratamento de pacientes à distância.

Assim surgiu o Projeto InfraVIDA - Infra-Estrutura de Vídeo Digital para Aplicações de Telemedicina (FERRAZ, 2001), com a premissa de uso de infra-estrutura de rede de vídeo digital sobre redes IP.

No InfraVIDA, uma das aplicações é a Segunda Opinião Médica, que consiste em uma sessão de Videoconferência pré-agendada, onde um médico especialista (consultor), o solicitante de uma Segunda Opinião, e possivelmente outros colaboradores convidados, analisam juntos o prontuário de um paciente e seus exames, inclusive imagens radiológicas e vídeos de procedimentos cirúrgicos ou exames de ultra-som, à distância em tempo real, com o objetivo de diagnóstico médico. As sessões de Videoconferência e Imagem e Vídeo sob Demanda podem ocorrer simultaneamente, concorrendo por recursos da rede entre si e com outras aplicações.

As aplicações multimídia, colaborativas e de acesso remoto a bancos de dados, que compõem o projeto, têm diferentes exigências em termos de Qualidade de Serviço (QoS – *Quality of Service*), que se traduzem em especificação de parâmetros de rede mensuráveis, basicamente requerimentos mínimos de banda, atraso, variação de atraso (*jitter*) e taxa de perda de pacotes.

As redes IP originalmente tratam o tráfego indiscriminadamente em função da disponibilidade dos recursos da rede (comportamento conhecido como melhor esforço ou *best effort*) e não oferecem meios de diferenciação e priorização de tráfego. Um grande número de pesquisas foi efetuado desde os anos 90 para introduzir Qualidade de Serviço em redes IP. O *Internet Engineering Task Force* (IETF) definiu duas arquiteturas para endereçar o problema: - *Integrated Services* (*IntServ*) e *Differentiated Services* (*DiffServ*).

No modelo *IntServ* as aplicações individuais ativas nos hospedeiros (*hosts*) sinalizam seus requisitos de Qualidade de Serviço para a rede, que encaminha a sinalização até o destino, de modo a reservar recursos para atender os fluxos individuais das aplicações fim-a-fim.

No modelo *DiffServ*, os elementos de rede são configurados para tratar um número finito de classes de tráfego distintas e bem definidas, que correspondem aos agregados de fluxos de tráfego de diversas aplicações que tenham requisitos de Qualidade de Serviço semelhantes.

O modelo *IntServ* provê uma solução de Qualidade de Serviço fim-a-fim com maior granularidade por meio de sinalização *Resource Reservation Protocol* (RSVP) fim-a-fim, controle de admissão e manutenção de estado para cada fluxo/reserva RSVP em cada elemento de rede.

O modelo *DiffServ* usa a abordagem mais simples de categorizar o tráfego em diferentes classes *Class of Service* (CoS) com parâmetros de QoS comuns e aplicar os mesmos mecanismos de QoS a todos os fluxos de uma mesma classe de serviço. Por não tratar cada fluxo separadamente, no *DiffServ* a sinalização é eliminada e o número de estados a ser mantido em cada elemento de rede é drasticamente reduzido, resultando numa solução de Qualidade de Serviço de menor granularidade, mas em contra-partida, mais eficiente e mais escalável.

Para que um tráfego IP possa obter uma determinada Qualidade de Serviço, os recursos devem ser alocados de uma maneira apropriada sob o controle de um administrador. A necessidade de otimização no uso de recursos das redes e a perspectiva de disseminação de serviços com conseqüente exigência de escalabilidade requerem um nível de flexibilidade da arquitetura de Qualidade de Serviço, que permita que os recursos de rede sejam negociados e alocados dinamicamente, inclusive com a configuração dinâmica dos elementos de rede para aceitar novos perfis de tráfego.

Essa é a razão da proposta de uma arquitetura de gerenciamento de recursos baseada em políticas. Esta arquitetura permite ao administrador criar políticas de alocação de recursos e de aplicá-las manual ou automaticamente em uma rede, fazendo uso, por exemplo, de um elemento servidor de políticas de QoS – também chamado *Bandwidth Broker* (BB) – que recebe dinamicamente requisições de alocação de recursos dos fluxos entrantes e, tendo uma visão completa da disponibilidade e alocação dos recursos da rede, pode autorizar e aceitar o fluxo após validação do contrato de serviço *Service Level Agreement* (SLA) com o cliente.

1.2 MOTIVAÇÕES E CONTRIBUIÇÕES ESPERADAS

As principais motivações dessa dissertação são:

- A disponibilidade de acesso à Internet e a redes IP oferece um meio universal e de baixo custo para viabilização de sistemas de telemedicina. No entanto, as aplicações que compõem tais sistemas – de acesso a bases de dados remotas, colaborativas e multimídia – requerem diferentes níveis de Qualidade de Serviço, que não são oferecidos originalmente pelas redes IP, inclusive a Internet. A primeira motivação dessa dissertação é a necessidade de definir cenários de aplicação, eleger uma arquitetura de QoS em redes IP e especificar os parâmetros de QoS das aplicações envolvidas, consideradas dentro do escopo do Projeto InfraVIDA;

- As aplicações multimídia de telemedicina previstas para o Projeto InfraVIDA tendem a consumir grande parte dos recursos da rede. Por exemplo, a alocação de sessões de Videoconferência em apoio à Segunda Opinião Médica, e de eventuais sessões concorrentes de Vídeo sob Demanda, pressupõe que sejam levados em consideração o conjunto de recursos da rede e os contratos de nível de serviço, de modo a não comprometer a operação das aplicações vigentes na rede. A segunda motivação dessa dissertação é então propor uma solução integrada que permita o gerenciamento dinâmico dos recursos de rede e o controle de admissão de novas sessões de aplicações dependentes de QoS no contexto do Projeto InfraVIDA;

- A viabilidade de Videoconferência e a qualidade de visualização de imagens médicas pressupõem a configuração dos elementos de rede para alocação de recursos que garantam QoS para determinadas aplicações como a Videoconferência e Imagem/Vídeo sob Demanda. A terceira motivação dessa dissertação consiste então em avaliar a viabilidade técnica da solução, através de simulações dos cenários e mecanismos de QoS em laboratório, usando uma rede protótipo experimental.

As contribuições dessa dissertação consistem em:

- Definir uma especificação dos parâmetros de Qualidade de Serviço (QoS – *Quality of Service*) para os serviços de Videoconferência, Imagem e Vídeo sob Demanda do InfraVIDA, tais como, requerimentos mínimos de banda, atraso, variação de atraso (*jitter*) e perda de pacotes, que associados compõem um conjunto finito de opções associadas a perfis de tráfego ou classes de serviço;

- Propor uma arquitetura de suporte de Qualidade de Serviço para os serviços de Vídeo sob Demanda e Videoconferência que contemple uma solução de integração da rede de vídeo

digital a um elemento gerenciador de políticas ao nível de plano de controle, de forma a permitir o gerenciamento dinâmico de recursos baseado em políticas. O gerenciador de políticas – ou *Bandwidth Broker* - permite a alocação dinâmica de recursos para as aplicações de acordo com contratos de nível de serviço pré-estabelecidos;

- Propor cenários de implantação dos serviços do InfraVIDA, de acordo com a disponibilidade de recursos de rede nas localidades remotas e avaliar a viabilidade técnica dos cenários através de simulações em laboratório com a implementação de software de gerenciamento de políticas de QoS e de mecanismos *DiffServ* em plataformas Linux;

- Oferecer uma visão das tendências de evolução do campo de gerenciamento baseado em políticas, em termos de Qualidade de Serviço em redes IP.

1.3 ORGANIZAÇÃO DA DISSERTAÇÃO

A dissertação está organizada em sete capítulos.

O presente capítulo introduz o contexto no qual a dissertação foi desenvolvida, e dá uma visão global da dissertação.

O capítulo 2 faz uma revisão dos conceitos de Qualidade de Serviço, das arquiteturas de QoS em redes IP e de gerenciamento baseada em políticas. Estas informações são necessárias para facilitar a leitura dos capítulos que se seguem.

O Projeto InfraVIDA é apresentado no capítulo 3.

O capítulo 4 apresenta as especificações para atendimento dos requisitos de QoS das aplicações multimídia do Projeto InfraVIDA e propõe uma arquitetura de suporte de Qualidade de Serviço que contempla a integração das aplicações de Videoconferência e Vídeo sob Demanda à estrutura de gerenciamento de recursos baseado em políticas, implementado pelo *Bandwidth Broker*.

O capítulo 5 descreve um cenário de validação com a implementação do *Bandwidth Broker* em Java em laboratório e validação dos resultados da aplicação de mecanismos de QoS *DiffServ* em rede protótipo experimental Linux na simulação de diversos cenários de aplicação para o Projeto InfraVIDA.

O capítulo 6 apresenta as tendências de evolução do campo de gerenciamento baseado em políticas, em termos de Qualidade de Serviço em redes IP.

O capítulo 7 conclui a dissertação com as perspectivas e questões abertas.

CAPÍTULO 2 QUALIDADE DE SERVIÇO EM REDES IP

Novas aplicações, entre elas as de telemedicina, tendem a aumentar o volume de tráfego multimídia na Internet. Estas aplicações têm requisitos diferentes das aplicações de dados e diferentes entre si. O vídeo requer maior largura de banda e baixa taxa de erro, enquanto a voz sobre IP (VoIP) requer atraso mínimo fim-a-fim, mas pode suportar uma taxa de erro maior que o vídeo, por exemplo. Estes requisitos diferenciados introduzem a noção de Qualidade de Serviço, à qual são associados parâmetros que descrevem o desempenho da rede como atraso, perda de pacotes, variação de atraso (*jitter*) e taxa de erro.

A Internet atual não faz distinção entre os diferentes tipos de tráfego e seus requisitos. O tráfego é encaminhado da melhor maneira possível (*best effort*), mas não há nenhuma garantia de Qualidade de Serviço fim-a-fim. Os roteadores tipicamente encaminham os pacotes à medidas que eles chegam (algoritmo FIFO – *first-in, first-out*). Se ocorrer uma situação de congestionamento onde é ultrapassada a capacidade dos *buffers* de um roteador, os pacotes são simplesmente descartados. Isto pode ser inaceitável para alguns tipos de aplicação multimídia, como por exemplo, a intervenção em uma cirurgia remota em telemedicina. Também não faz sentido que pacotes de voz aguardem a transmissão de dados de tamanho variável, que introduzem variação de atraso, quando o atraso é um requisito importante para a voz e não para os dados.

Esse capítulo apresenta mecanismos de tratamento e priorização de tráfego pelos roteadores, uma arquitetura de Qualidade de Serviço que permita a sistematização do tratamento do tráfego pela rede fim-a-fim e, além disso, uma arquitetura de gerenciamento baseado em políticas.

O *Internet Engineering Task Force* (IETF) definiu duas arquiteturas para endereçar o problema de QoS: - *Integrated Services* (*IntServ*) e *Differentiated Services* (*DiffServ*).

A arquitetura *IntServ*, (BRADEN; CLARK; SHENKER, 1994) primeira tentativa de introduzir QoS em redes IP, define duas classes de serviço, além do *best effort*: serviço Garantido e serviço de Carga Controlada, sendo o serviço Garantido utilizado por aplicações de tempo real com exigências em termos de banda passante e atraso, enquanto o serviço de Carga Controlada oferece ao tráfego IP desempenho semelhante ao de uma rede IP tradicional

não sobrecarregada e é portanto adequado a aplicações que possam se adaptar ao estado da rede. Na arquitetura *IntServ*, os recursos em cada roteador no caminho são reservados para cada fluxo - caracterizado por endereço IP de origem e de destino e porta de aplicação de origem e de destino - graças ao protocolo de sinalização *Resource Reservation Protocol* (RSVP) (BRADEN et al., 1997).

A manutenção dos estados de reserva de recursos para cada fluxo em cada roteador *IntServ* gera um problema de escala, já que um roteador no meio da rede pode ter que manter milhões de estados. Para endereçar este problema, a arquitetura *DiffServ* foi desenvolvida. Esta arquitetura não provê QoS por fluxos individualizados, mas sim, por agregados de fluxo.

Na arquitetura *DiffServ*, uma identificação de classe de serviço *Class of Service* (Cós) é codificada no cabeçalho dos pacotes IP, de modo que agregados de tráfego de mesma classe tenham um mesmo tratamento específico de condicionamento para adequação aos critérios de admissão pela rede, enfileiramento e encaminhamento dos pacotes pelos roteadores no caminho. A discriminação do tráfego de uma aplicação é feita pela especificação dos seus parâmetros de QoS: banda, atraso, variação de atraso (*jitter*) e perda de pacotes. Estes parâmetros associados compõem um conjunto finito de opções associadas a agregados de tráfego ou classes de serviço na arquitetura *DiffServ* (BLAKE et al., 1998).

A diferenciação do tráfego em classes de serviço (CoS) é feita marcando-se o *Differentiated Services Code Point* (DSCP) no cabeçalho dos pacotes IP (NICHOLS et al., 1998). Os pacotes classificados e marcados na entrada da rede, recebem tratamento apropriado associado a políticas de roteamento específicas para cada classe, chamado PHB (*Per-Hop Behavior*), que é aplicado aos pacotes em cada roteador ou elemento de rede, de forma a garantir banda mínima e limites máximos de atraso e variação de atraso para conjuntos de aplicações com requisitos de QoS semelhantes ou *Behavior Aggregates* (BAs). Como o campo DSCP tem 6 *bits*, o número máximo de PHBs é 64 e não há problemas de escalabilidade.

Na arquitetura *DiffServ*, a solicitação de recursos de QoS não é feita por meio de sinalização e sim através de um Contrato de Serviço – *Service Level Agreement* (SLA) (BLAKE et al., 1998), negociado entre o cliente e a rede, e cujas premissas estão de acordo com as políticas definidas pela administração da rede. Os parâmetros de QoS fazem parte do *Service Level Specification* (SLS) - parte técnica do contrato - e, uma vez estabelecidos, são fornecidos através de configuração de facilidades de priorização, enfileiramento, escalonamento e até mesmo descarte seletivo nos roteadores.

O aspecto dinâmico das redes é um dos grandes desafios à implementação de QoS. A definição de classes de serviço e o gerenciamento de políticas requerem conhecimento das aplicações e do comportamento do tráfego na rede. Outro desafio é o fato de que os mecanismos de QoS associados têm que ser provisionados nos roteadores.

Considere-se a situação em que múltiplas estações clientes associadas a um único contrato de serviço, se conectam a diferentes pontos de entrada da rede e iniciam aplicações que requerem dinamicamente acesso aos serviços e recursos da rede, que deve atender aos diversos clientes, sem prejuízo de um ou de outro. Verifica-se que o gerenciamento de recursos e de QoS na rede torna-se bastante complexo. Este é exatamente o cenário do Projeto InfraVIDA, onde a proposta é que múltiplos postos de saúde distantes se conectem a uma rede hospitalar usando uma infra-estrutura de rede pública. No caso, enquanto projeto de pesquisa, o InfraVIDA é suportado pela Rede Nacional de Pesquisa (RNP), que atende a comunidade acadêmica.

Uma solução de gerenciamento é proposta com a introdução de uma entidade lógica chamada *Bandwidth Broker* (BB) (JACOBSON, V.; NICHOLS, K.; ZHANG, 1999) responsável pela configuração automática dos roteadores, controle de admissão de fluxos das aplicações e também pela negociação dos contratos de serviço (SLAs) entre um provedor de serviço e um usuário. O *Bandwidth Broker* retém a base de conhecimento das políticas da rede, tem total conhecimento da topologia de rede e da disponibilidade e alocação dos recursos de rede do seu domínio, incluindo roteadores e enlaces. Além de gerenciar recursos do seu domínio, ele pode negociar com *Bandwidth Brokers* de domínios vizinhos para permitir o fluxo de tráfego entre domínios. A arquitetura do *Bandwidth Broker* é apresentada em seção subseqüente.

Vale a pena mencionar que os modelos *IntServ* e *DiffServ* podem coexistir para prover QoS fim-a-fim para aplicações individualmente. O modelo *DiffServ* provê QoS a nível de classe, mas pode surgir a necessidade de garantir QoS a nível de um fluxo específico. Os sistemas operacionais nos hospedeiros atuais suportam RSVP, o que faz com que as aplicações ativas em hospedeiros fora do domínio *DiffServ* possam solicitar reserva de banda usando RSVP. O domínio *DiffServ* passa as requisições de reserva de recursos transparentemente para o hospedeiro de destino, enquanto provê tratamentos diferenciados baseados em políticas dentro da nuvem *DiffServ*.

A seguir detalharemos alguns conceitos relevantes em QoS em redes IP.

2.1 MECANISMOS DE TRATAMENTO DE TRÁFEGO NOS ROTEADORES

Os roteadores implementam diversos mecanismos de tratamento de tráfego a fim de permitir a priorização, policiamento, condicionamento, descarte seletivo e escalonamento do tráfego das diversas aplicações. A Figura 2-1 ilustra os mecanismos de tratamento de tráfego em um roteador (MARTINS et al., 2003). Descreveremos sucintamente os mais significativos.

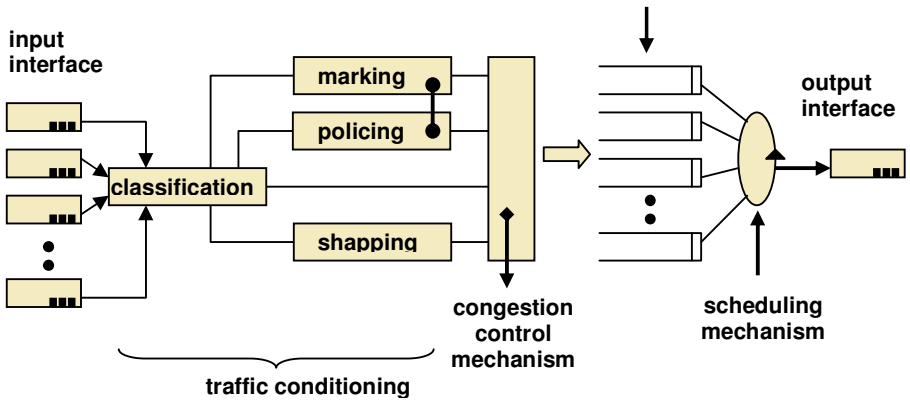


Figura 2-1: Mecanismos de tratamento de tráfego em um roteador (MARTINS et al., 2003)

2.1.1 Mecanismos de Condicionamento de Tráfego

Usados nos roteadores de acesso ou nos roteadores de borda de um domínio administrativo, os condicionadores de tráfego incluem recursos de classificação, marcação, policiamento e suavização de tráfego. A classificação consiste em verificar o conteúdo do cabeçalho do pacote IP para determinar que tipo de tratamento o pacote deva receber. A marcação consiste em gravar ou re-gravar rótulos eventualmente associados aos valores de precedência do pacote IP. O policiamento consiste em verificar conformidade com os parâmetros de tráfego pré-definidos. Os pacotes excedentes podem ter a sua precedência de descarte aumentada (ou serem simplesmente descartados). A suavização do tráfego consiste fundamentalmente em atrasar pacotes para ajustar a taxa de entrada na rede conforme o perfil contratado, se necessário.

As funções de policiamento e condicionamento (suavização) de tráfego são implementadas alternativamente por mecanismos '*leaky bucket*' ou '*token bucket*'.

O *leaky bucket* é um mecanismo que permite controlar o tráfego de entrada em um *buffer* para que ele seja apresentado à rede como um fluxo de tráfego contínuo. Emula um balde que recebe o tráfego dentro do limite do seu tamanho e 'pinga', liberando um dado

número de *bytes,* a cada unidade de tempo. Se em um dado momento o balde 'enche', o tráfego subseqüente é descartado, até que algum espaço seja liberado no balde, que 'pinga' na próxima unidade de tempo. O tamanho do *buffer* e a taxa de transmissão em *bytes* são configuráveis. É usado para limitar a taxa máxima de tráfego que é aceito pela rede.

O *token bucket* é a abstração de um mecanismo de policiamento com capacidade estendida de condicionamento de tráfego, cujos parâmetros são a taxa média de tráfego e o tamanho da rajada (ou *burst*). O controle do volume de tráfego a ser transmitido é baseado no número de fichas (ou *tokens*), unidades de tráfego em *bytes*, presentes no balde.

No *token bucket* o balde suporta até *'b'* fichas, ou unidades de tráfego em *bytes*. Novas fichas que podem potencialmente ser adicionadas ao balde são gerados a uma taxa de *'r'* fichas por unidade de tempo. Novas fichas são adicionadas, se houver espaço no balde, ou seja, o balde tem no máximo *'b'* fichas. O controle do volume de tráfego é feito da seguinte maneira: para que um pacote seja transmitido na rede, ele deve retirar do balde o número de fichas equivalentes ao seu tamanho. Se não houver fichas suficientes, ele deve ser retido no *buffer* à espera de novas fichas. Como há, no máximo, *'b'* fichas no balde, o tamanho máximo da rajada é de *'b' bytes*. Como a taxa de geração de fichas é *'r'*, a taxa máxima de tráfego que pode entrar na rede a qualquer momento é de *'rt + b'*. Ou seja, a taxa de geração de fichas limita o tráfego médio aceito pela rede (KUROSE, J.; ROSS, K., 2003), como ilustrado na Figura 2-2.

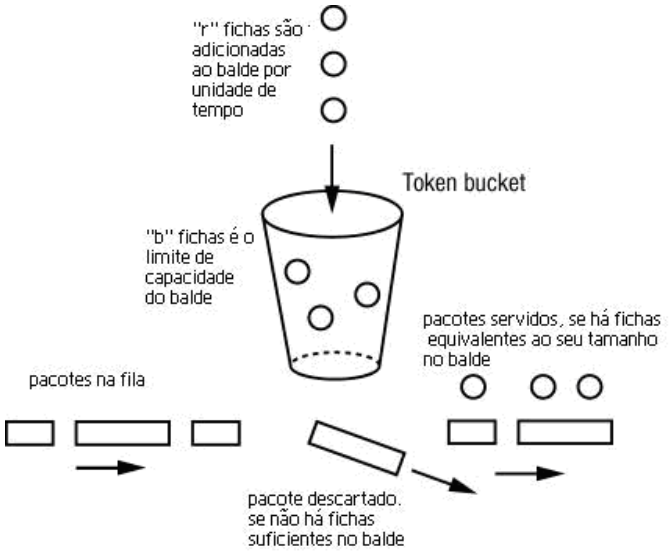


Figura 2-2: Mecanismo de policiamento de tráfego usando *token bucket* (www.cesnet.cz)

O *token bucket* é adequado para o condicionamento de tráfego de entrada com taxa muito variável (rajada ou *burst).* Dois *token buckets* em série (GUERIN, R; HEINANEN, J.,

1999) podem ser usados para controlar tanto o tráfego médio *Commited Information Rate* (CIR) quanto o pico máximo de tráfego *Peak Information Rate* (PIR). O mecanismo permite o policiamento de parâmetros que em geral constam de um contrato de nível de serviço: a banda contratada, a rajada contratada e a rajada em excesso. De outro modo, o uso de múltiplos *token buckets* em paralelo serve à implementação de filas com múltiplos níveis de precedência de descarte.

2.1.2 Mecanismos de Escalonamento de Tráfego

Mecanismos de escalonamento ou gerenciamento de filas determinam a ordem em que pacotes conformes alocados em filas específicas recebem serviço. São descritos a seguir os mecanismos mais importantes.

First-In, First-Out (FIFO) é a disciplina de fila padrão na maioria dos roteadores usada em conjunto com informações de roteamento para encaminhamento dos pacotes. Os pacotes são encaminhados à medida que chegam. É altamente eficiente quando a banda disponível e a capacidade de processamento dos roteadores são adequadas ao volume de tráfego. Mas se há sobrecarga da rede o tamanho da fila cresce, gerando atrasos e perda dos últimos pacotes quando a fila estoura, com conseqüente degradação dos níveis de serviço.

Priority Queueing (PQ) é um mecanismo de enfileiramento de prioridade absoluta, com preempção dos recursos (*strict priority)*. Permite níveis de granularidade ou várias filas de preferência de serviço. O tráfego em filas PQ é sempre servido, em detrimento do tráfego em outras filas. É recomendável que seja usado em conjunto com mecanismo de policiamento que limite a banda alocada à PQ, de modo a evitar que o tráfego prioritário monopolize os recursos e bloqueie outras classes de tráfego de menor prioridade. Um impacto de desempenho pelo processamento de re-ordenamento de pacotes pelo roteador pode ser notável em enlaces de alta velocidade.

Generalized Processor Sharing (GPS) (PAREKH, A. K.; GALLAGER, R. G., 1992) é um algoritmo que serve uma quantidade infinitesimal de dados de cada fila, de forma eficiente, com flexibilidade e com compartilhamento justo da banda. Apesar de ser uma abstração, ou seja, uma solução apenas teórica e não realizável, já que o tamanho dos pacotes não é infinitesimal, serve de referência ao desempenho de disciplinas de fila que se propõem a implementar o algoritmo.

Class-Based Queueing (CBQ) (FLOYD, S.; JACOBSON, V., 1995) é um mecanismo que aloca frações da banda da interface a múltiplas filas e garante o serviço do tráfego em

cada fila, medido em *bytes*, a cada rodada do escalonador, até a banda configurada. É um mecanismo mais justo que o PQ, pois garante que todas as filas sejam servidas. Além disso, consegue garantir razoavelmente a latência e não aumenta substancialmente o *jitter* (variação de atraso). No entanto, o processamento de alocação dos pacotes nas filas acaba impactando o desempenho em enlaces de alta velocidade. O CBQ é apropriado apenas para enlaces de baixa velocidade.

Weighted Round Robin (WRR) é uma proposta de mecanismo que tem intenção de alocar percentuais da banda da interface de saída a cada fila. Para isso serve um dado número de pacotes em cada fila a cada rodada do escalonador. Esse número normalizado é igual ao peso ou prioridade atribuída dividido pelo tamanho médio do pacote. Como é difícil determinar o tamanho médio do pacote de antemão, essa disciplina não garante justiça no compartilhamento da banda e ainda causa um *jitter* significativo.

Deficit Round Robin (DRR) (SHREEDHAR, M.; VARGHESE, G., 1995), versão modificada do WRR, distribui melhor o serviço de pacotes de tamanhos diferentes sem conhecer o tamanho médio dos pacotes *a priori*. Enquanto o WRR serve todas as filas não vazias, o DRR serve pacotes à frente de filas não vazias, cujo contador de déficit não exceda o tamanho do pacote a ser servido. O contador de déficit, que serve de parâmetro para o serviço de pacotes, é inicializado com um dado tamanho máximo de pacote, ou *quantum.* Este valor é subtraído do tamanho do pacote a ser tratado. Pacotes que excedam este tamanho são retidos até a próxima rodada do escalonador, quando o contador de déficit é acrescido de um *quantum*. Se o tamanho do pacote for menor que o contador de déficit ele é servido, e o valor do contador de déficit é decrescido de um valor igual ao do tamanho do pacote. O DDR é adequado a enlaces de alta velocidade.

Fair Queueing (FQ) (NAGLE, 1985) garante tratamento prioritário aos fluxos de tráfego de baixo volume que, em geral, correspondem ao tráfego de aplicações em tempo real que exigem tempo de resposta e comportamento previsível de tráfego. O mecanismo FQ identifica e distribui pacotes de diferentes fluxos nas filas, de acordo com o tamanho dos pacotes, e serve prioritariamente os menores, enquanto distribui o restante da banda com os demais filas, de forma justa. O *Weighted Fair Queueing* (WFQ) (DEMERS et al., 1990) leva em conta a precedência configurada no cabeçalho do pacote IP ao escalonar o tráfego dos fluxos nas filas, atribuindo pesos de atendimento e permitindo que o usuário ou administrador interfira na priorização do tráfego.

Low Latency Queueing (LLQ) é uma implementação proprietária Cisco (ODOM, 2003) que combina os mecanismos de PQ e WFQ. A Cisco chama de *Class-Based Weighted*

*Fair Queueing (*CBWFQ) o mecanismo WFQ que leva em conta o valor do DSCP, o qual define a que classe o pacote está associado, para fins de *DiffServ*. O LLQ é um mecanismo CBWFQ, com o diferencial de que algumas filas podem ser configuradas como filas de prioridade estrita (PQs), mantendo-se um mecanismo de policiamento, que descarta o tráfego que exceder um limite máximo nestas filas. A combinação de mecanismos permite otimizar a latência do tráfego nas filas de prioridade estrita, enquanto mantêm a equidade no serviço às demais filas.

2.1.3 Mecanismos de Controle de Congestionamento

Mecanismos de controle de congestionamento detectam congestionamento potencial e descartam pacotes randomicamente, para que a rede ajuste a taxa de transporte usando os conhecidos mecanismos de ajuste de tamanho de janelas do TCP (JACOBSON, 1988).

O *Transport Control Protocol (*TCP) implementa um mecanismo de janelas deslizantes para controle de fluxo. Para evitar que a origem envie dados mais rapidamente do que o destino é capaz de processar, a cada segmento TCP recebido, o destino especifica o tamanho da janela de recepção, ou seja, a quantidade de dados (em *bytes*) que ele espera receber a seguir naquela conexão. A origem pode enviar apenas aquela quantidade de dados até receber uma confirmação de recebimento e uma atualização do tamanho da janela de recepção. Confirmações de recepção ou a sua falta dentro do prazo de temporizadores, são interpretadas pela origem para inferir as condições de congestionamento na rede.

Para o controle de congestionamento na rede, o TCP mantém um mecanismo de janela semelhante ao usado no controle de fluxo, que limita o número total de pacotes que transitam fim-a-fim sem confirmação de recebimento. A janela de congestionamento é inicializada com um tamanho pequeno (*slow start*), mas cresce rapidamente a cada confirmação recebida, dobrando de tamanho a cada período estimado de ida e volta entre origem e destino -*Round Trip Time* (RTT). Quando há congestionamento ou um pacote é perdido, confirmações duplicadas são recebidas na origem e a janela de congestionamento é reduzida ao seu tamanho inicial. Quando o congestionamento na rede é muito severo, pacotes de diferentes fluxos são descartados por estouro das filas nos nós intermediários ao mesmo tempo, fazendo com que múltiplas fontes reduzam simultaneamente as suas janelas de congestionamento, mas que também as ampliem simultaneamente, voltando a alimentar o congestionamento.

Random Early Detection (RED) (FLOYD, S.; JACOBSON, V., 1993) é um mecanismo de gerenciamento ativo de filas desenvolvido com o objetivo de evitar que o problema de sincronização global das janelas de TCP leve a um colapso de tráfego na rede.

Em contraste com mecanismos tradicionais de gerenciamento de filas que descartam pacotes apenas quando o *buffer* estoura, o algoritmo RED descarta pacotes probabilisticamente. A probabilidade de descarte aumenta à medida que o tamanho médio estimado da fila aumenta, ou seja, responde a um tamanho de fila médio no tempo, não ao tamanho imediato da fila. Portanto, se a fila estava praticamente vazia recentemente, o RED não tende a descartar pacotes (a menos que o *buffer* estoure!). No entanto, se a fila estava relativamente cheia recentemente, indicando congestionamento persistente, é mais provável que novos pacotes sejam descartados pelo RED. Pacotes descartados em fluxos aleatórios em diferentes momentos sinalizam às fontes de que devem reduzir o tráfego enviado, mas evitam que todas as fontes o façam no mesmo momento, ou seja, evitam a sincronização global das janelas do TCP. O algoritmo RED é ilustrado na Figura 2-3. O *Weighted Random Early Detection* (WRED) leva em conta o valor configurado no campo DSCP no cabeçalho dos pacotes IP ao selecionar os fluxos para descarte nas filas.

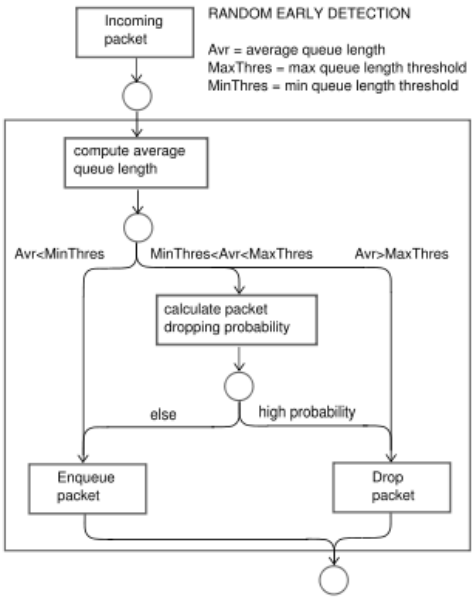


Figura 2-3: Mecanismo de controle de congestionamento usando RED (www.computerbase.de)

A implementação de mecanismos de gerenciamento ativo de filas (*active queue management*) é fortemente recomendável nos roteadores da Internet (BRADEN et al., 1998).

O mecanismo RED (ou o WRED) é referendado como solução para, em conjunto com o mecanismo de janelas, conterem congestionamentos causados por fluxos TCP.

No entanto, dois tipos de aplicação geram preocupação com o futuro da Internet: em primeiro lugar, implementações de TCP "rápido", não compatível com a especificação padrão do TCP, ou métodos de obtenção de desempenho agressivo do TCP, como abrir múltiplas conexões simultâneas com um mesmo destino; depois, o crescimento de aplicações que fazem uso do *User Datagram Protocol* (UDP) (como VoIP, *streaming* e *multicast* de áudio e vídeo), já que o UDP é não-responsivo a notificações de congestionamento pela rede. Em geral as aplicações de *streaming* incorporam mecanismos de controle de tráfego, mas é preciso esforço de pesquisa para desenvolver mecanismos que permitam à rede se defender de fluxos não responsivos ou agressivos.

A adição de notificações explícitas de congestionamento *Explicit Congestion Notification* (*ECN)* é um mecanismo adicional de prevenção de congestionamento à arquitetura da Internet (RAMAKRISHNAN, K.; FLOYD, S., 1999). A implementação requer um campo ECN de dois *bits* no cabeçalho IP: o bit ECT (*ECN-Capable Transport*) para ser habilitado pela fonte para indicar que suporta a facilidade de ECN e o bit *Congestion Experienced* (CE) para ser habilitado pelo roteador para indicar congestionamento aos hospedeiros.

2.2 ARQUITETURA *DIFFSERV*

O principal objetivo do IETF na definição da arquitetura *DiffServ* foi possibilitar a implementação de diferenciação de serviços na Internet de forma escalável. Isto se viabiliza por meio da agregação de tráfego em CoSs categorizadas pela marcação do DSCP no cabeçalho dos pacotes IP. Os pacotes classificados e marcados recebem um tratamento específico para cada classe em todos os nós ao longo da rota. Os recursos de rede são alocados aos agregados de tráfego de acordo com políticas de provisionamento de serviço. A abordagem *DiffServ* no provisionamento de qualidade de serviço em redes IP emprega um conjunto limitado e bem definido de elementos a partir dos quais uma variedade de tratamentos de agregados de tráfego pode ser construída. O modelo tem dois elementos-chave: - a marcação dos pacotes na entrada da rede usando o DSCP e o tratamento consistente dos pacotes pelos elementos de roteamento da rede (PHB).

2.2.1 DSCP (Differentiated Services Code Point)

O DSCP, um padrão de 6 *bits* que é parte do campo DS (8 *bits*), definido na RFC 2474 (NICHOLS et al., 1998) como uma atualização do campo *Type of Service* (ToS) (8 *bits*) no cabeçalho do IPv4 ou do campo *Traffic Class* (TC) (8 *bits*) no cabeçalho do IPv6, como ilustrado na Figura 2-4, é usado para marcar um pacote para que ele receba um tratamento específico (PHB) em cada nó. Até 64 classes ou agregados de tráfego podem ser suportados, apesar de apenas alguns serem padronizados. A padronização é importante para uso entre domínios, interoperabilidade entre fabricantes e avaliação consistente do tratamento esperado dos agregados de tráfego na rede.

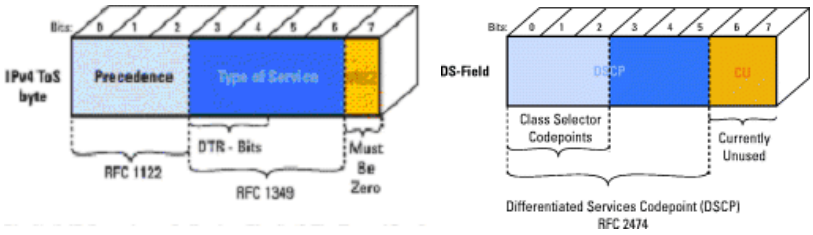


Figura 2-4: Campo ToS original (IPv4) e campo DSCP (CISCO, 2001)

Os *bits* 6 e 7 dos campos ToS do IPv4 e TC do IPv6 não usados foram designados ao campo ECN. O *bit* 6 como ECT e o *bit* 7 como CE (RAMAKRISHNAN, K.; FLOYD, S.; BLACK, D., 2001).

2.2.2 Per-Hop Behavior (PHB)

Um PHB é definido como um tratamento de encaminhamento externamente observável aplicado a um agregado de tráfego (BA – *Behavior Aggregate*) em um nó conforme ao *DiffServ* (BLAKE et al., 1998). A especificação do *DiffServ* não define nenhum serviço em particular, ele simplesmente define o tratamento que o pacote vai receber em cada nó da rede. Todos os pacotes que tem o mesmo valor de DSCP em uma mesma direção são reconhecidos como um agregado de tráfego (BA). Portanto, pacotes de diversas aplicações ou fontes podem pertencer a um mesmo BA, ou seja, um agregado de fluxos da mesma classe. Em termos práticos, um PHB indica o tratamento dado aos pacotes no que se refere à priorização de serviço, enfileiramento, policiamento e condicionamento de um mesmo agregado de tráfego, configurado de acordo com alguma política ou contrato de serviço (SLA). O IETF definiu alguns padrões de PHBs *DiffServ*.

O **PHB *Expedited Forwarding* (EF)** deve prover um serviço de baixa perda de pacotes, baixa latência e baixa variação de atraso, com garantia de banda mínima fim-a-fim (DAVIE et al., 2002). EF corresponde ao serviço *premium*, que corresponde ao de uma linha privativa virtual, com a garantia de banda mínima. Perda, latência e *jitter* são conseqüências da formação de filas. Para evitá-las e fazer com que o tráfego preferencial encontre fila mínima ou vazia, independentemente de outros tráfegos concorrentes no nó, o PHB EF precisa fazer com que a taxa de entrada seja menor ou igual que a taxa de saída em cada nó. EF pode ser implementado, por exemplo, usando *Priority Queueing* (PQ), mecanismo de enfileiramento de prioridade absoluta, com preempção dos recursos. Para evitar que o tráfego EF monopolize os recursos e bloqueie outras classes de tráfego de menor prioridade é preciso que se limite a banda alocada para o agregado de tráfego EF, implementando-se um mecanismo de policiamento (*token bucket rate limiter*), que descarta o tráfego EF que exceder um limite máximo. Os limites de banda mínima e máxima são configuráveis e devem ser definidos pelo administrador da rede. O valor de DSCP recomendado para o EF é '101110'.

Apesar de o PHB EF prover um serviço preferencial quando implementado numa rede *DiffServ*, ele deve ser aplicado apenas às aplicações mais críticas, pois se uma situação de congestionamento existir não é possível tratar todo o tráfego com alta prioridade. Em geral, aplicações como voz sobre IP (VoIP), vídeo e transações *on line* são candidatas ao serviço EF, pois tem requisitos de baixa perda, baixa latência, baixa variação de atraso e garantia mínima de banda.

O **PHB *Assured Forwarding* (AFxy)** consiste de um grupo de quatro classes de serviço, dentro das quais podem ser alocados três níveis de precedência de descarte. O esquema permite prover níveis diferenciados de garantia de serviço a diferentes agregados de tráfego (BAs) (HEINANEN et al., 1999).

O PHB AF define quatro classes AF1, AF2, AF3 e AF4, sem precedência entre elas. Cada classe AF recebe certa quantidade de recursos (espaço em *buffer* e banda na interface) em cada nó DS (*Differentiated Services*), de acordo com o SLA ou a política do provedor de serviço. Os pacotes são alocados em diferentes filas dependendo da classe de serviço. A recomendação não especifica a implementação do escalonador, mecanismo de atendimento das filas, que pode ser configurado para alocar frações da banda a cada fila, por exemplo, *Class Based Queueing* (CBQ).

Dentro de cada classe AFx, os pacotes podem ser marcados com três níveis de precedência de descarte, ativados se houver congestionamento. O algoritmo de controle de congestionamento *Random Early Detection* (RED) é dos mais utilizados para a

implementação do mecanismo de descarte. A probabilidade de descarte é tal que dP(AFx1) <= dP(AFx2) <= dp(AFx3), onde dP(AFxy) é a probabilidade de descarte de pacotes da classe AFxy. O índice 'y' em AFxy denota a precedência de descarte dentro de uma classe AFx. Este conceito é útil para penalizar fluxos dentro de um BA que excedam a banda alocada à classe. Os pacotes destes fluxos podem ser remarcados por um mecanismo de policiamento para terem maior precedência de descarte em caso de congestionamento.

A Figura 2-5 mostra os valores de DSCP para as classes AFxy, na forma 'xyzab0', onde 'xyz' representa a classe e 'ab' representa a precedência de descarte.

Prioridade De Tratamento

Precedência de Descarte

Classe #1	Classe #2	Classe #3	Classe #4
(AF11) 001010	(AF21) 010010	(AF31) 011010	(AF41) 100010
(AF12) 001100	(AF22) 010100	(AF32) 011100	(AF42) 100100
(AF13) 001110	(AF23) 010110	(AF33) 011110	(AF43) 100110

Figura 2-5: Valores de DSCP para o PHB AF (CISCO, 2001)

O padrão *DiffServ* exige que os nós conformes selecionem PHBs de acordo com o campo DSCP de 6 *bits*. **PHBs *Class-Selector*** com valores de DSCP 'xxx000', preservam compatibilidade com o *IP Precedence*, um campo de 3 *bits* parte do ToS do IPv4. Para garantir que pacotes recebidos com um valor de DSCP desconhecido não provoquem mau funcionamento da rede, devem ser mapeados para o **PHB *Default*** (BLAKE et al., 1998), com valor de DSCP '000000', que equivale ao serviço de melhor esforço (*best effort*) equivalente ao da Internet atual.

Combinações de PHBs EF, AF e do serviço de melhor esforço da Internet atual, permitem uma variedade de esquemas de diferenciação de tráfegos sensíveis a QoS, que sejam transmitidos através de domínios *DiffServ*. Esta diferenciação de tráfego em classes pode ser usada, seja por domínios administrativos, seja por operadoras de telecomunicações para servir e tarifar seus usuários de maneira diferenciada.

O Quadro 2-1 ilustra um exemplo de política de QoS:

Serviço Premium
Garantia de banda de 128 kbps (DSCP EF)
Serviço Ouro
50% da banda disponível, tráfego em excesso marcado com prioridade de descarte baixa (DSCP AF13)
Serviço Prata
25% da banda disponível, tráfego em excesso marcado com prioridade de descarte média (DSCP AF23)
Serviço Bronze
10% da banda disponível, tráfego em excesso marcado com prioridade de descarte alta (DSCP AF33)
Serviço Básico
Sem garantias (DSCP default)

Quadro 2-1: Exemplo de política de QoS (AUTOR, 2004)

2.2.3 Terminologia *DiffServ*

Para que se possa determinar que tipo de tratamento (*Per-Hop Behaviour*) um pacote deve receber, uma forma de negociação de serviço precisa ser levada adiante antes que os pacotes sejam transmitidos. A arquitetura *DiffServ* prevê um contrato de serviço *Service Level Agreement* (SLA) entre um usuário e um provedor de serviço que especifica que tipo de serviço o usuário deve receber (BLAKE et al., 1998). O SLA é um contrato formal que cobre tanto questões comerciais e aspectos legais ligados ao negócio quanto à especificação técnica do serviço, incluindo parâmetros de desempenho e disponibilidade. O *Service Level Specification* (SLS) é parte do SLA e contém exclusivamente detalhes técnicos, sendo essencialmente uma tradução das especificações do SLA em informações necessárias à configuração dos roteadores. No contexto do *DiffServ*, o SLS define parâmetros como DSCP e PBHs específicos. Ao SLS é associado um conjunto de parâmetros e métricas *Traffic Conditoning Specification* (TCS) na terminologia *DiffServ* (GROSSMAN, 2002) , que devem ser monitorados para verificar o cumprimento do SLA.

2.2.3 Arquitetura *DiffServ*

A Figura 2-6 mostra a arquitetura *DiffServ* e seus principais elementos. Uma região *DiffServ* é composta de um ou mais domínios *DiffServ*, possivelmente sob múltiplas autoridades administrativas.

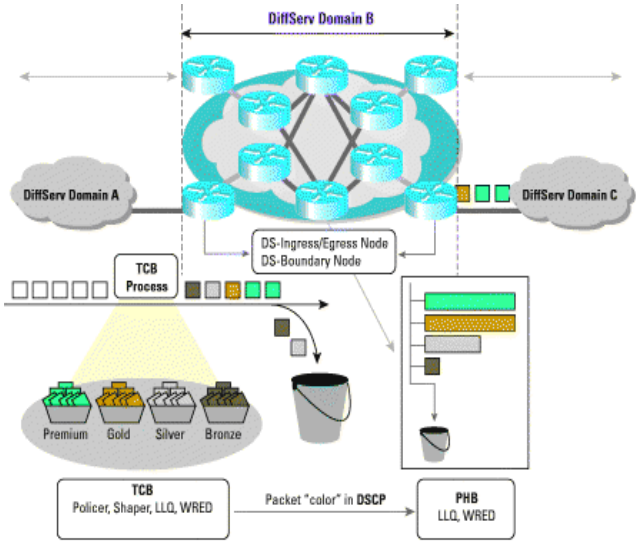


Figura 2-6: Arquitetura *DiffServ* [CISCO, 2001]

Um domínio *DiffServ* é composto de nós DS de entrada, nós DS interiores e nós DS de saída. Os nós de borda (entrada e saída) podem ter funções de fronteira com outros domínios *DiffServ* ou não-*DiffServ*.

Cada domínio é configurado usando DSCPs e diferentes PHBs. Para garantir QoS fim-a-fim, todas as redes na rota do pacote IP devem ser conformes ao padrão *DiffServ*. Para especificar QoS são definidas políticas e contratos de serviço (SLAs).

O condicionamento de tráfego na entrada da rede permite reforçar o controle do uso de recursos e obter parâmetros quantitativos das métricas de desempenho dos elementos de rede. O condicionador de tráfego é controlado por regras definidas no TCS, parte do SLS, que tipicamente incluem o perfil de tráfego (caracterizado por parâmetros associados ao *token bucket*), métricas de desempenho (como banda, atraso, *jitter* e perda) e ações requeridas para o tratamento de tráfego fora do perfil.

Mecanismos de classificação, marcação, policiamento e condicionamento de tráfego *Traffic Conditioning Behavior* (TCB) são tipicamente implementados apenas na entrada da rede. O nó DS de borda classifica os pacotes entrantes em agregados de tráfego (BAs), mede o volume de tráfego contra parâmetros de tráfego pré-definidos para verificar se ele está dentro do perfil contratado, marca os pacotes gravando ou re-gravando o valor de DSCP e, finalmente, coloca os pacotes no *buffer* para condicionar o tráfego à taxa de entrada aceitável pela rede, ou ainda descarta os pacotes em caso de congestionamento. Os nós DS internos aplicam o PHB apropriado empregando mecanismos de controle de congestionamento,

enfileiramento e priorização no escalonamento de serviço. Reduzir o número de operações complexas nos nós interiores torna a operação mais eficiente e escalável.

Devido ao papel determinante dos mecanismos de classificação e condicionamento de tráfego nos nós DS de borda, são eles os pontos de interação do plano de controle de gerência de QoS.

2.2.4 Per-Domain Behavior (PDB)

Aplicações típicas da Internet muito freqüentemente têm origem e destino em domínios diversos, o que faz com que o tráfego fim-a-fim atravesse múltiplos domínios. Para se ter QoS fim-a-fim, o tratamento oferecido às CoSs deve ser consistente nos domínios *DiffServ* que o tráfego atravessa.

Existe a necessidade de um mecanismo entre domínios para negociação de especificação de serviço (SLS) para o tráfego em trânsito, que permita o compartilhamento de políticas de serviço e ainda garanta que a alocação de recursos é suficiente para o uso simultâneo de serviços preferenciais.

O conceito *Per-Domain Behavior* (PDB) (NICHOLS, 2001) é definido como um bloco de técnicas de associação de PHBs específicos (classificadores, condicionadores de tráfego) e configurações particulares que resultem em um conjunto de atributos externamente observáveis no tratamento experimentado por pacotes que atravessam um domínio *DiffServ*. O PDB é caracterizado por métricas específicas definidas em SLSs entre domínios e tem o objetivo de permitir que operadoras construam ofertas de serviços diferenciados para tráfego em trânsito, independentemente do PHBs usados. Este nível de abstração facilita a composição de serviços entre domínios, ocultando detalhes internos das redes, mas oferecendo elementos suficientes para habilitar QoS.

Este é um assunto complexo e ainda em discussão. Até o momento, o único PDB definido é o *Lower Effort* (LE) (BLESS et al., 2003). Usado para enviar tráfego sem nenhuma prioridade através de um domínio *DiffServ*, é esperado que os pacotes sofram perdas ou atrasos quando qualquer outro tráfego estiver presente. Ou seja, é definida uma categoria de tráfego com prioridade inferior à do tradicional tráfego *best effort*, ao qual é assegurado no mínimo acesso igualitário aos recursos disponíveis, enquanto o tráfego *lower effort* só tem acesso aos recursos se não houver outro tráfego presente. Este PDB permite que administradores de rede protejam as suas redes de determinados tipos de tráfego, por exemplo, de aplicações multimídia que tipicamente usam protocolos não adaptativos como o

UDP, ao invés de oferecer qualquer tipo de tratamento preferencial a determinados agregados de tráfego.

2.2.5.Questão em Aberto no *DiffServ*

O *DiffServ* não prevê o controle de admissão dinâmico nos nós de borda. Isto obriga a administração da rede a se certificar da disponibilidade de recursos suficientes para SLAs contratados e torna pouco eficiente a expansão do serviço (JHA; HASSAN, 2002).

Para endereçar esta questão em aberto no *DiffServ,* o IETF propõe uma arquitetura de gerenciamento baseado em políticas (PBNM - *Policy-Based Network Management*). A idéia de uma entidade lógica chamada *Bandwidth Broker*, proposta por Jacobson (JACOBSON, V.; NICHOLS, K.; ZHANG., 1999) e desenvolvida pelo Internet2 *QBone Signaling Design Team* (CHIMENTO et al., 2002) vem complementar os esforços do IETF na definição de uma arquitetura de gerenciamento baseado em políticas.

2.3 GERENCIAMENTO BASEADO EM POLÍTICAS

Como discutido anteriormente, o IETF propôs o *IntServ* e o *DiffServ* como arquiteturas de QoS que suportassem níveis de serviço diferenciados para as diversas aplicações, inclusive as aplicações multimídia e em tempo real. Por exemplo, o *DiffServ* é capaz de prover níveis de serviço fim-a-fim bem definidos entre domínios autônomos, seja entre redes de clientes e provedores ou entre redes de diferentes provedores. Para isto, estes domínios precisam entrar em acordo contratual sobre o tratamento a ser dado aos agregados de tráfego que enviam e recebem um do outro. Estes contratos precisam ser convertidos em ações implementadas e reforçadas pelos elementos de rede. Tudo isto requer a definição de políticas de provisionamento, que, implementadas nos roteadores de borda entre domínios, permitem o mapeamento do tráfego em classes *DiffServ*. Estas questões fazem parte do escopo do PBNM (*Policy-Based Network Management*) ou gerenciamento baseado em políticas, uma área de pesquisa significativa nos últimos anos.

2.3.1. Grupo de Estudos de Protocolo de Alocação de Recursos (RAP WG)

O conceito de política denota uma regra de acesso aos recursos de rede e serviços baseada em critérios administrativos. O *Resource Allocation Protocol Workgroup* (RAP WG)

do IETF, desenvolveu uma arquitetura de gerenciamento baseado em políticas que permite a alocação de recursos e a configuração automática dos dispositivos de rede para suporte a QoS, usando qualquer dos modelos *IntServ* ou *DiffServ*. Para garantir QoS, o sistema deve ser capaz de monitoração e contabilização do uso dos recursos. A entidade responsável pelo controle de admissão pode usar um sistema baseado em políticas para aceitar uma requisição de reserva de recursos (por exemplo, uma mensagem *IntServ* RSVP RESV) ou alocar um determinado PHB *DiffServ* para o fluxo entrante.

O RAP WG definiu uma nomenclatura específica (YAVATKAR, R.; PENDARAKIS, D.; GUERIN, R., 2000). O *Policy Enforcement Point* (PEP) é uma entidade lógica responsável por reforçar as políticas do domínio para os diversos tráfegos. É implementado em roteadores, que podem suportar diversos PEPs ou diferentes instâncias de gerenciamento de políticas de QoS, segurança, etc. O *Policy Decision Point* (PDP) é uma entidade lógica que toma decisões sobre políticas de distribuição de recursos para si e para outros elementos de rede que as requeiram. O PDP tem uma visão dos recursos em todo o domínio administrativo, obtida através dos seus PEPs. O PDP faz controle de admissão, e pode usar protocolos adicionais como o *Remote Authentication Dial In User Service* (RADIUS) para autenticação e contabilização. Um dos protocolos que pode ser utilizado na comunicação entre PEPs e PDP é o *Common Open Policy Protocol* (COPS), criado especificamente para administração, configuração e aplicação de políticas de rede. A Figura 2-7 ilustra a arquitetura proposta pelo RAP WG.

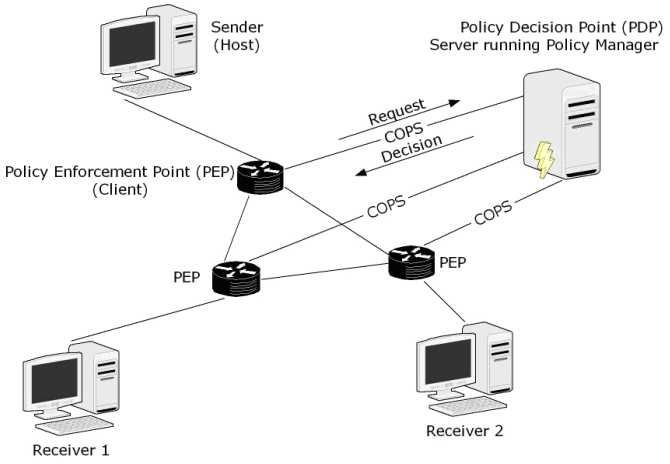


Figura 2-7: PEPs e PDP em arquitetura proposta pelo IEEE RAP WG [students.csci.unt.edu]

2.3.2. COPS (Common Open Policy Protocol)

O protocolo COPS, definido na RFC 2748 (DURHAM et al., 2000), descreve um modelo cliente/servidor simples, onde PEPs estão permanentemente conectados a geralmente um único PDP usando uma conexão TCP, para suportar a aplicação de políticas na rede. O modelo é baseado na idéia do servidor de políticas (PDP) retornar decisões para requisições de políticas feitas por roteadores clientes (PEPs). No COPS, é o PEP que inicializa a conexão TCP com o PDP, por onde envia requisições e recebe decisões do PDP em resposta. Depois de receber uma decisão, o PEP deve enviar uma notificação de que a política especificada na decisão enviada pelo PDP foi aplicada no PEP, com sucesso ou não. Essas notificações são usadas para fins de monitoração e contabilização. Qualquer mudança de estado no PEP deve ser informada imediatamente ao PDP por meio de uma notificação não-solicitada.

A operação do COPS descrita acima e ilustrada na Figura 2-8 é conhecida como modo de *outsourcing*. No *IntServ*, as aplicações cliente em hospedeiros usam mensagens RSVP para encaminhar requisições de reserva de recursos aos roteadores (PEPs) e estes tomam a iniciativa de encaminhar as requisições ao PDP, como no modelo cliente/servidor do COPS.

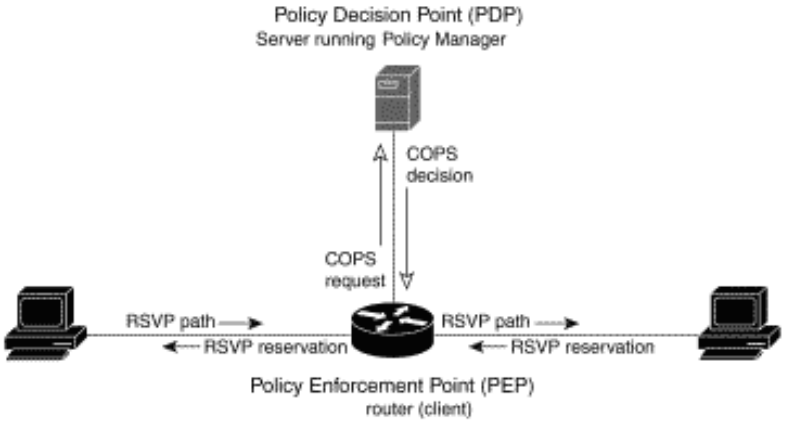


Figura 2-8: Modo *outsourcing* de operação do COPS [www.cisco.com]

No entanto, o modo de *outsourcing* do COPS não se aplica inteiramente à operação do modelo *DiffServ*. No *DiffServ*, as decisões do PDP não são tipicamente provocadas por solicitações dos roteadores, e sim por solicitações de alocação de recursos vindas de clientes externos (aplicações em hospedeiros), encaminhadas diretamente ao PDP.

2.3.3. COPS-PR (Common Open Policy Protocol Provisioning)

Outro modo de operação do COPS é o modo de provisionamento de políticas (*provisioning*), conhecido como COPS-PR e descrito na RFC 3084 (CHAN et al., 2001), desenvolvido para atender ao modelo *DiffServ*. A operação do COPS-PR é ilustrada na Figura 2-9.

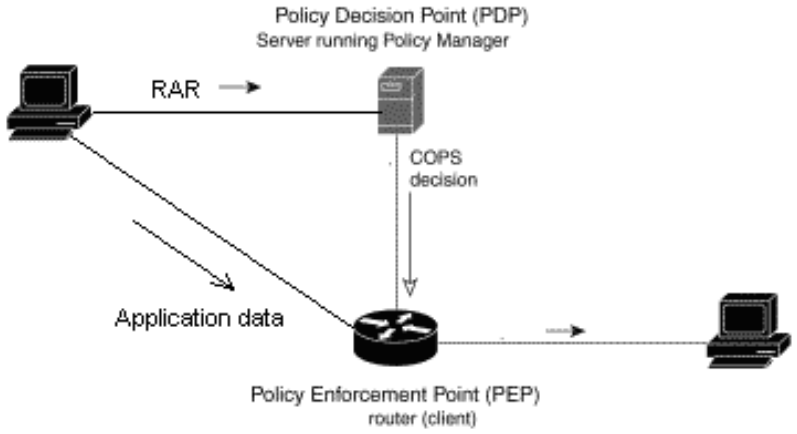


Figura 2-9: Modo *provisioning de* operação do COPS-PR (www.cisco.com)

Uma requisição de alocação de recursos *Resource Allocation Request* (RAR) feita por uma aplicação cliente ao PDP, quando aceita, faz com que o PDP envie uma mensagem com especificações de reconfiguração aos roteadores de borda no domínio (PEPs) para que eles implementem um tratamento específico para os pacotes deste cliente. O fato de que o PDP pode enviar uma decisão não-solicitada para que o PEP mude seu estado caracteriza o modo de provisionamento do COPS-PR.

O protocolo é orientado a eventos, o que significa que o *polling* entre PDP e PEP é eliminado, aumentando a eficiência no que diz respeito ao tráfego na rede. O evento que faz com que o PDP envie dados ao PEP pode ocorrer em uma fonte externa, um importante aspecto da operação do PDP.

2.3.4. Operação do COPS-PR

Ao ser inicializado, um PEP estabelece uma conexão TCP com o PDP e envia sua identificação de tipo de cliente. O PDP então extrai toda a informação de políticas relevantes associadas àquele PEP em particular do seu repositório de dados e envia estes dados ao PEP, que se configura a partir das especificações recebidas. O PEP então envia uma notificação do

sucesso ou não na aplicação da política ao PDP. Durante a operação normal, se o PDP atende a uma requisição de alocação de recursos (RAR) ou detecta uma mudança de política, envia nova mensagem de atualização ou eliminação de configuração ao PEP, que por sua vez, devolve uma notificação ao PDP. O PEP pode ainda requisitar sua configuração ao PDP no momento de sua inicialização.

A Figura 2-10 ilustra a troca de mensagens do COPS-PR na inicialização de um dispositivo (I), na solicitação pelo PEP de atualização de políticas (II) ou na notificação aos PEPs pelo PDP de mudança de políticas causada por evento externo (III).

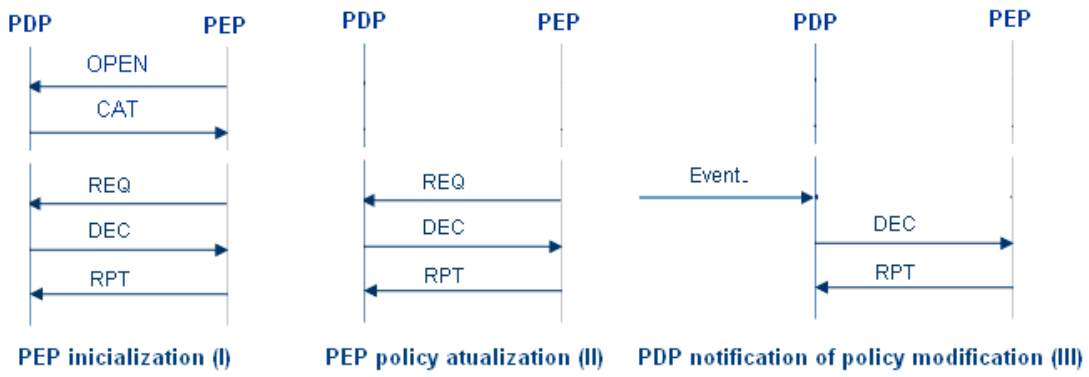


Figura 2-10: Troca de mensagens do COPS-PR (FRANCO et al., 2006)

As principais mensagens do COPS-PR são descritas abaixo:

- *Client-Open* (OPEN): é enviada por um PEP ao PDP para abrir uma sessão. Se o PDP suporta o tipo de cliente uma mensagem *Client-Accept* (CAT) é enviada pelo PDP ao PEP para aceitar a sessão. Senão, o PDP envia a mensagem *Client-Close* (CC) ao PEP.

- *Request* (REQ): é enviada por um PEP a um PDP para requerer dados de configuração. A mensagem de requisição de configuração funciona como uma solicitação do PEP ao PDP de políticas de provisionamento que o PDP possa ter armazenado para o PEP. Isto permite que o PEP obtenha a política mais atual.

- *Decision* (DEC): é enviada do PDP ao PEP de forma solicitada ou não-solicitada. Isto é, uma mensagem DEC pode ser enviada pelo PDP mesmo que o PEP não tenha feito uma requisição (por exemplo, para comunicar uma atualização de política). Este é o aspecto de provisionamento da operação do COPS-PR. Cada mensagem DEC pode conter várias decisões (por exemplo, suprima uma determinada política e instale outra). Uma importante orientação da RFC 3084 é que todas as decisões contidas numa mensagem DEC sejam

instaladas ou, em caso de falha, nenhuma. Neste caso, o PEP volta ao estado anterior ao recebimento da mensagem DEC.

- *Report State* (RPT): é enviada de um PEP ao PDP de forma solicitada em resposta a um DEC enviado pelo PDP e funciona como uma confirmação (*acknowledgement*) de que as políticas contidas na mensagem DEC recebida foram completamente instaladas. Uma mensagem RPT também pode ser enviada de forma não-solicitada, com o objetivo de contabilização ou de notificação de mudança de status do PEP.

O PEP e o PDP guardam todas as informações de estado relativas aos diversos eventos em bases de dados de políticas, local e central, que o COPS busca manter sincronizadas. A RFC 3084 (CHAN et al., 2001) introduz o conceito de *Policy Information Base* (PIB), uma estrutura de dados que mantém instâncias de especificações de políticas que são transportadas pelo COPS-PR. O PIB é baseado no modelo de *Structure of Management Information* (SMI) e *Management Information Bases* (MIBs), o mesmo usado pelo *Simple Network Management Protocol* (SNMP).

O PIB é uma estrutura de espaço de nome em árvore, onde os galhos representam classes de provisionamento *Provisioning Classes* (PRCs) e as folhas representam instâncias das regras de provisionamento *Provisioning Instances* (PRIs) de uma dada classe (PRC). Cada PRI é identificado por um *Provisioning Instance Identifier* (PRID), que funciona como um ponteiro da estrutura. O espaço de nomes do PIB é comum ao PEP e ao PDP e as instâncias de dados dentro desse espaço são únicas dentro do escopo de um dado par (*Client-Type, Request-State*) por conexão TCP entre um PEP e um PDP.

A próxima seção trata da arquitetura de um servidor de políticas (PDP) proposta pelo *QBone Signaling Design Team* da Internet2.

2.4 *BANDWIDTH BROKER* (BB)

O *Bandwidth Broker* (BB) tem a função de automatizar a negociação de SLS entre domínios, a tomada de decisões de controle de admissão de requisições de clientes, o gerenciamento de recursos e a configuração dos elementos de rede, de acordo com o conjunto de políticas de provisionamento de serviços na rede (CHIMENTO et al., 2002). A idéia de um BB foi sugerida após se identificar que seria impraticável e ineficiente que usuários individuais tivessem conhecimento da topologia da rede e de suas políticas para marcar os pacotes corretamente (JACOBSON, V.; NICHOLS, K.; ZHANG, 1999).

O BB é uma entidade lógica e sua implementação não é objeto de recomendação pelo IEFT, e sim pelo *QBone Signaling Design Team* da Internet2 (CHIMENTO et al., 2002). No entanto, o papel do BB corresponde ao do PDP na arquitetura proposta pelo IEEE RAP WG.

Um dos objetivos do *DiffServ* é simplificar a informação que o roteador deve manter, já que não é preciso manter estados para cada fluxo individualmente. No entanto, o BB permite que cada fluxo seja monitorado a nível administrativo. O BB gerencia os recursos de rede dentro de um domínio *DiffServ* baseado nos SLSs acordados.

O BB é também responsável por se comunicar com BBs em domínios adjacentes para negociar o SLS para fluxos de tráfego em trânsito entre domínios. A maior parte da configuração da rede envolve os roteadores de borda que se interligam aos domínios adjacentes. Inicialmente foi proposto um único BB por domínio, mas múltiplos BBs podem ser considerados por questões de confiabilidade e escabilidade.

O BB monitora e armazena o estado dos recursos dentro do seu domínio e nos roteadores de borda conectados a domínios adjacentes. A informação do estado dos recursos da rede e o conhecimento das políticas de provisionamento de serviço permitem que o BB processe as solicitações de recursos de clientes dentro do seu domínio, levando em conta a capacidade da rede como um todo em prover a QoS requisitada fim-a-fim (CHIMENTO et al., 2002). Todas as solicitações de alocação de recursos devem ser feitas por um cliente ao BB dentro do seu domínio. É função do BB determinar se a solicitação pode ser aceita e se BBs em domínios vizinhos devem ser contactados. Após processar a solicitação, o BB deve notificar ao cliente o fato da solicitação ter sido atendida ou rejeitada.

Uma requisição de um cliente a um BB solicitando o uso de recursos é chamada de *Resource Allocation Request* (RAR). Um RAR pode conter parâmetros como banda requerida, duração do fluxo pretendido e destino do fluxo. Um RAR deve ser conforme a um SLS pré-negociado ao qual o cliente é subscrito. Vários RARs podem ser mapeados a um único SLS desde que o conjunto de solicitações dos diferentes RARs não exceda as condições do SLS. Um RAR não conforme a SLS é rejeitado. Se o RAR é admitido, os pontos de reforço de políticas de QoS (roteadores, com função equivalente à de PEPs) devem ser reconfigurados para admitir o novo tráfego, de acordo com o SLS.

Um *Resource Allocation Answer* (RAA) é uma resposta retornada ao hospedeiro que enviou a solicitação (RAR) para confirmar se o recurso foi alocado e os elementos de rede foram reconfigurados com sucesso de acordo com o SLS vigente. É importante notar que um RAR só pode solicitar recursos conforme o SLS ao qual está associado. Portanto o RAR deve incluir o identificador do SLS associado ao solicitar recursos.

2.4.1 Implementação do Conceito de Bandwidth Broker

A Figura 2-11 ilustra a arquitetura do *Bandwidth Broker*, proposta pelo *QBone Signaling Design Team* da Internet2 (CHIMENTO et al., 2002).

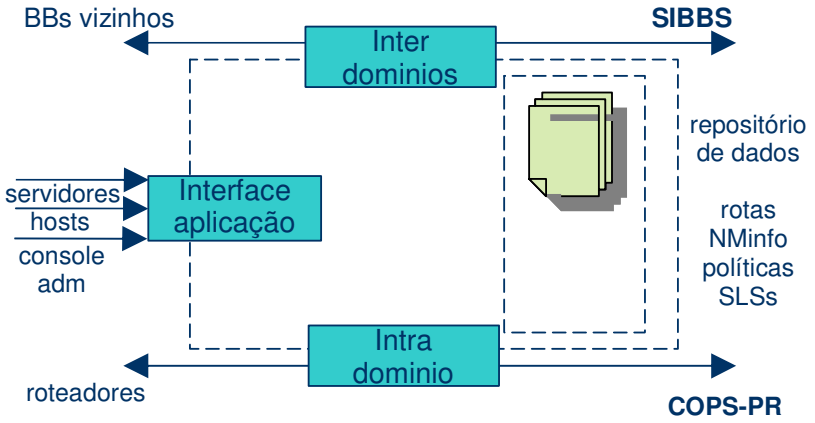


Figura 2-11: Arquitetura do *Bandwidth Broker* proposta pelo QBone (LAGE, 2004)

Os principais componentes propostos são: o protocolo usado para comunicação intra-domínio com os roteadores (COPS-PR); o protocolo para comunicação inter-domínios entre BBs em domínios adjacentes (SIBBS – *Simple Inter-Domain Bandwidth Broker Protocol*); o repositório de dados que contém informações de topologia da rede, disponibilidade/alocação de recursos, SLA/SLSs e especificações técnicas das políticas vigentes; e as interfaces com o usuário administrador e com as aplicações.

2.4.2 Protocolo Intra-Domínio

Diversos métodos de comunicação podem ser usados para comunicar as decisões do BB aos roteadores na forma de parâmetros de configuração dos mecanismos que implementam políticas de serviço. Estes métodos incluem uso dos protocolos COPS, SNMP ou Telnet para configuração por linha de comando. É importante notar que nem todos os roteadores suportam os mesmos métodos de configuração remota.

Em caso de adoção do COPS, o BB é configurado como PDP, aonde são tomadas decisões de controle de admissão de solicitações de recursos levando-se em consideração as políticas existentes. Os roteadores de borda implementam os PEPs, pois eles são configurados

para que as decisões baseadas em políticas sejam implementadas. Os roteadores de borda (PEPs) também consultam o BB (PDP) para obter especificações de políticas próprias (ex. configuração de listas de acesso) no momento de sua inicialização.

Para a operação do BB dentro da arquitetura *DiffServ*, o modelo de provisionamento do COPS-PR, descrito anteriormente, é mais adequado como protocolo intra-domínio e é adotado na implementação de BB usada no escopo dessa dissertação.

2.4.3. Protocolo Inter-Domínios

Grande parte do tráfego na Internet envolve um hospedeiro e um destino que não estão no mesmo domínio. Esta interconectividade, que permite que qualquer pessoa interaja com qualquer outra pessoa ligada à Internet é a razão do seu sucesso. Portanto, a conectividade entre domínios *DiffServ* precisa ser considerada para o provimento de QoS fim-a-fim.

Domínios *DiffServ* adjacentes são conectados por enlaces entre roteadores DS de borda em cada domínio. Como única entidade responsável pelo controle de recursos de QoS em um domínio, o BB tem a função de configurar estes roteadores de borda para permitir o trânsito de fluxos de tráfego identificados. O BB deve definir o tratamento oferecido ao tráfego em trânsito, inclusive remarcação dos pacotes originários de outro domínio, se necessário. Torna-se necessário um mecanismo para permitir que o BB tenha conhecimento de fluxos de tráfego originários de outros domínios e também dar ao BB a habilidade de requerer banda de domínios *DiffServ* adjacentes.

Não existe um protocolo padronizado para este fim. O *Simple Inter-Domain Bandwidth Broker Protocol* (SIBBS) é uma proposta do *QBone Signaling Design Team* da Internet2 (CHIMENTO et al., 2002). Como o nome sugere, o SIBBS é um protocolo simples baseado em modelo cliente-servidor para comunicação entre BBs em domínios *DiffServ* adjacentes, com o objetivo de permitir a negociação de SLS e reserva de recursos para tráfego inter-domínios. Para sua operação, o SIBSS assume que os termos dos SLSs bilaterais estabelecidos sejam conhecidos ou propagados *out-of-band*, de modo que qualquer par de BBs adjacentes tem um completo entendimento do SLS que existe entre eles (CHIMENTO et al., 2002). O SIBBS faz uso de um modelo de tunelamento, no qual túneis são pré-estabelecidos entre cada par de domínios de origem e de destino possíveis (MANTAR et al., 2004). Os túneis, identificados pelo prefixo IP do domínio de destino e valor do DSCP, transportam o tráfego multiplexado de fluxos de cada classe e solicitações de reserva de recursos. As mensagens de sinalização e status nos roteadores de borda são tratadas do ponto

de vista de túneis entre BBs, reduzindo significativamente a sinalização e o controle de admissão, se comparados a esquemas de reservas por fluxo. Apesar de eficiente para pequenas redes, como VPNs corporativas, o SIBBS tem sérios problemas de escalabilidade na Internet. Então foi proposta como melhoria a agregação de destinos. O BB agrega todas as requisições de túneis para uma mesma região de destino e classe de QoS geradas por domínios *upstream* em um único túnel para o domínio *downstream*. O protocolo modificado é chamado eSIBBS (*enhanced SIBBS*) (MANTAR et al., 2006).

2.4.4. Interface de Dados

A interface de dados mantém toda a informação necessária para que o BB execute as suas funções. Contratos de serviço, políticas e especificações técnicas associadas (SLAs/SLSs), topologia da rede, alocação atual dos recursos de rede e estatísticas de gerenciamento de rede são itens que fazem parte do repositório de dados do BB. Quando o BB recebe um RAR de um cliente, ele checa seu repositório de dados para determinar se o novo fluxo pode ser aceito.

A implementação pode ser feita, por exemplo, com *Light-Weight Directory Access Protocol* (LDAP), um padrão de protocolo aberto para serviço de acesso a informações, de natureza simples e sem grandes exigências de recursos. No entanto, o LDAP não suporta políticas complexas devido à estrutura de diretórios em forma de árvore que usa para armazenar informações. Outra opção, adotada na implementação usada dentro do escopo desta dissertação, faz uso de Banco de Dados Relacional *Relational DataBase Management System* (RDBMS) e da linguagem *Structured Query Language* (SQL). O uso de RDBMS permite que elementos de políticas complexas sejam armazenados e que as tabelas sejam facilmente modificadas para incluir novos elementos via SQL. O fato do SQL ser uma linguagem padrão abre uma variedade de opções em sistemas de banco de dados para o BB.

2.4.5 Interfaces com o Usuário e com as Aplicações

Uma interface *web* acessível por qualquer navegador deve permitir ao usuário administrador interagir com o BB, fazer requisições, ver respostas ou simplesmente consultar requisições em uso ou detalhes de SLSs vigentes.

Além disto, para implementar o gerenciamento dinâmico de políticas de QoS, o BB deve ser capaz de receber dinamicamente as requisições de alocação de recursos (RARs) das

aplicações que solicitam QoS da rede. A implantação do BB, como parte integrante da arquitetura de suporte de QoS proposta no escopo desta dissertação faz uso de Serviços Web (WS – *Web Services*) [W3C] – componentes independentes disponíveis na web, como esquemas *eXtensible Markup Language* (XML) e do protocolo *Simple Object Access Protocol* (SOAP). Na implementação de BB adotada, esquemas XML são usados na especificação de requisições de alocação de recursos (RARs), contendo dados como parâmetros de QoS requeridos pelas aplicações e identificador de contrato de serviço (SLA). As vantagens do uso de esquemas XML como alternativa de definição de mensagens é o suporte a vários tipos de dados e a extensíbilidade, permitindo a criação de novos tipos de dados ou tipos derivados, além de possibilidade de, por exemplo, conversão e validação usando código JavaScript,. O SOAP, um protocolo simples e leve, baseado em XML, é usado na troca de informações estruturadas entre aplicações de rede num ambiente distribuído e descentralizado, de forma independente da plataforma de aplicação. Na implementação de BB adotada, o protocolo SOAP é usado para a propagação dos esquemas XML entre os elementos do plano de controle das aplicações de Videoconferência e Vídeo sob Demanda e o *Bandwidth Broker*.

CAPÍTULO 3 PROJETO INFRAVIDA E APLICAÇÕES DE TELEMEDICINA SOAP

3.1 APRESENTAÇÃO DO PROJETO INFRAVIDA

Como mencionado, o presente trabalho desenvolveu-se como parte integrante do projeto InfraVIDA.[1] O projeto **InfraVIDA — Infra-Estrutura de Vídeo Digital para Aplicações de Tele-Saúde** — visou o desenvolvimento de um sistema de telemedicina, incluindo aplicações de telediagnóstico e segunda opinião médica e um sistema de educação continuada à distância para profissionais de saúde baseados na Internet.

Os conceitos de tele-saúde e telemedicina são muito próximos. Tele-saúde é a promoção de saúde, relacionada a serviços de informação, através de tecnologias de telecomunicações. Telemedicina pode ser definida como o conjunto de tecnologias e aplicações que permitem a realização de ações médicas à distância. Como vantagens do uso da telemedicina, temos: redução do tempo e dos custos em transportar pacientes; ajuste do gerenciamento dos recursos de saúde devido à avaliação e triagem por especialistas; acesso rápido a especialistas em casos de acidentes e emergências; diminuição da pressão sobre hospitais já comprometidos pela falta de leitos e recursos; uso mais eficiente de recursos, através da centralização de especialistas e da descentralização da assistência, alcançando um número maior de pessoas; cooperação e integração de pesquisadores com o compartilhamento de registros clínicos e maior qualidade dos programas educacionais para médicos e residentes localizados em zonas fora de centros especializados.

O projeto InfraVIDA foi desenvolvido em parceria entre Universidade Federal de Pernambuco (UFPE), Universidade Federal da Bahia (UFBA), Universidade Federal do Rio Grande do Norte (UFRN), (Universidade Federal da Paraíba) e Universidade Salvador (UNIFACS), envolvendo também o Real Hospital Português (RHP) de Recife, a Faculdade de Medicina (FAMED) da UFBA e o Hospital das Clínicas da UFPE, com o apoio do Conselho Nacional do Desenvolvimento Científico e Tecnológico / Rede Nacional de Pesquisa

[1] A maior parte do conteúdo deste capítulo foi apresentado no Workshop InfraVIDA no SBRC 2004 [LAGE, 2004].

(CNPq/RNP), tendo sido aprovado no Edital de Redes Avançadas CNPq 10/2001 — ProTeM/RNP 01/2001.

No contexto do projeto InfraVIDA, integrou-se o *Open H.323* (RABELO et al., 2001), uma implementação de código aberto do padrão H.323 com suporte a videoconferência em enlaces de baixa velocidade, o sistema *DynaVideo* (LEITE, L.; SOUZA FILHO, G.; BATISTA, T., 2001) de distribuição de vídeo digital, o sistema *HealthNet* (BARBOSA, 2001) de suporte ao trabalho cooperativo de médicos e o sistema Ambiente Brasileiro de Aprendizagem (ABRA, 2004), um portal de educação à distância, adotando-se uma arquitetura de QoS para a oferta de serviços diferenciados em redes IP (*DiffServ*).

3.2 ARQUITETURA DO PROJETO INFRAVIDA

Um sistema de suporte a telemedicina inclui funções colaborativas específicas fazendo uso de ferramentas básicas de colaboração como vídeo e áudio conferência, serviços de diretório, *whiteboards* e *chats*. O sistema InfraVIDA visa a dar suporte aos serviços de Telediagnóstico e Segunda Opinião Médica através de um sistema de conferência multimídia envolvendo a distribuição de fluxos de vídeo de diferentes qualidades (LAGE, 2004).

O *HealthNet* é um sistema de telemedicina de apoio à realização de Segunda Opinião Médica na forma de telediagnóstico ou de cooperação, possibilitando a troca de informações de saúde de pacientes entre serviços e profissionais de diferentes localidades. O telediagnóstico é um serviço assíncrono (baseado em acesso a bases de dados e troca de e-mails), enquanto que a cooperação é em serviço que pode ser assíncrono (baseado em fórum de discussão) ou primordialmente síncrono (baseado em Videoconferência). O sistema *HealthNet* utiliza os serviços de Controle de Acesso, Qualidade de Serviço, Videoconferência, Imagem e Vídeo sob Demanda e Anotações, disponibilizados pelo projeto InfraVIDA:

- O serviço de Videoconferência, utilizado para visualização de imagens ou vídeo onde há necessidade de uma sincronização dos elementos utilizados entre os agentes de saúde, ou seja, na modalidade Cooperação síncrona.

- O serviço de Imagem/Vídeo sob Demanda, utilizado para visualização de imagens ou vídeo onde não há a necessidade de uma sincronização entre agentes de saúde, seja para telediagnóstico ou cooperação assíncrona. O serviço de Vídeo sob Demanda é também utilizado para visualização de imagens e vídeos durante sessões de Cooperação síncrona.

- Os mecanismos de Qualidade de Serviço aplicados às sessões de Videoconferência e em todo tipo de visualização de Imagem ou Vídeo sob Demanda, de forma a minimizar perdas ou atrasos na sua exibição, para que as mesmas não sofram pausas indevidas nem percam a qualidade de resolução exigida.

- Os mecanismos de Controle de Acesso, utilizados na validação de usuários, na disponibilização de funcionalidades adequadas para os mesmos e na restrição de visualização por parte dos agentes de saúde de informações consideradas eticamente protegidas (ex.: identificação de pacientes, informações de diagnóstico em determinados casos clínicos, etc.).

- O serviço de Anotações em Vídeo, utilizado para inserir comentários em vídeos ou imagens médicas utilizadas em uma Segunda Opinião ou telediagnóstico.

Os documentos relevantes anexados à cooperação, como casos clínicos e todos os seus anexos, inclusive imagens e vídeos de eventos clínicos, possíveis anotações e outros documentos anexados de outras fontes são armazenados juntamente com o registro da cooperação ao prontuário do paciente, tornando-se assim posteriormente acessível por outros médicos ou estudantes interessados em estudar o caso.

No escopo do projeto InfraVIDA foram desenvolvidos elementos da solução nas áreas de redes, sistemas distribuídos/*middleware*, bancos de dados/multimídia e engenharia de *software* em suporte às aplicações de telemedicina e educação à distância.

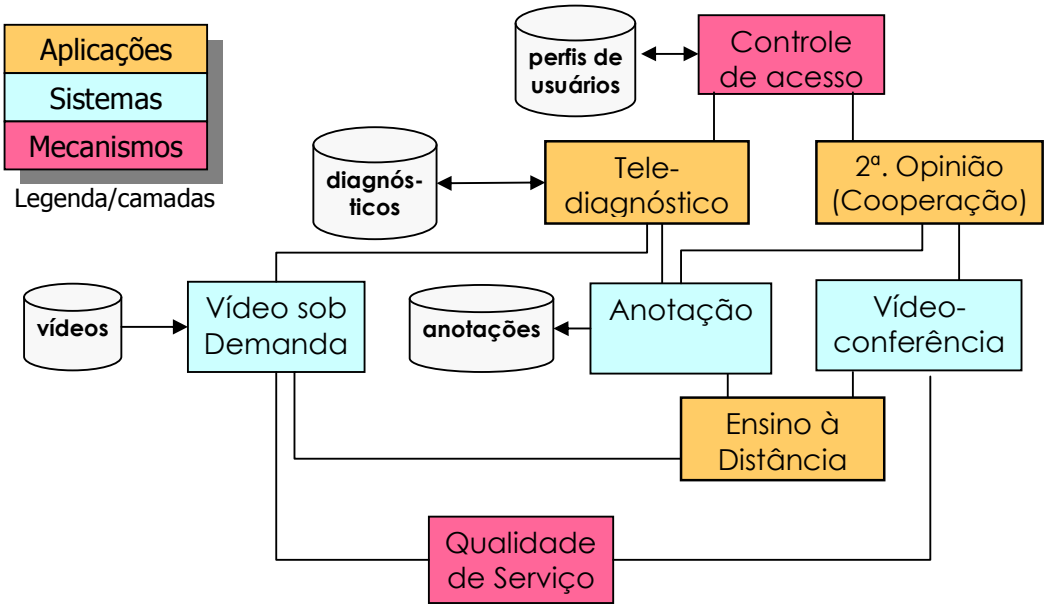


Figura 3-1: Arquitetura do Projeto InfraVIDA (FERRAZ, 2001)

A arquitetura da solução proposta, incluindo as aplicações, os sistemas e os mecanismos de apoio, pode ser visualizada na Figura 3-1.

O sistema *HealthNet* do projeto InfraVIDA foi desenvolvido segundo uma arquitetura orientada a serviços *Service Oriented Architecture* (SOA) (BARRY, 2003), que é uma abordagem para computação distribuída onde recursos de *software* são vistos como serviços disponíveis na rede ou uma coleção de serviços que se comunicam. Estes componentes podem ser publicados, invocados e descobertos por qualquer outro. A vantagem desta abordagem é a facilidade de composição, já que os componentes são auto-contidos e modulares, com pouca dependência entre módulos. Os módulos têm interfaces endereçadas por rede e podem ser descobertos e ligados dinamicamente independentemente da localização. Isto facilita a interoperabilidade e a auto-recuperação.

O piloto *HealthNet* faz uso de WS – componentes programáveis que podem ser publicados, descobertos e invocados numa rede IP, usando protocolos abertos e interoperáveis, como *Java APIs for XML-Based Remote Procedure Call* (JAX-RPC) e SOAP, como metodologia de desenvolvimento de transações e dos mecanismos de segurança da solução.

3.3 QUALIDADE DE SERVIÇO, ONDE SE APLICA?

O Projeto InfraVIDA prevê a utilização de *middleware* multimídia, em tempo real, com QoS. Os mecanismos de QoS em redes IP que são aqui propostos pretendem priorizar o tráfego dos sistemas multimídia - Videoconferência e Imagem/Vídeo sob Demanda utilizados pela aplicação de Segunda Opinião Médica, em modalidade de cooperação síncrona. A intenção é de que o tráfego destas aplicações em tempo real receba tratamento diferenciado pelas redes que o suportam, de forma que o tráfego concorrente de dados dos demais sistemas componentes da solução InfraVIDA e de outras aplicações que façam uso concorrente das mesmas redes não comprometam o tempo de resposta e a qualidade das imagens apresentadas durante a execução da sessão de cooperação síncrona, já que são fatores críticos para a consecução da Segunda Opinião Médica.

O *HealthNet* prevê três perfis distintos para seus usuários: o ‘Solicitante’, qualquer profissional de saúde que requer um telediagnóstico, o ‘Consultor’, que é o especialista responsável pelo parecer médico numa Segunda Opinião Médica e o ‘Colaborador’, que pode ser outro médico convidado a participar da Segunda Opinião Médica. Solicitantes, consultores e colaboradores podem manter uma comunicação assíncrona via *e-mail* com fins de telediagnóstico, mas o projeto provê o agendamento das sessões de cooperação síncrona,

que incluem a ativação de sistemas de Videoconferência e de distribuição de Imagem/Vídeo sob Demanda associadas a uma sessão específica de Segunda Opinião, haja ou não anotação. Ou seja, as sessões de Videoconferência e Imagem/Vídeo sob Demanda ocorrem simultaneamente.

A sessão de cooperação síncrona pré-agendada será ativada pelo 'Consultor' e o serviço de Videoconferência deve permitir deve que novos usuários sejam incluídos numa sessão em curso. O sistema de Vídeo sob Demanda deve disponibilizar imagens e vídeos nos servidores mais próximos aos clientes nos horários e localizações em que serão solicitados (por agendamento) para aperfeiçoar o uso dos recursos da rede e agilizar o serviço como um todo.

As sessões seguintes descrevem a arquitetura e os mecanismos dos sistemas de Videoconferência e de Imagem/Vídeo sob Demanda, os requisitos das aplicações e a integração das soluções no provimento e gerenciamento dinâmico de QoS para as aplicações de telemedicina do InfraVIDA.

3.4 SISTEMA DE IMAGEM E VÍDEO SOB DEMANDA

A arquitetura proposta pelo Grupo de Trabalho de Vídeo Digital da Rede Nacional de Pesquisa (GTVD-RNP) para o serviço de distribuição de Vídeo sob Demanda baseia-se em uma rede de vídeo digital – *DynaVideo*, onde os elementos ativos são servidores de vídeo (ELIAS et al., 2004). A arquitetura suporta o serviço de Vídeo sob Demanda (VoD – *Video on Demand*) de transmissão de vídeo em tempo real em modo *streaming*, onde uma parte inicial do conteúdo do vídeo é retida em um *buffer* e o vídeo é então apresentado à medida que é transferido à máquina do cliente. Suporta ainda o serviço de Imagem sob Demanda (IoD – *Image on Demand*) em modo *download*. A arquitetura provê quatro funcionalidades – aplicação, coordenação, acesso e armazenagem – e prevê desacoplagem entre funcionalidades para garantir flexibilidade no atendimento aos serviços de distribuição de vídeo e imagem.

A Figura 3-2 ilustra a arquitetura da rede *DynaVideo* como proposta pelo GTVD-RNP.

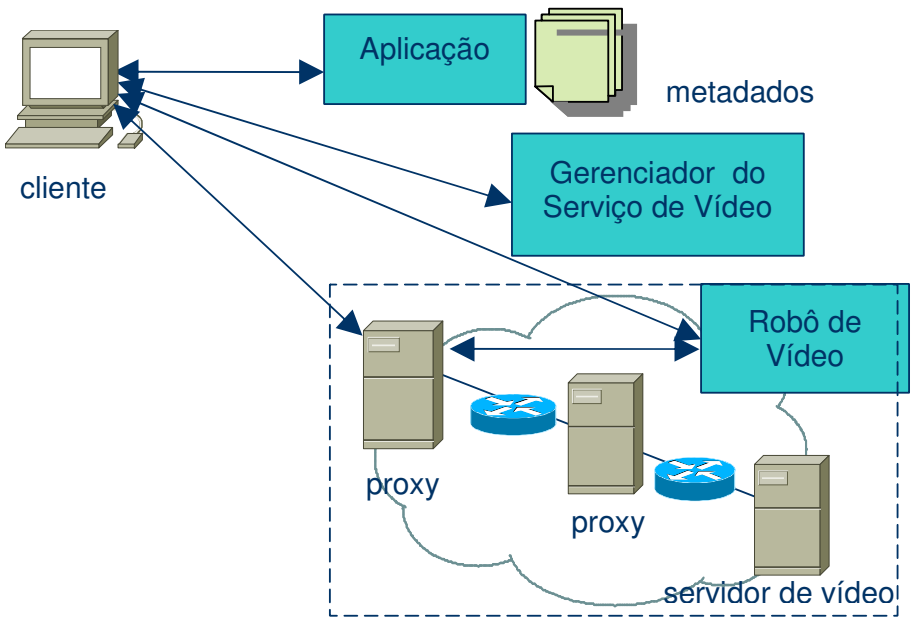


Figura 3-2: Arquitetura da rede de vídeo digital *(DynaVideo)* (LAGE, 2004)

A funcionalidade de armazenamento é implementada por servidores de vídeo que são repositórios permanentes de imagem e vídeo digital. A arquitetura prevê um esquema de replicação parcial, onde réplicas de vídeo podem existir simultaneamente em alguns servidores.

A funcionalidade de acesso é implementada por *proxies*. Um *proxy* implementa um *cache* para aqueles vídeos de maior audiência em sua área de atendimento, definida pela proximidade dos usuários. A arquitetura permite a conexão em cascata de *proxies*, configurando uma hierarquia de múltiplos níveis de *cache*. A principal vantagem desta abordagem é a distribuição otimizada do tráfego na rede da RNP, nas redes regionais e nas redes institucionais.

O Gerenciador do Serviço de Vídeo implementa a funcionalidade de coordenação. A sua função é a de determinação da melhor rota de distribuição entre um cliente e um servidor fonte de um determinado vídeo ou imagem.

A funcionalidade de aplicação na arquitetura da rede de vídeo digital *DynaVideo* é implementada por um robô, que de posse do identificador de vídeo e da melhor rota de distribuição, inicia a carga de uma cópia do arquivo no servidor de acesso (*proxy* mais próximo ao cliente).

A seqüência de ações executada quando um cliente acessa o serviço é descrita a seguir. Inicialmente o cliente acessa via Internet uma aplicação de busca, que consulta um repositório de metadados dos vídeos (informações sobre os formatos e tamanhos dos vídeos, além dos

endereços dos servidores fonte) disponibilizados pela instituição responsável. Isto torna o serviço proposto independente de um servidor de vídeo específico e de uma aplicação de busca específica. A aplicação de busca retorna os seguintes elementos: o identificador do vídeo desejado, os endereços dos servidores fonte do vídeo, e o endereço do elemento Gerenciador do Serviço de Vídeo.

Uma vez recuperado o identificador do vídeo, o cliente então envia uma requisição ao Gerenciador do Serviço de Vídeo, informando o identificador do vídeo desejado e os endereços dos servidores fonte daquele vídeo. O Gerenciador do Serviço de Vídeo é responsável por selecionar um dos servidores disponíveis, juntamente com a cadeia de *proxies*, que serão utilizados para atender a solicitação do cliente. Como resultado, o gerenciador do serviço de vídeo retorna ao cliente uma URL com a melhor rota de distribuição (servidor e *proxies*) a ser utilizada para distribuir o vídeo até aquele cliente e um *hiperlink* para o servidor de acesso mais próximo (o primeiro *proxy* na rota de distribuição).

Quando o *hiperlink* retorna ao cliente, isto faz com que o aplicativo de reprodução de vídeo seja automaticamente instanciado e dispare o robô que solicita o vídeo ao servidor de acesso, que por sua vez solicita o vídeo ao próximo *proxy*, e assim sucessivamente, até que a requisição chegue ao servidor fonte, responsável por armazenar uma cópia persistente do vídeo. Neste momento a distribuição é iniciada e o cliente começa a receber o fluxo de vídeo. O servidor de acesso é inercial, continua a receber e a armazenar o *streaming* de vídeo mesmo que o cliente desista e interrompa o aplicativo de reprodução de vídeo. Quando o vídeo atravessa a cadeia de servidores vai sendo armazenado em *cache*, de modo que nas próximas requisições ele estará mais próximo do local onde está o cliente. Alternativamente, quando parte do vídeo encontra-se armazenado na *cache* de algum *proxy* intermediário, o cliente começa a receber o fluxo antes que a requisição atinja o servidor fonte. Caso o vídeo esteja totalmente armazenado na *cache* de algum *proxy* intermediário, a requisição se propaga apenas até este *proxy*.

O serviço de gerenciamento de vídeo deve possuir a noção de localização dos diversos servidores fonte e *proxies*, de modo que uma rota de distribuição otimizada possa ser estabelecida. Para isto é usado um algoritmo de determinação do melhor caminho entre dois pontos, cliente e servidor. A solução implementada para a definição da rota de *proxies* tem como base a distribuição dos blocos de endereços na RNP (bloco ‘vs’ localização) representada em um grafo de conectividade. Para criar este grafo, a equipe do GTVD especificou um esquema XML capaz de representar nós da rede, conexões entre estes nós, associação de blocos aos nós, e distribuição de servidores e *proxies* nestes diversos nós. A

partir da especificação da rede em XML, o serviço de gerenciamento gera um grafo e aplica algoritmos de determinação de melhor caminho para calcular a melhor rota de distribuição entre um cliente e um servidor fonte de um determinado vídeo desejado pelo usuário (SILVA, O.; ELIAS, G.; LEMOS, G., 2004).

O capítulo 4 contém as contribuições dessa dissertação para uma Arquitetura de Suporte de Gerência de QoS para Aplicações de Telemedicina, que incluem considerações em relação à arquitetura do serviço de vídeo específicas para o projeto InfraVIDA e uma especificação dos parâmetros de Qualidade de Serviço associados aos serviços de IoD/VoD. Contém ainda uma proposta de integração dos serviços de IoD/VoD e QoS que inclui a especificação de requisição de alocação de recursos (RAR), de modo que, a cada imagem ou vídeo solicitado, o Gerenciador do Serviço de Vídeo comunique ao Gerenciador de Recursos de Rede ou Servidor de Políticas de QoS (BB) os parâmetros de QoS associados a um identificador de contrato de serviço (SLA).

3.5 SISTEMA DE VIDEOCONFERÊNCIA

O sistema de Videoconferência do InfraVIDA é baseado no *OpenH323* (RABELO et al., 2001), uma implementação em código aberto do H.323. O H.323 é uma especificação do ITU-T para viabilizar aplicações de videoconferência de baixa velocidade em micro-computadores, via transmissão de áudio, vídeo e dados através de redes IP. O padrão H.323 aborda sinalização e controle de chamadas, transporte e controle de tráfego multimídia e controle de banda para conferências ponto-a-ponto e multiponto [H.323, 2006].

Os diversos componentes e protocolos que compõem a arquitetura H.323 incluem:

- sinalização, registro e admissão de chamadas (H.225);
- controle de canais de mídia (H.245);
- *codecs* de áudio (G.711, G.722, G.723, G.728, G.729);
- *codecs* de vídeo (H.261, H.263);
- compartilhamento de dados (T.120);
- transporte de mídia na rede IP (RTP/RTCP - Real-Time Transport Protocol/Real-Time Control Protocol).

Um sistema H.323 é composto de diversos elementos: terminais, *gateways*, *gatekeepers* e *multipoint control units* (MCUs). Os terminais ou clientes implementam a funcionalidade de conferência de áudio, vídeo e, opcionalmente, compartilhamento de dados.

Os *gateways* (GWs) interconectam a rede IP a uma rede telefônica, por exemplo, para o serviço de telefonia IP. O *gatekeeper* (GK) é o elemento do plano de controle que faz o controle de admissão e tradução de endereços para os participantes de uma conferência e GWs. Os MCU são equipamentos de distribuição de áudio e vídeo que implementam facilidade de conferência multiponto entre diversos terminais ou GWs e mantém o controle de mídia com os terminais participantes. A funcionalidade de GK pode ser implementada no MCU.

A Figura 3-3 ilustra os elementos da arquitetura H.323 de interesse para o serviço de Videoconferência do InfraVIDA. São premissas da implantação *Open H.323* adotada, o uso inicialmente de um único GK e de uma única MCU, além da ausência de GWs para expansão do serviço usando a rede de telefonia tradicional. Uma arquitetura com múltiplos MCUs pode vir a favorecer a otimização do uso dos recursos da rede.

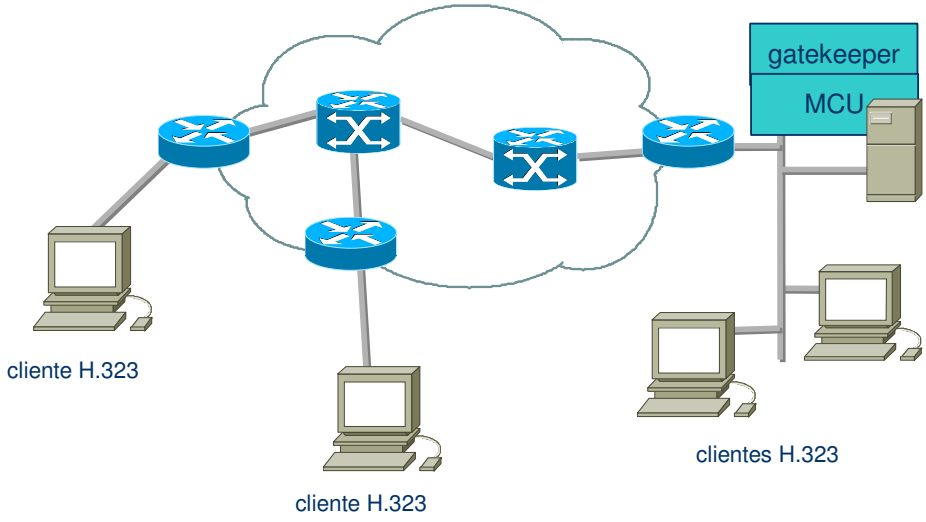


Figura 3-3: Arquitetura do serviço de Videoconferência *(Open H.323)* (LAGE, 2004)

A rede IP pode transportar áudio e vídeo em modo *unicast* ou *multicast*, desde que suportado pelos roteadores e estações de clientes. No H.323, sessões paralelas de áudio e vídeo ocorrem nas duas direções (*full-duplex*) e são transportadas via UDP e RTP/RTCP entre clientes ponto-a-ponto. O RTP garante o sincronismo das sessões de áudio e vídeo e permite a identificação do tipo de mídia, enquanto o RTCP permite a verificação de parâmetros de atraso e variação de atraso via selo de tempo (*timestamping*).

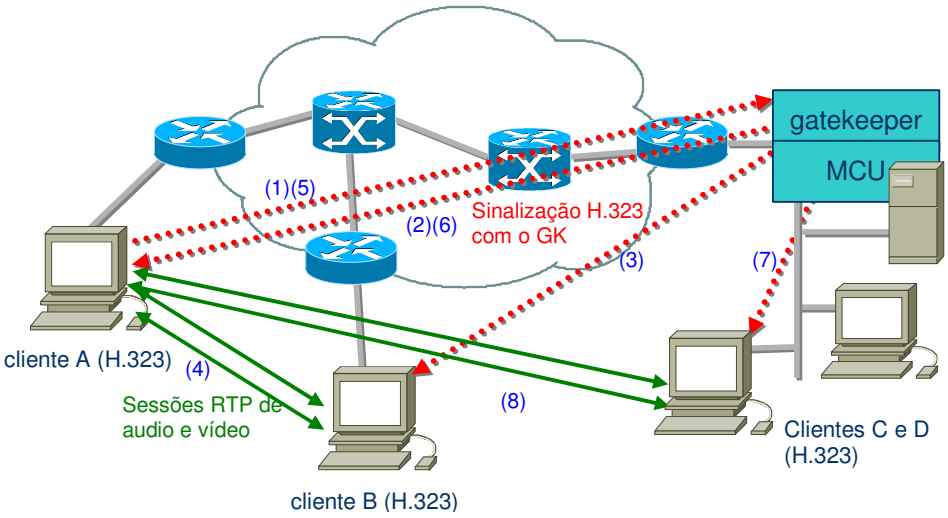


Figura 3-4: Estabelecimento de sessões de Videoconferência (A&B, A&C) (LAGE, 2004)

A Figura 3-4 ilustra a sinalização trocada para estabelecimento de sessões de videoconferência e a forma como clientes H.323, MCUs e *gatekeepers* (GKs) interagem. Os clientes H.323 são agrupados em zonas e cada zona é administrada por um *gatekeeper*. Cada cliente H.323 é registrado no GK no momento de inicialização e o GK mantém uma lista de todos os clientes *on line*. Quando um cliente A deseja iniciar uma sessão de videoconferência com o cliente B, contacta o GK da sua zona, que faz o controle de admissão em uma sala de conferência e retorna o endereço IP de B para A. O GK retorna o endereço IP do cliente destino em outra zona, se o GK local é configurado para acessar o GK da zona remota. O cliente A então estabelece a sessão de videoconferência (sessões RTP paralelas de áudio e vídeo) com B. Cada novo cliente que deseja participar de uma sessão de videoconferência pré-existente contacta o GK para controle de admissão na sala de conferência e o GK retorna os endereços IP dos outros participantes da mesma sessão. O novo cliente estabelece sessões RTP de áudio e vídeo com cada um dos demais participantes. A MCU integra a função de controle (GK) e é também um ponto de replicação, mantendo diversas sessões ponto-a-ponto com terminais subordinados a ela, que são clientes finais das sessões de videoconferência, também registrados no GK.

Na arquitetura do serviço de Videoconferência, o *gatekeeper* é o elemento do plano de controle responsável pelo controle de admissão dos participantes em uma dada sessão. Além disto, o *gatekeeper* tem informações de controle de banda alocada a cada sessão.

Essa dissertação contém uma proposta de integração do serviço de Videoconferência com o serviço de gerenciamento de QoS que prevê uma comunicação dinâmica do *Gatekeeper* com o *Bandwidth Broker* a cada nova sessão de videoconferência e a cada cliente adicionado a sessões pré-existentes. A especificação da requisição de alocação de recursos proposta contém os endereços IP dos clientes envolvidos e da MCU (se for o caso), tipo de distribuição (*unicast* ou *multicast*, e neste caso quem é o *root*), identificação da sessão de videoconferência, parâmetros de QoS para as sessões paralelas de áudio e vídeo e um identificador de SLA, como descrito no capítulo seguinte.

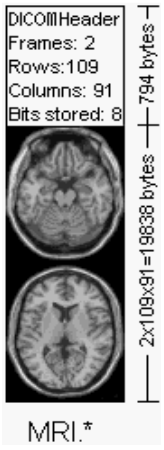
CAPÍTULO 4 ARQUITETURA DE SUPORTE DE GERÊNCIA DE QOS PARA APLICAÇÕES DE TELEMEDICINA

4.1 REQUISITOS DAS APLICAÇÕES DE TELEMEDICINA

Apresentamos a seguir os padrões e as métricas adotadas para as aplicações de Telemedicina do projeto InfraVIDA.

4.1.1 Padrão de Imagens Digitais e Comunicações em Telemedicina

Aplicações médicas requerem alta resolução de imagem e vídeo, com tolerância mínima a perdas no transporte. O padrão *Digital Imaging and Communications in Medicine* (DICOM, 2004), desenvolvido para facilitar a interoperabilidade de equipamentos de imagens médicas, especifica a sintaxe e semântica de comandos de protocolo de transferência de imagens médicas, com suporte a classes de serviço para garantia de qualidade de imagens radiológicas e ultra-sonográficas. Especifica ainda um padrão de formato de arquivos e uma estrutura de diretório médico para armazenamento e recuperação de imagens. A Figura 4-1 mostra aspectos do formato de arquivo DICOM.



O formato de arquivo DICOM contém um *header* com identificação do paciente, tipo de *scan*, dimensões da imagem, etc. A imagem à esquerda é de um arquivo DICOM hipotético. Neste exemplo, 794 *bytes* são usados no *header*, que descreve dimensões da imagem, tipo de *scan*, identificação do paciente, etc. O *header* define uma imagem com 109x91x2 *voxels*, e resolução de 1 *byte* por *voxel* (o tamanho total da imagem é 19838 *bytes*). Os *pixels* da imagem e o *header* são armazenados no mesmo arquivo.

Figura 4-1: Formato de arquivo DICOM (University of Nothingham, 2004)

O protocolo DICOM se refere a uma camada de aplicação, suportada pelas camadas de transporte e interconexão de redes, implementadas pelos protocolos TCP/IP na Internet e na

RNP. O padrão de imagens médicas é definido pelo tipo de equipamento *scanner* utilizado, com imagens nucleares típicas de 256x256 *voxels* de 500 *kbytes* e imagens ultrasonográficas típicas de 512x512 *voxels* de 1 MB, podendo-se chegar a 20 MB por imagem. Tais imagens são digitalizadas usando padrões de codificação de imagem, descritos a seguir.

4.1.2. Padrões de Codificação para Aplicações Multimídia

São diversos os padrões de codificação de imagem, vídeo e áudio para as aplicações multimídia aplicadas à Telemedicina – IoD/VoD e Videoconferência, consideradas no escopo do projeto InfraVIDA.

O padrão *Joint Photographic Experts Group* (JPEG) [JPEG *Committee Home Page*], trata separadamente os componentes de luminância e de crominância da imagem, em macroblocos independentes *(Y, U, V)*, para cada porção de 8 x 8 *pixels* do quadro. Os macroblocos são submetidos ao método de transformadas *Discrete Cosine Tranform* (DCT) para suprimir áreas da imagem de alta freqüência não visíveis ao olho humano, e os coeficientes DCT obtidos são então quantizados e codificados (na ordem da varredura *zig-zag* na matriz de coeficientes DCT), como ilustra a Figura 4-2.

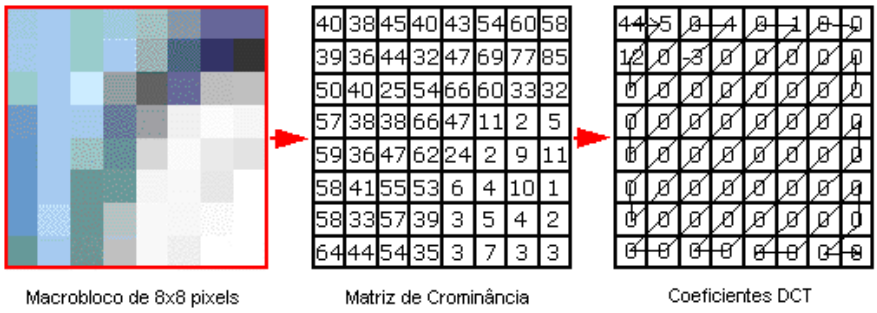


Figura 4-2: Método DCT (Discrete Cosine Transform) (AUTOR, 2008)

O padrão JPEG2000 ou JP2 (CHARRIER, M.; SANTA CRUZ, D.; LARSSON, M. , 1999), uma evolução do JPEG, introduz métodos de transmissão e distribuição progressiva de imagens sem perdas (wavelets), considerando resolução, componentes de crominância e de luminância e sua localização espacial dentro da imagem. Tais métodos são baseados ou em método de codificação DTC progressiva, que usa propriedade de interpolação do DTC para reconstruir progressivamente a qualidade de imagem na recepção, ou em método hierárquico

de codificação em camadas, que permite reduzir a resolução espacial da imagem na recepção, decodificando-se apenas parte dos *bits* recebidos.

Por exemplo, uma imagem original de 512x512 *pixels* comprimida gera um arquivo de 100 *kbytes*. Uma pequena imagem de baixa resolução (32x32 *pixel*) de 10 *kbytes* pode ser enviada e visualizada. O envio de mais 15 *kbytes* aumenta a resolução para 64x64 *pixels*, e assim por diante, até que todos os 100 *kbytes* tenham sido enviados e a imagem original de 512x512 *pixels* seja recuperada no destino.

Além do método de distribuição progressiva de imagens de alta qualidade, o JPEG2000 permite altas taxas de compressão de imagens com baixa perda, o que favorece a transmissão em enlaces de baixa velocidade. O JPEG2000 oferece melhor qualidade de imagem para arquivos do mesmo tamanho ou redução de 25-35% no tamanho de arquivos para qualidade de imagem equivalente. A qualidade de imagem obtida com o JPEG2000 é boa mesmo em altas taxas de compressão, acima de 25:1.

O que é conhecido como vídeo MPEG é, na verdade, um conjunto de especificações referentes ao tratamento de áudio e vídeo: MPEG-1, MPEG-2, MPEG-4, MPEG-7 e MPEG-21 desenvolvidas pelo *Moving Picture Experts Group* (MPEG) [MPEG Home Page] para os organismos de padronização *International Organization for Standardization* (ISSO) e *International Eletrotecnical Commission* (IEC).

O MPEG-1, padrão inicial de compressão de áudio e vídeo, foi desenvolvido para armazenar e distribuir vídeo em qualidade VHS e qualidade de áudio de CD-ROM, e hoje suporta os atuais formatos VCD e MP3. Tipicamente trabalha em resoluções de vídeo de 352x240 *pixels* a 30 quadros/s (padrão NTSC) ou 352x288 *pixels* a 25 quadros/s (padrão PAL), com taxa de transmissão em torno de 1,5 Mbps.

O padrão MPEG-1 usa um esquema de codificação híbrido DCT/DPCM de transformadas DCT (*Discrete Cosine Tranform*) e codificação preditiva do DPCM (*Differential Pulse Code Modulation*), semelhante ao do padrão H.261, que permite correlações tanto espaciais quanto temporais entre quadros, com compensação de movimento e refinamentos em predição para permitir acesso randômico à mídia digital armazenada. O algoritmo MPEG-1, ilustrado na Figura 4-3, codifica um quadro de uma seqüência de vídeo no modo intra-quadro, equivalente à codificação JPEG, gerando um quadro *'I'* e codifica um quadro subseqüente usando predição de compensação de movimento inter-quadros, em relação ao quadro *'I'* anterior, gerando um quadro *'P'*. São também gerados quadros *'B'*, pela interpolação bidirecional da imagem entre os quadros *'P'* e *'I'* anterior e posterior a eles.

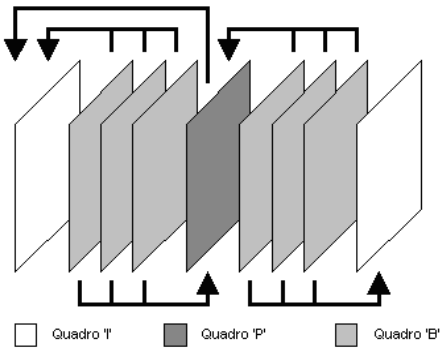


Figura 4-3: Método de codificação de imagem intra e inter-quadros do MPEG-1 (AUTOR, 2008)

O MPEG-2 (padrão equivalente ao H.262 do ITU-T) introduz a codificação de vídeo entrelaçado, suporta formatos de maior resolução e qualidade de imagem, como o DVD, tem maior escalabilidade e se aplica à transmissão de televisão digital (HDTV). O padrão MPEG-2 é descrito por perfis e níveis. O perfil define a escalabilidade da palavra em *bits* e a resolução do espaço de cores, enquanto o nível define a resolução da imagem e a máxima taxa de transmissão por perfil. Os descritores para o perfil principal são: nível baixo (resolução de vídeo de 352x288 *pixels* a 30 quadros/s a 4 Mbps), nível principal (resolução de vídeo de 720x576 *pixels* a 30 quadros/s a 15 Mbps), nível alto (resolução de vídeo de 1440x1152 *pixels* a 60 quadros/s a 60 Mbps), e nível muito alto (resolução de vídeo de 1920x1152 *pixels* a 60 quadros/s a 80 Mbps) (SIKORA, 1999).

O MPEG-4 é um padrão voltado a conteúdo multimídia interativo e comunicações móveis, permite a manipulação de 'objetos de mídia' em composições audiovisuais. Suas principais características são: padroniza unidades de representação de conteúdo audiovisual, chamadas 'objetos de mídia', descreve a composição desses objetos para criar objetos de mídia compostos que formam cenas audiovisuais; multiplexa e sincroniza os dados associados com objetos de mídia, de maneira que possam ser transportados por canais de rede que ofereçam QoS apropriada à natureza de objetos de mídia específicos; e permite interação com o cenário audiovisual gerado na recepção. O MPEG-4 tem duas versões, MPEG-4 Parte 2 e MPEG-4 Parte 10. O MPEG-4 Parte 2, no perfil simples (SP – *Simple Profile*), suporta a transmissão de conteúdo multimídia em redes de relativamente baixa capacidade (10 kbps a 1 Mbps), sendo freqüentemente usado na Internet, por exemplo, no formato *Quicktime*. O perfil avançado (ASP – *Advanced Simple Profile*) suporta codificação de vídeo entrelaçado, usa quadros *'B'* de interpolação bidirecional e compensação de movimento global e está na base do formato DVX.

O mais recente padrão MPEG-4 Parte 10 ou MPEG AVC (*Advanced Video Coding*) (equivalente ao H.264 do ITU-T) usa mecanismos mais sofisticados de predição e compensação de movimento, com uso de blocos de tamanho variável entre 4x4 *pixels* a 16x16 *pixels*, e avanços nos métodos de DCT, quantificação, varredura e codificação PCM do vídeo sem perdas e com uso de menor número de *bits*. Suporta diversos perfis: básico (BP- *baseline profile*) originalmente voltado para uso de recursos computacionais limitados, usado em videoconferência e computação móvel; principal (MP – *main profile*), para distribuição e armazenamento de vídeos; estendido (XP – *extended profile*) para *streaming* de vídeo; alto (HiP – *high profile*), para distribuição e aplicações de armazenamento de vídeo em disco, usado nos formatos HD DVD e *Blu-Ray Disc*; e ainda perfis profissionais (Hi10P, Hi422P e Hi444PP), que usam maior número de *bits* para codificação da imagem e métodos avançados de amostragem de cor e de predição e representação de regiões específicas das imagem. Os níveis suportados no perfil básico do MPEG-4 AVC permitem o seu uso, por exemplo, em taxas tão variadas quanto: 768 kbps, com resolução de 352x288 *pixels* a 30 quadros/s (nível 1); 2 Mbps ou 4 Mbps, com resolução de 352x480 *pixels* a 30 quadros/s (nível 2); 10 a 20 Mbps, com resolução de 720x480 *pixels* a 1280x720 *pixels* a 30 quadros/s (nível 3); 20 a 50 Mbps, com resolução de até 1024x2048 *pixels* a 30 quadros/s (nível 4); até 240 Mbps em 4096x2048 *pixels* a 30 quadros/s (nível 5).

O MPEG-7 (ou *Multimedia Content Description Interface*) é um padrão de descrição e busca de conteúdo audiovisual por usuários ou sistemas computadorizados. O MPEG-7 oferece um extenso conjunto de ferramentas de descrição audiovisual (*Description Tools*), que consiste de metadados elementares, sua estrutura e relações entre eles, definidos em esquemas de descrição e descritores, que formam a base de aplicações que oferecerem acesso a conteúdo multimídia, de forma efetiva e eficiente, seja através de busca, filtros ou navegação.

O MPEG-21 define um padrão de compartilhamento de direitos, permissões e restrições de uso de conteúdo digital. Baseado em XML, o MPEG-21 permite que as informações de licença de uso sejam comunicadas entre computadores, de maneira inequívoca e segura.

O H.261, um dos padrões de vídeo integrantes da arquitetura H.323, foi originalmente desenvolvido como parte da arquitetura *Integrated Services Digital Network* (ISDN) pelo *International Telecommunication Union - Telecommunication Standardization Sector* (ITU-T) para suportar os serviços de videoconferência e vídeotelefonia fazendo uso de *'n'* canais ISDN de voz de 64 kbps (até 2 Mbps). O H.261 usa um esquema de codificação híbrido

DCT/DPCM com compensação de movimento, semelhante ao usado pelo JPEG e MPEG-1, apesar de ter sido desenvolvido antes deles (SCHAFER, R. SIKORA, T., 1995). Numa seqüência de vídeo, as imagens geralmente são fortemente relacionadas, alguns componentes se movimentam, enquanto o pano de fundo da imagem é mantido. Então, o H.261 codifica a primeira imagem no modo intra-quadro (quadro *'I'*) e cada quadro subseqüente usa predição de compensação de movimento inter-quadros (quadro *'P'*) em relação ao quadro anterior. Novos quadros *'I'* são gerados a cada 132 quadros no H.261 para compensar possíveis erros acumulados ou quando o movimento detectado o justifica.

O H.263, também parte da arquitetura H.323, usa o mesmo esquema de codificação do H.261, otimizado para melhorar o desempenho. Pode ser até 50% mais eficiente no consumo de banda para uma mesma qualidade de vídeo. As diferenças do H.263 em relação ao H.261 são: maior precisão (meio *pixel*); partes da codificação hierárquica são opcionais, o que permite uso menor de banda e melhor recuperação de erro; interpolação bidirecional de quadros *'B'* (semelhante ao MPEG), e suporte a diferentes resoluções de imagem.

O ITU-T especifica ainda diversos padrões de codificação de áudio: G.711, G.722, G.723, G.726, G.728 e G.729, que integram a arquitetura H.323 (à exceção do G.726), e que também são usados em aplicações de Voz sobre IP (VoIP – *Voice over IP*) (HERSENT, J. P.; GURLE, D, 2005).

O padrão G.711 ou *Pulse Code Modulation* (PCM), que faz uso de escalas semi-logarítmicas para digitalizar a voz analógica, sendo a escala *μ-Law* usada nos Estados Unidos e Japão, e a escala *A-law* usada na Europa e demais países, inclusive o Brasil. Em telefonia, a voz analógica usa a faixa de freqüência de até 4 kHz, sendo amostrada a cada 125 μs. O G.711 codifica cada amostra de voz analógica usando 8 *bits*, requerendo 64 kbps para a transmissão de voz unidirecional. A qualidade de voz do G.711, obtida pela média de opinião de usuários *Mean Opinion Score* (MOS) é de 4,2. Em aplicações VoIP, cada pacote IP geralmente transporta 80 amostras G.711, em quadros de 10 ms, ou ainda 160 amostras em 20 ms.

O G.722 obtém melhor qualidade de voz usando um espectro de freqüência maior, de 7 kHz, em taxas de 48, 56 ou 64 kbps, mas não é largamente suportado por equipamentos de videoconferência. O codificador G.722.1 opera em 32, 24 e até 16 kbps, fazendo uso de quadros de 20 ms e *lookahead* (atraso inicial de codificação) de 20 ms.

O codificador G.723.1 opera em 5,3 kbps ou 6,4 kbps, em quadros de 30 ms e *lookahead* de 7,5 ms, com qualidade de voz menor (MOS de 3,7 em 5.3 kbps e 3,9 em 6.4 kbps). Usa técnicas de codificação *Multi-Pulse Maximum Likelihood Quantisation* (MP-

MLQ), e não se aplica a transmissão de música, de tons DTMF, nem de sinais de modem ou fax.

O G.726 usa codificação *Adaptive Differential Pulse Code Modulatio (*ADPCM)**,** para codificar amostras de áudio a cada 125 µs usando 2, 3 ou 4 *bits* a taxas de 16, 24 ou 32 kbps. Na taxa de 32 kbps oferece qualidade de voz (MOS de 4,3).

O padrão G.728 usa a técnica de codificação *Low-Delay, Code Excited Linear Prediction* (LD-CELP) e obtém qualidade (MOS de 4,3) usando apenas 16 kbps. O CELP é otimizado para voz: especifica padrões de sons vocais e usa este conjunto de códigos lineares de predição para encontrar o que melhor corresponde à forma de onda da voz a ser transmitida, enviando apenas os parâmetros que correspondem a este padrão, como por exemplo, o tom de voz. A codificação é lenta (cada amostra é obtida entre 625 µs e 2,5 ms), o que impacta o atraso fim-a-fim na transmissão de voz. O G.728 não é largamente suportado por equipamentos de videoconferência.

O padrão G.729 usa a técnica de codificação *Conjugate Structure, Algebraic Code Excited Linear Prediction* (CS-ACELP) e obtém MOS de 4,0 a taxas de apenas 8 kbps, em quadros de 80 *bits* a cada 10ms, com *lookahead* de 5 ms. É muito utilizado pela sua qualidade associada à economia de banda.

4.1.3 Requerimentos de Qos para Aplicações Multimídia

Diversas considerações sobre requerimentos de qualidade de serviço para aplicações avançadas na Internet são objeto de estudo pela Internet2 (MIRAS, 2002). Estes requerimentos, geralmente traduzidos em termos de métricas de parâmetros de rede (banda requerida, atraso, variação de atraso e perdas toleráveis), devem refletir a percepção de qualidade obtida pelos usuários das aplicações. Traçamos aqui algumas considerações utilizadas na especificação dos requisitos das aplicações do projeto InfraVIDA.

A banda é a capacidade requerida para a transmissão fim-a-fim dos pacotes que contém informação útil de uma dada aplicação. O atraso fim-a-fim é o valor acumulado dos atrasos de propagação, os introduzidos por codificação, empacotamento e serialização na origem, e por decodificação e compensação de variação de atrasos no destino, além dos introduzidos por enfileiramento de pacotes nos nós intermediários da rede. Os atrasos variáveis introduzidos por filas nos nós intermediários quando há congestionamento na rede são a causa da variação de atraso na recepção de diferentes pacotes, o que é categorizado

como *jitter*. O congestionamento na rede pode ser também a causa de perda de pacotes durante a sua transmissão.

Diferentes aplicações têm diferentes requerimentos em relação à banda, atraso, variação de atraso e tolerância a perdas. Aplicações ditas elásticas, como a transferência de dados, suportam restrições de banda e atraso, sem comprometer os dados obtidos. O mesmo não se aplica ao caso de serviços interativos de dados ou de aplicações em tempo real como voz ou vídeo, onde a percepção de qualidade pelo usuário é afetada por perdas ou atrasos.

A recomendação G.114 do ITU-T especifica o atraso máximo de 400 ms na transmissão fim-a-fim para voz, videoconferência e outras aplicações interativas, sendo recomendável que o atraso fim-a-fim não exceda 150 ms quando aplicações de perfil de tráfego muito variável sejam usadas [G.114, 2003].

A Figura 4-4, parte da recomendação G.114, mostra que atrasos fim-a-fim de até 200ms não afetam a percepção de qualidade de voz pelos usuários. O mesmo se aplica a aplicações de videoconferência, igualmente interativas. Aplicações não interativas de transferência de dados, imagens e vídeo sob demanda têm maior tolerância ao atraso.

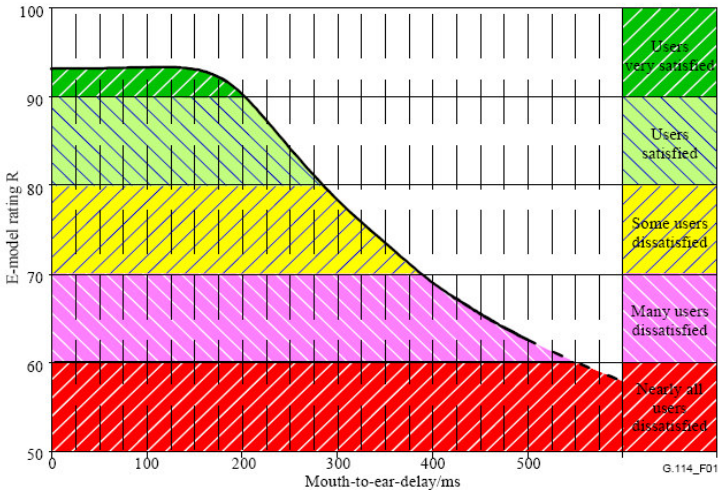


Figura 4-4: Efeitos do atraso fim-a-fim na percepção da qualidade de voz (ITU-T G.114, 2003)

A voz tem um perfil de tráfego ameno e é transportada em pacotes de tamanho pequeno em redes IP. Um canal de voz requer pouca banda útil, algo entre 8 a 64 kbps. A voz tem certa tolerância a perdas, já que é possível inferir o que é dito, mesmo que se percam pequenas partes da conversação. Mas é sensível ao atraso e à variação de atraso, que interferem na qualidade da conversação.

A natureza do tráfego de vídeo é extremamente variável, resultando em picos de tráfego, por exemplo, no momento do envio de quadros codificados em tela cheia, em contraste com o envio de quadros codificados apenas com informações de predição e compensação de movimento. Seu requerimento de banda também é extremamente variável e dependente do padrão de codificação adotado. Sua tolerância ao atraso é dependente da característica de interatividade da aplicação que faz uso do vídeo. A tolerância a variação de atraso é limitada pelas perdas de sinal que possam advir daí, pois a recuperação do vídeo pode ser afetada pela perda de parte das informações codificadas. Mecanismos de retenção temporária em *buffers* na recepção, que oferecem compensação de variação de atraso, são geralmente disponibilizados à nível de aplicação.

A transferência de imagens requer larguras de banda variáveis, dependendo da resolução da imagem e do tipo de codificação usado. A imagem é sensível a perdas de informação (principalmente considerando-se imagens médicas). Por não ser uma aplicação em tempo real, não é comprometida por atrasos e pela variação de atraso.

Para que o conteúdo multimídia seja transportado fim-a-fim em redes IP, o protocolo RTP estabelece e mantém sessões para cada canal de áudio e vídeo e mantém a sincronização entre elas, além de reportar o andamento das sessões com auxílio do RTCP (SCHULZRINNE et al., 2003). O conteúdo multimídia, por sua característica dinâmica de serviço em tempo real, não deve ser submetido ao controle de fluxo ou à retransmissão oferecidos pelo TCP, sendo adequado o seu transporte pelo protocolo UDP em redes IP.

Assim, a saída do codificador, seja de áudio, vídeo ou imagem, é empacotada para transporte pela rede IP, ou seja, é sucessivamente encapsulada, recebendo os cabeçalhos dos protocolos RTP, UDP e IP, e ainda o cabeçalho de camada 2 do enlace de rede que o transporta, seja Ethernet, PPP ou Frame Relay. Isto é ilustrado na Figura 4-5, onde se pode verificar que o pacote recebe um volume adicional de 40 *bytes* (além de 9 *bytes* do quadro PPP, se considerarmos o seu transporte em enlaces de longa distância).

40 bytes = 20 *bytes* do IP + 8 *bytes* do UDP + 12 *bytes* do RTP

camada 2	IP	UDP	RTP	conteúdo multimídia
9 *bytes* (PPP)	20 *bytes*	18 *bytes*	12 *bytes*	variável

Figura 4-5: Encapsulamento de conteúdo multimídia em redes IP (AUTOR, 2008)

Este volume adicional do cabeçalho é considerável se o tamanho útil dos pacotes for pequeno. Isto é verdade no caso da voz. Por exemplo, no caso do codificador G.711 (SOLLAUD, 2006), em que tipicamente um pacote seja enviado a cada 20 ms, temos 50 pacotes por segundo (pps). A banda útil por canal de voz G.711 é 64 kbps e o tamanho de pacote é de 1280 *bits*, já que 64.000 bps / 50 pps = 1280 *bits* ou 160 *bytes*/pacote. Assim um único pacote G.711 gerado à taxa de 50 pps tem o tamanho de 200 *bytes* = 40 *bytes* de cabeçalho (IP+UDP+RTP) + 160 *bytes*. A 50 pps, gera-se um fluxo de 10.000 *bytes* ou 80 kbps. Verifica-se aí um peso adicional de 20% de banda devido ao encapsulamento, de 64 kbps para 80 kbps.

No caso do transporte de vídeo H.261 para videoconferência em redes H.323 (EVEN, 2006), deve-se considerar que a saída do codificador entrega quadros H.261 de 512 *bits*, sendo 2 *bits* de sincronismo, 492 de carga útil e 18 *bits* para correção de erro. O H.323 interlaça os canais de áudio e vídeo, o que nos leva a considerar os mesmos parâmetros do exemplo do G.711, ou seja, 160 amostras por pacote a cada 20 ms e 50 pps. A carga por pacote é de 160 amostras x 512 *bits* = 81.920 *bits* ou 10.240 *bytes*. Para o encapsulamento do H.261 em RTP, acrescentamos 4 *bytes* de cabeçalho H.261 aos 40 *bytes* de encapsulamento IP+UDP+RTP. Verificamos que o custo do encapsulamento IP+UDP+RTP+H.261 é desprezível em relação ao total de 10.240 + 44 = 10.284 *bytes*.

O mesmo ocorre no transporte de imagem JPEG ou vídeo MPEG. Isto porque, a informação útil tende a ser muito maior que o tamanho de um pacote IP, que tem o tamanho máximo de 65.536 *bytes*. Por exemplo, o quadro na saída do codificador JPEG [BERC et al., 1998] tem uma carga útil de até 2^24 *bytes*, ou seja, pode ocupar até 256 pacotes IP. Mesmo acrescentando 8 *bytes* do JPEG ao cabeçalho IP+UDP+RTP de 40 *bytes*, o custo do encapsulamento IP+UDP+RTP+JPEG é desprezível em relação ao total do pacote IP e não precisa ser considerado no dimensionamento da rede. O mesmo vale para o MPEG [HOFFMAN et al., 1998] (WENGER et al., 2005).

4.2 QUALIDADE DE SERVIÇO PARA AS APLICAÇÕES DO INFRAVIDA

4.2.1 Especificação dos Parâmetros dos Serviços do Infravida

Para o InfraVIDA foram inicialmente considerados os formatos de imagem, vídeo e áudio suportados pelas implementações adotadas para os serviços de Videoconferência e IoD/VoD, no escopo do projeto.

No caso dos serviços de IoD/VoD, os padrões de codificação considerados são o JPEG2000 para imagens, o MPEG-2 para vídeos. As vantagens do JPEG2000 sobre o JPEG justificam a opção pelo JPEG2000 no escopo do InfraVIDA: além da distribuição progressiva de imagens, o JPEG2000 oferece melhor qualidade de imagem para arquivos do mesmo tamanho ou redução de 25-35% no tamanho de arquivos para qualidade de imagem equivalente. O padrão MPEG-2 é adequado à exigência de qualidade na transmissão de imagens médicas e, apesar de consumir maior banda de transmissão, foi considerado por ter sido objeto de implementação dentro do escopo do projeto InfraVIDA (ARAÚJO et al., 2003).

No caso do serviço de Videoconferência para Segunda Opinião Médica, é considerado o uso dos padrões G.711 para o áudio e H.261 para o vídeo, por serem os de melhor resultado na implementação do *Open H.323* no escopo do projeto InfraVIDA (RABELO et al., 2001). O G.711 requer 64 kbps para cada canal de áudio unidirecional. O padrão de vídeo H.261 suporta velocidades múltiplas de 64 kbps, mas sua implementação do codificador no *Open H.323* possibilita apenas o uso de 64 kbps ou 128 kbps, sendo o H.263 ainda não bem suportado no *Open H.323*, no momento da implantação.

Baseadas nas considerações sobre os requerimentos de QoS das aplicações feitas na seção anterior foram definidas as métricas dos serviços de Videoconferência e IoD/VoD.

As métricas dos parâmetros de QoS para Videoconferência são então: banda de 64 ou 128 kbps, atraso de até 200 ms fim-a-fim, variação de atraso de até 30 ms, perda de até 3% para o vídeo H.261; e para o áudio G.711, banda de 80 kbps, considerando-se encapsulamento, atraso de até 200 ms, variação de atraso (*jitter*) de até 30 ms e perda de até 10%.

As métricas para o serviço de IoD são estimativas adotadas no contexto do projeto InfraVIDA considerando imagens médicas típicas comprimidas com JPEG2000: banda típica variável de 200 kbps a 4 Mbps, atraso de até 1 seg e tolerância a variação de atraso (*jitter*), já que não é uma aplicação em tempo real, e perdas de 1%.

As métricas para o serviço VoD com uso de MPEG-2 são banda de 4 Mbps, atraso de até 2 seg, variação de atraso (*jitter*) de 20 ms e perda menor que 1%. Tais valores são baseados em estudo de avaliação de qualidade de vídeo com introdução de fonte de ruído via simulação [ARAÚJO et al., 2003].

O Quadro 4-1 resume a especificação dos parâmetros de qualidade de serviço para aplicações multimídia no InfraVIDA:

Serviço	Banda	Atraso	*Jitter*	Perda
Imagem sob Demanda				
JPEG2000	200 kbps / 4 Mbps	< 1 seg	---	< 1 %
Vídeo sob Demanda				
MPEG-2	4 Mbps	< 2 seg	< 20 ms	< 1%
Videoconferência				
Áudio (G.711)	80 kbps	< 200 ms	< 30 ms	< 10%
Vídeo (H.261)	64 ou 128 kbps	< 200 ms	< 30 ms	< 3%

Quadro 4-1 Especificação dos parâmetros dos serviços de Imagem, vídeo e áudio para o InfraVIDA (LAGE, 2004)

Uma possível revisão dos padrões de codificação no escopo do projeto InfraVIDA poderia incluir o mais recente padrão H.264 (MPEG-4 AVC) para o serviço de VoD, e os padrões H.263 para vídeo e G.729 para voz, para o serviço de Videoconferência.

O H.264 ou MPEG-4 AVC requer 768 kbps, com resolução de 352x288 pixels a 30 quadros/s (nível 1) e 2 Mbps ou 4 Mbps, com resolução de 352x480 pixels a 30 quadros/s (nível 2), com qualidade superior de imagem que a do MPEG-2, que requer 4 Mbps, para resolução de vídeo de 352x288 pixels a 30 quadros/s (nível baixo).

O H.263 oferece eficiência de 50% da banda para a mesma qualidade de imagem de videoconferência. Já o G.729 oferece uma maior eficiência ainda na banda por canal de voz. A banda útil requerida é de apenas 8 kbps, com o uso do algoritmo CS-ACELP. Se considerarmos 50 pps, o tamanho útil do pacote é de 160 *bits*, já que 8.000 bps / 50 pps = 160 *bits* ou 20 *bytes*/pacote. Acrescentando-se 40 *bytes* de cabeçalho (IP+UDP+RTP) aos 20 *bytes* do G.729, temos pacotes de 60 bytes, onde se paga um preço adicional de 200% da banda útil em encapsulamento. Mesmo assim, a banda total requerida é de apenas 24 kbps.

O Quadro 4-1a resume a revisão das métricas para os serviços do InfraVIDA.

Serviço	Banda	Atraso	*Jitter*	Perda
Imagem sob Demanda				
JPEG2000	200 kbps / 4 Mbps	< 1 seg	---	< 1 %
Vídeo sob Demanda				
H.264	768 kbps / 4 Mbps	< 2 seg	< 20 ms	< 1%
Videoconferência				
Áudio (G.711)	24 kbps	< 200 ms	< 30 ms	< 10%
Vídeo (H.263)	64 ou 128 kbps	< 200 ms	< 30 ms	< 3%

Quadro 4-1a - Revisão da especificação dos parâmetros dos serviços de imagem, vídeo e áudio para o Projeto InfraVIDA (AUTOR, 2008)

Outra otimização é adotar o cabeçalho RTP comprimido [KOREN et al., 2003], de 12 *bytes* para 1 ou 2 *bytes*, ainda não suportado por muitos dispositivos.

4.2.2 Mapeamento dos Serviços Infravida em Classes de Qos Diffserv

Uma vez definidos os parâmetros dos serviços de IoD/VoD e Videoconferência para o projeto InfraVIDA, as classes de serviço e o mapeamento de perfis de tráfego nessas classes traduzidas em *DiffServ Per-Hop Behaviors* e DHCP são especificados, considerando-se ainda que os serviços de transferência de documentos médicos devem ter maior prioridade sobre os demais serviços de transferência de dados.

As classes de serviço definidas são:

- O serviço ***Expedited Forwarding*** (***EF -****Premium*) oferece garantia mínima de banda, baixa latência e baixa variação de atraso e tem prioridade absoluta sobre os demais;

- Os serviços ***Assured Forwarding*** (*AFx)* (Platina, Ouro, Prata e Bronze) equivalem a classes de serviços diferenciados, com possibilidade de até três níveis de prioridade de descarte em cada classe;

- O serviço ***Default*** (Padrão) não oferece nenhuma garantia de entrega.

A partir dos requisitos de QoS das aplicações do projeto InfraVIDA é feito a seguir o seu mapeamento em classes de serviço (CoSs) *DiffServ*, com a definição dos respectivos valores de DSCP e do tratamento que os pacotes devem receber pelos roteadores (PHBs). Estas definições serão usadas na implantação da rede protótipo experimental.

O mapeamento das aplicações InfraVIDA em classes de serviço leva em conta as características dos serviços de áudio e vídeo e a dinâmica da aplicação de Segunda Opinião Médica do InfraVIDA. O vídeo requer maior largura de banda e baixa taxa de erro, a imagem requer largura de banda variável, enquanto a voz requer menor banda, pode suportar uma taxa de erro maior que o vídeo, mas requer atraso mínimo fim-a-fim. Por esta característica, o áudio foi mapeado para o serviço Premium (LAGE, 2004).

A aplicação de Segunda Opinião Médica implica em uma sessão de Videoconferência entre profissionais de saúde, durante a qual podem ser solicitados Vídeos ou Imagens sob Demanda. Durante a visualização de um vídeo ou imagem médica, o seu conteúdo é mais importante para os médicos do que a visualização das imagens dos outros participantes da videoconferência. Ou seja, um vídeo ou imagem no contexto de uma sessão de Segunda Opinião Médica ganha uma característica conversacional.

Enquanto o serviço de IoD/VoD deve ser priorizado no contexto da Segunda Opinião, outras imagens e vídeos solicitados, por exemplo, no contexto de EaD (Ensino à Distância) não devem ganhar prioridade, pois consumiriam muitos recursos, prejudicariam a qualidade da aplicação foco e poderiam inclusive bloquear a rede. Em função disto, devem ser definidos diferentes SLAs ou contratos de serviço para as diferentes aplicações (como EaD e Segunda

Opinião, por exemplo), de forma a distinguir serviços de recuperação de conteúdo e serviços conversacionais.

O Quadro 4-2 mostra o mapeamento das aplicações InfraVIDA em classes de serviço.

Classes de Serviço	***Per-Hop Behaviors DiffServ***	**Valor de DSCP**	**Mapeamento das Aplicações**
Premium	EF *Expedited Forwarding*	(DSCP 5) '101110'	Videoconferência (áudio G.711)
Platina	AF4 *Assured Forwarding*	(DSCP 4) '100010'	Videoconferência (vídeo H.261)
Ouro	AF3 *Assured Forwarding*	(DSCP 3) '011010'	VoD - Vídeo sob Demanda (MPEG-2) ref. 2ª opinião InfraVIDA
Prata	AF2 *Assured Forwarding*	(DSCP 2) '010010'	IoD - Imagem sob Demanda (JPEG2000) ref. 2ª opinião InfraVIDA
Bronze	AF1 *Assured Forwarding*	(DSCP 1) '001010'	Documentos médicos ref. 2ª opinião InfraVIDA
Regular	BE *Best Effort*	(DSCP 0) '000000'	Documentos, imagens e vídeos de outras aplicações diferentes da 2ª. opinião

Quadro 4-2 Mapeamento de serviços InfraVIDA nas classes de serviço *DiffServ* (LAGE, 2004)

4.2.3 Especificações dos Cenários e Serviços Diffserv para o Infravida

Foi definido o estudo de três cenários, ilustrados na Figura 4-6: um cenário de rede local com enlace de 10 Mbps, um cenário em rede de longa distância com enlaces de 2 Mbps e outro cenário com nós remotos em rede de longa distância com enlaces de 512 kbps.

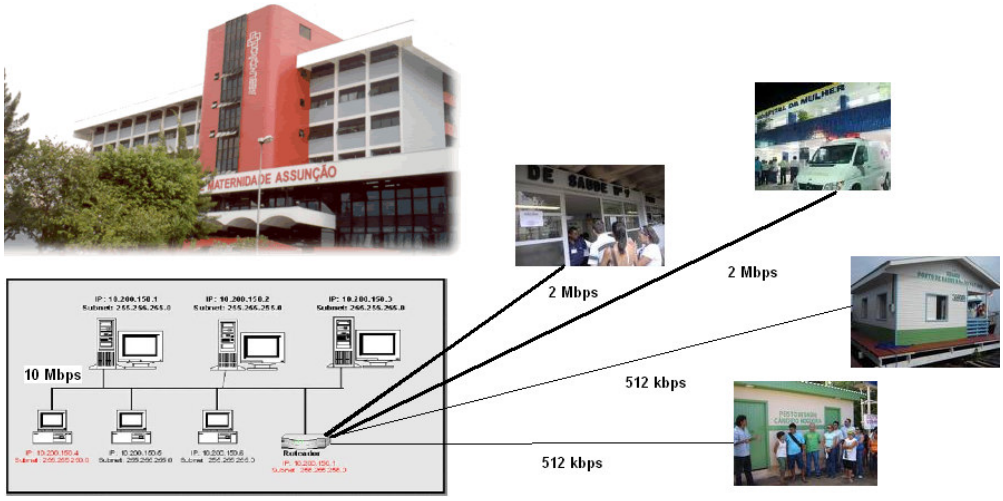


Figura 4-6: Cenários considerados no Projeto InfraVIDA (Autor, 2008)

Algumas premissas foram assumidas:

- oferecer uma garantia de até 4 participantes de videoconferência, o que implica no estabelecimento de 3 sessões de áudio (3x80 kbps) e 3 sessões de vídeo (3x128 kbps ou 3x64 kbps) em *full-duplex* com o 'Consultor', considerando distribuição centralizada. Considera-se

o tamanho médio dos pacotes de áudio e vídeo na videoconferência de 200 *bytes*. O serviço tem prioridade sobre as demais aplicações InfraVIDA, sendo que o áudio tem prioridade absoluta, para garantia de atraso e de variação de atraso mínimos;

- otimizar a distribuição de VoD, oferecendo banda de 4 Mbps equivalente à de distribuição em tempo real onde houver banda disponível (cenário 1), banda de 1 Mbps para distribuição em vídeo *streaming* no cenário 2, e bloqueando a distribuição de vídeo onde ele não puder ser entregue com qualidade sem monopolizar recursos e prejudicar as demais aplicações (cenário 3). Considera-se o tamanho médio dos pacotes de vídeo de 1500 *bytes*. O serviço de distribuição de vídeo tem prioridade sobre o serviço de distribuição de imagem;

- oferecer possibilidade de distribuir imagem típica de 10 *Mbytes* em aproximadamente 30 segundos (cenário 1), 2min30seg no cenário 2 ou 5 minutos no cenário 3 (cálculo aproximado da banda requerida em rede local: 10 Mbytes / 30seg ~= 2,5 Mbps). Considera-se o tamanho médio dos pacotes de imagem sob demanda de 1500 *bytes*. O serviço de distribuição de imagens tem prioridade sobre a distribuição de documentos;

- oferecer prioridade a documentos médicos (prontuários, anotações e outros documentos) sobre o tráfego das demais aplicações. Considera-se o tamanho médio dos pacotes de documentos médicos transportados via FTP de 1500 *bytes*;

- reservar uma parte da banda do enlace para o tráfego de outras aplicações de menor prioridade.

O Quadro 4-3 resume as especificações dos cenários e serviços *DiffServ* definidos.

				Cenário 01 (10 Mbps)	**Cenário 02 (2 Mbps)**	**Cenário 03 (512 kbps)**
Serviço	Formato	Característica do Serviço	Classe de Serviço	Banda alocada	Banda alocada	Banda alocada
Videoconf. (áudio)	G.711	80 kbps (200 bytes)	EF *Premium*	3 x 80 kbps 240 kbps	2 x 80 kbps 160 kbps	1 x 80 kbps 80 kbps
Videoconf. (vídeo)	H.261	64 / 128kbps (200 bytes)	AF4 Platina	3 x 128 kbps 384 kbps	2 x 64 kbps 128 kbps	1 x 64 kbps 64 kbps
Vídeo sob Demanda	MPEG-2	4 Mbps (1500 bytes)	AF3 Ouro	4 Mbps 4096 kbps	1 Mbps (*) 1024 kbps	---- (**)
Imagem sob Demanda	JPEG2K	10 Mbytes (1500 bytes)	AF2 Prata	2.5 Mbps 2560 kbps	480 kbps	240 kbps
Documentos Médicos	(vários FTP)	200 kbps (1500 bytes)	AF1 Bronze	512 kbps	128 kbps	64 kbps
Outras Aplicações	(vários)	(variável)	BE Regular	512 kbps	128 kbps	64 kbps

(*) *video streaming;* (**) distribuição de vídeo bloqueada, por não poder ser oferecida com qualidade
Quadro 4-3 Especificações dos serviços *DiffServ* para o InfraVIDA (LAGE, 2004)

É importante estudar uma definição de controle de admissão no caso de novo fluxo de tráfego de um serviço de maior prioridade solicitar mais recursos do que os disponíveis na rede. O serviço de áudio em videoconferência foi mapeado como *Premium*. Para manter

coerência com esta especificação, todo e qualquer nova solicitação de sessão de videoconferência deve ser aceita, mesmo que isto implique em realocar recursos de outras aplicações. O serviço *Premium* (EF) pode ser implementado com *Priority Queueing*, um algoritmo de escalonamento dos pacotes em filas com prioridade absoluta.

Os serviços de Vídeo e Imagem sob Demanda devem ser aceitos até o limite de banda especificado, fazendo uso de distribuição progressiva em vídeo *streaming* ou transmissão progressiva de imagens, onde se aplicar. Tráfegos excedentes aos limites de banda dos serviços IoD/VoD em cada cenário podem ser aceitos se houver banda disponível, mas devem ter os pacotes marcados com maior precedência de descarte em caso de congestionamento. Por exemplo, o tráfego de VoD dentro do limite de banda do cenário deve ser marcado com DSCP ’011010’ (AF31) e o excedente deve ser marcado com DSCP ’011100’ (AF32). O tráfego de IoD dentro do limite de banda do cenário deve ser marcado com DSCP ’100010’ (AF41) e o excedente deve ser marcado com DSCP ’100100’ (AF42). A Figura 2-5 - PHB AF - *Assured Forwarding* permite visualizar as relações de prioridade de tratamento e descarte das classes AF e valores de DSCP associados.

Recomenda-se que no InfraVIDA, onde imagens e vídeos médicos tipicamente de alta resolução geram arquivos de grande tamanho, haja um servidor de vídeo na rede local dos grandes hospitais e que outras localidades importantes tenham enlaces de pelo menos 2 Mbps com estes hospitais para viabilizar a distribuição destes vídeos. Em localidades remotas, com enlaces de 512 kbps ou menores a distribuição de vídeo é bloqueada por não poder ser oferecida com qualidade, e apenas a distribuição de imagens é feita.

4.3 GERENCIAMENTO INTEGRADO DE QOS PARA APLICAÇÕES DO INFRAVIDA

A aplicação dos mecanismos de QoS que garantem os serviços especificados para cada uma das classes de serviço *DiffServ* pode ser implementada através de configuração manual de cada um dos roteadores que compõem o domínio *DiffServ*. Esta é a primeira abordagem adotada no projeto InfraVIDA, onde diferentes entidades terão suas redes interligadas à RNP, todas com diferentes autoridades administrativas.

No entanto, uma solução de gerência e configuração automatizada é desejável para garantir flexibilidade e escalabilidade. Esta solução foi considerada através da prototipação do *Bandwidth Broker* (BB), que tem a função de automatizar a tomada de decisões de controle de admissão de requisições de clientes, o gerenciamento de recursos e a configuração dos

elementos de rede, de acordo com o conjunto de políticas de provisionamento de serviços na rede (CHIMENTO et al., 2002).

O BB recebe dinamicamente uma requisição de alocação de recursos para cada fluxo entrante, tem uma visão completa da disponibilidade e alocação dos recursos da rede, e pode autorizar e aceitar o fluxo após validação do contrato de serviço (SLA) e, inclusive, configurar dinamicamente os roteadores para aceitar novos perfis de tráfego.

A opção de introdução do BB na arquitetura de suporte proposta de gerenciamento de Qualidade de Serviço para as aplicações de IoD/VoD e de Videoconferência propicia o aspecto de integração dos serviços na arquitetura de suporte. Esta integração, ao nível do plano de controle, favorece a sinalização das requisições e autorizações de disponibilização de recursos de rede para as aplicações, realizadas entre os elementos gerenciadores da rede de vídeo digital (Gerenciador do Serviço de Vídeo) e da rede H.323 de videoconferência (*Gatekeeper*) e o elemento gerenciador de Qualidade de Serviço (*Bandwidth Broker*).

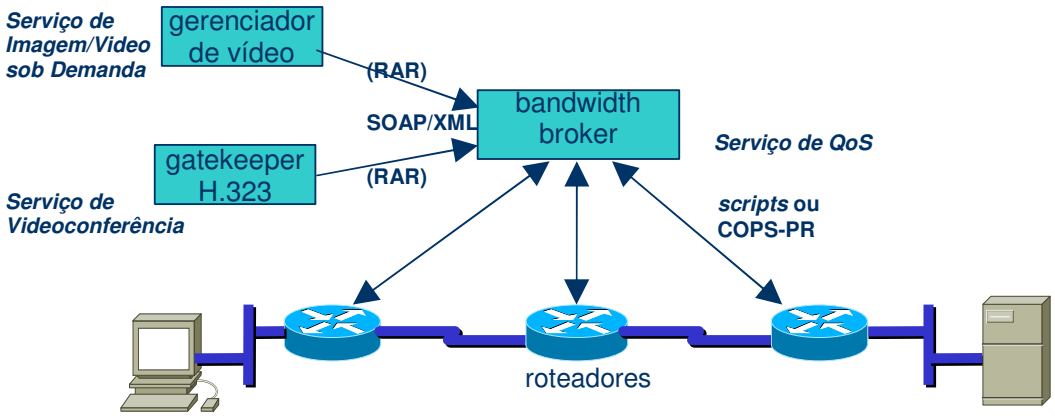


Figura 4-7: Integração das aplicações no nível do plano de controle (LAGE, 2004)

A Figura 4-7 mostra a integração que propomos entre os componentes do plano de controle dos sistemas de Videoconferência (*gatekeeper*) e VoD (gerenciador de vídeo) com o servidor de políticas de QoS e gerenciador dos recursos de rede (BB). Na prototipação do BB, as requisições de alocação de recursos (RARs) são transmitidas ao BB através de mensagens SOAP/XML, enquanto que as configurações de QoS nos roteadores podem ser acionadas manualmente via roteiros (*scripts*) ou automaticamente via protocolo COPS-PR. A interação entre os elementos componentes da solução é detalhada a seguir.

4.4 ARQUITETURA DE SUPORTE DE QOS PARA SERVIÇO DE VÍDEO SOB DEMANDA

4.4.1 Arquitetura do Serviço de Vídeo Sob Demanda

O serviço de IoD/VoD faz uso da rede de vídeo digital *DynaVideo*. Nessa seção o funcionamento da rede *DynaVideo* proposta pelo GTVD-RNP é descrito e ilustrado para permitir que a partir daí sejam construídos os elementos de integração ao gerenciamento de QoS propostos como uma das contribuições dessa dissertação e descritos na seção seguinte. A Figura 4-8 ilustra a troca de informações entre os elementos do sistema de VoD.

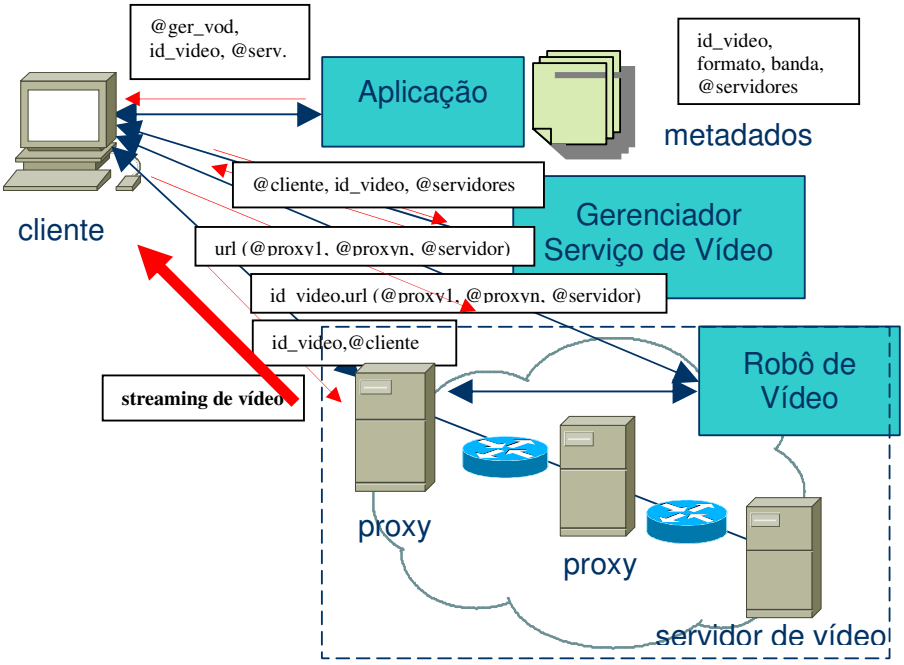


Figura 4-8: Troca de informações entre elementos do serviço de VoD (LAGE et al., 2004b)

No serviço de VoD:

- O usuário usa uma aplicação de busca e respectiva base de metadados para obter o identificador de um determinado vídeo e os endereços de servidores onde cópias deste vídeo são armazenadas;

- A aplicação cliente então encaminha o identificador de vídeo e os endereços de servidores para um elemento gerenciador do serviço de vídeo;

- O gerenciador do serviço de vídeo, de posse do identificador do vídeo solicitado, calcula um grafo e gera um arquivo XML com a identificação do melhor servidor e dos *proxies* no caminho até o servidor que arquiva o vídeo em caráter permanente;

- O gerenciador do serviço de vídeo passa esta informação ao cliente em forma de uma URL;

- A aplicação cliente, de posse da URL, então instancia o aplicativo de reprodução de vídeo e dispara automaticamente o robô de distribuição do vídeo.

Um procedimento equivalente é usado no serviço de IoD.

4.4.2 Arquitetura de Suporte Integrado de Qos Para Video Sob Demanda

A arquitetura de QoS proposta nessa dissertação prevê a integração entre o elemento gerenciador do serviço de vídeo *DynaVideo* e do elemento servidor de políticas de QoS (BB), que responde a requisições de alocação de recursos (RARs) do Gerenciador do Serviço de Vídeo (LAGE et al., 2004b).

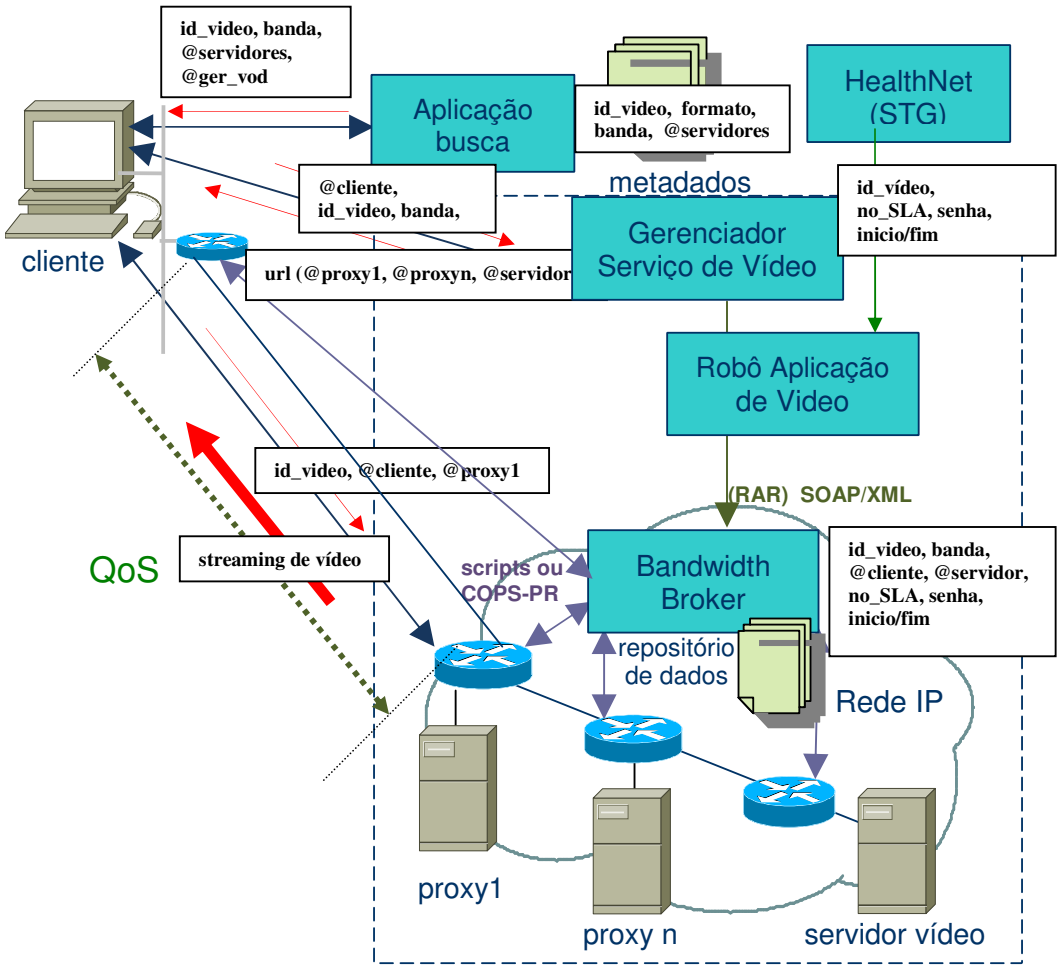


Figura 4-9: Integração dos serviços de VoD e QoS para Segunda Opinião Médica (LAGE et al., 2004b)

A Figura 4-9 ilustra a integração dos elementos do plano de controle dos serviços de

gerência de QoS e distribuição de imagem e vídeo com a aplicação de cadastro geral e agendamento de Segunda Opinião (STG do *HealthNet*).

A seguir é descrito o funcionamento da solução integrada proposta:

- o STG do *HealthNet* deve possuir a informação dos SLAs (número de contrato, e senha) associados a cada aplicação em uma dada instituição cliente. Diferentes SLAs devem existir para cada serviço (IoD, VoD e Videoconferência) associado à Segunda Opinião. Outros SLAs podem ser associados aos mesmos serviços quando usados por outras aplicações de menor prioridade;

- o STG do *HealthNet,* como aplicação responsável pelo agendamento das sessões de Segunda Opinião, deve obter através de aplicação de busca e base de metadados do *DynaVideo* a identificação de cada imagem ou vídeo, formato e banda da imagem ou vídeo e os endereços dos servidores onde há cópias do conteúdo.

- o STG deve comunicar ao gerenciador de vídeo a identificação de vídeo, formato, banda e servidores de conteúdo, além de número de contrato e senha associados ao serviço de VoD para uma dada instituição cliente e os horários de inicio e fim das sessões agendadas. Isto permitirá ao gerenciador de vídeo gerar uma requisição de alocação de recursos (RAR) para o BB;

- o STG deve passar o identificador de vídeo, o formato e a banda do vídeo solicitado, além do endereço IP do cliente ao robô da camada de aplicação do *DynaVideo*. A informação é passada ao robô de vídeo para manter a generalidade da solução através da desacoplagem das camadas;

- o robô de vídeo deve implementar mecanismo de marcação do campo DSCP dos pacotes IP de Imagem ou Vídeo sob Demanda que tem origem no servidor de conteúdo. O valor de DSCP é associado ao SLA correspondente.

As requisições de alocação de recursos para cada fluxo especificam a banda requerida por diferentes formatos de vídeo e imagem, endereços IP de origem e destino dos fluxos, além de identificador de contrato de serviço (SLA) e senha, e horários de agendamento de distribuição da imagem/vídeo para a sessão de Segunda Opinião do InfraVIDA. A comunicação é feita usando SOAP/XML.

A especificação proposta de RAR é mostrada no Quadro 4-4.

RAR.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAPENV="
http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
<slaNumber>5</slaNumber>
<passWord>sla005</passWord>
<startDate>2003-10-10</startDate>
<startTime>00:00:00</startTime>
<endDate>2003-11-11</endDate>
<endTime>00:00:00</endTime>
<bandwidth>6000</bandwidth>
<sourceIP>127.1.1.0</sourceIP>
<destIP>127.1.1.1</destIP>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Quadro 4-4 RAR em XML/SOAP (Autor, 2008)

São a seguir feitas considerações sobre a arquitetura proposta:

O *DynaVideo* suporta qualquer formato decodificado pelos aplicativos de reprodução de vídeo comerciais. A codificação é transparente para o serviço de distribuição de vídeo, ela só afeta os aplicativos de reprodução de vídeo. No entanto, a codificação de vídeo não é transparente para o serviço de QoS, pois a taxa em que o vídeo vai ser disparado é um parâmetro crucial para o controle de admissão pelo BB, que vai decidir em função da disponibilidade do recurso de banda na rede. O mesmo vale para a distribuição de IoD.

Na implementação atual, o gerenciador do serviço de vídeo não possui as informações de formato e banda requerida para a distribuição de vídeos específicos. Estas informações estão na base de metadados da aplicação de busca acessada pelo cliente, mas não são passadas ao gerenciador de vídeo, que recebe apenas a identificação do vídeo e endereços dos servidores para o cálculo da melhor cadeia de *proxies*-servidor para distribuir aquele vídeo para aquele cliente. Para viabilizar a arquitetura proposta, a informação de banda de imagens e vídeos existentes na base de metadados da aplicação de busca do *DynaVideo* deve ser retornada à aplicação cliente.

O sistema de IoD/VoD deve disponibilizar imagens e vídeos nos servidores mais próximos aos clientes nos horários e localizações em que serão solicitados (por agendamento) para o serviço de Segunda Opinião. Para isto é necessária a integração do sistema *DynaVideo* e do STG do *HealthNet*, que faz o agendamento de sessões de Segunda Opinião com a identificação dos participantes e imagens e vídeos relacionados a um determinado caso médico. Receber as requisições do STG no agendamento, permite que o *DynaVideo* disponibilize os vídeos no *proxy* mais próximo ao cliente previamente, otimizando a resposta

do sistema. De outra forma, as requisições do STG podem ser enviadas ao se iniciar a sessão de Segunda Opinião.

Deve ser implementado mecanismo de marcação do campo DSCP dos pacotes IP de imagens ou vídeo transmitidos para mapeamento das classes de serviço *DiffServ* e conseqüente tratamento diferenciado pela rede para garantia de QoS. Esta marcação deve ser feita pela aplicação que dá origem aos fluxos. É necessário, portanto que o robô de vídeo conheça o valor de DSCP correspondente para marcação dos pacotes que tem origem no servidor de conteúdo. O valor de DSCP é associado ao SLA de cada serviço IoD/VoD associado a uma aplicação (Segunda Opinião ou outra) para cada instituição.

4.5 ARQUITETURA DE SUPORTE DE QOS PARA SERVIÇO DE VIDEOCONFERÊNCIA

4.5.1 Arquitetura do Serviço de Vídeoconferência

O sistema de Videoconferência do InfraVIDA é baseado no *OpenH323* (RABELO et al., 2001), uma implementação em código aberto do H.323, cuja arquitetura H.323 é descrita na seção 3.5.

4.5.2 Arquitetura De Suporte Integrado De Qos Para Vídeoconferência

Na arquitetura do serviço de Videoconferência, o GK é o elemento do plano de controle responsável pelo controle de admissão e registro dos participantes em uma dada sala/sessão de videoconferência, inclusive endereços IP e portas alocadas para as sessões RTP de áudio e vídeo. Além disto, o GK tem informações de controle de banda alocada a cada sessão.

A proposta de integração do serviço de Videoconferência com os mecanismos de QoS prevê uma comunicação dinâmica do GK com o BB a cada nova sessão de videoconferência e a cada cliente adicionado a sessões pré-existentes. A arquitetura proposta é ilustrada na Figura 4-10.

A comunicação entre o GK e o BB é feita via protocolo SOAP/XML e a requisição de alocação de recursos (RAR) usa o mesmo formato mostrado no Quadro 4-4.

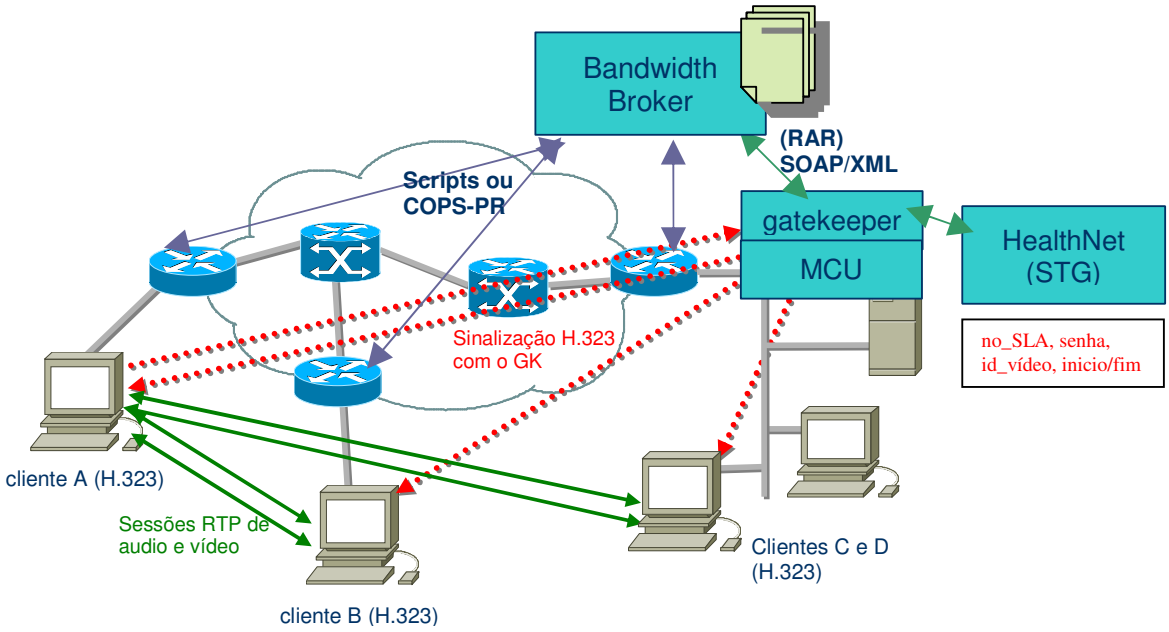


Figura 4-10: Integração de elementos dos serviços de Videoconferência e QoS (LAGE, 2004)

No caso da Videoconferência duas requisições de recursos (RARs) para as sessões paralelas de áudio e vídeo devem ser geradas pelo GK, contendo os endereços IP dos clientes envolvidos (ou MCU, se for o caso), banda (associada aos formatos de áudio ou vídeo utilizados), além de identificador de SLA, senha e horários de início e fim das sessões. No caso de distribuição *multicast* a requisição de alocação de recursos contém o endereço IP do *root* e o endereço de *multicast*, banda, além de identificador de SLA, senha e horários de início e fim das sessões.

Para viabilizar a proposta de integração, algumas novas funcionalidades devem ser implementadas pelos sistemas de Videoconferência e STG:

- o STG do *HealthNet* deve possuir a informação de SLA (número de contrato e senha) associada aos serviços de áudio e vídeo associados à aplicação de videoconferência para uma dada instituição cliente;

- o serviço de Videoconferência deve estar integrado com o sistema de agendamento das sessões de Segunda Opinião (STG) para que o *Gatekeeper* aceite as informações de horários de início e fim das sessões agendadas, número de contrato de serviço e senha;

- o GK faz o registro de salas de videoconferência a cada novo registro de sessão ou de participante no GK. Deve ser desenvolvido um módulo aplicativo capaz de ler o arquivo de registro das salas / sessões / participantes do GK e os dados obtidos na interação com o STG para gerar as requisições de alocação de recursos (RAR) referentes às sessões de áudio e vídeo a serem enviados ao BB;

- a aplicação cliente deve integrar mecanismo de marcação do campo DSCP dos pacotes IP de videoconferência que tem origem nos *hosts*. O valor de DSCP é associado ao SLA correspondente.

CAPÍTULO 5 IMPLEMENTAÇÃO DA REDE EXPERIMENTAL DE TESTES *(TESTBED)*

Este capítulo descreve um cenário de validação com a implementação do *Bandwidth Broker* em Java em laboratório e validação dos resultados da aplicação de mecanismos de QoS *DiffServ* em rede protótipo experimental Linux na simulação de diversos cenários de aplicação para o Projeto InfraVIDA.

5.1 REDE PROTÓTIPO EXPERIMENTAL (*TESTBED*)

No capítulo anterior foram especificados os parâmetros de QoS para as aplicações de telemedicina no contexto do projeto InfraVIDA e o mapeamento desses parâmetros para CoSs *DiffServ*. Foram ainda definidos cenários de aplicação, que serviram de orientação à especificação e implantação de uma rede protótipo experimental, com o objetivo de validar as especificações de QoS das aplicações do projeto InfraVIDA. Resultados obtidos na rede protótipo experimental foram reportados em artigos publicados (LAGE et al., 2004a, 2004b)

O ambiente operacional adotado para implantação da rede protótipo experimental é o GNU/Linux por ser código aberto, permitir uma configuração flexível de roteadores e oferecer suporte ao *DiffServ*. A distribuição Red Hat 9 foi utilizada. O suporte ao *DiffServ* no GNU/Linux exige versão do *kernel* 2.4 ou mais recente e é oferecida dentro de uma arquitetura mais genérica de controle de tráfego. O utilitário *tc* (*traffic control*) é parte do pacote *iproute2* que já é parte das distribuições mais recentes do Linux (LINUX DIFFSERV, 2001).

O *traffic control* do GNU/Linux permite a implantação de inúmeras políticas de controle de tráfego através de uma combinação de algoritmos de escalonamento, classes e filtros a depender das necessidades de QoS requeridas pelas aplicações. Como exemplo, pode-se combinar filtros e algoritmos de escalonamento para construir um serviço de classe de encaminhamento expresso (EF) do *DiffServ* (HUBERT, 2001).

A estrutura implantada para a rede protótipo experimental foi composta de 03 PCs Dell P4 2,26GHz 512MB com GNU/Linux instalado, configurados como roteadores. Cada PC possui 03 interfaces de rede *ethernet*; 1 comutador (s*witch*) 24 portas 10/100 3Com com 5

VLANs (*virtual LANs*) configuradas e 2 PCs utilizados para geração de tráfego e realização de medições associadas às campanhas de teste (OLIVEIRA, 2006).

A Figura 5-1 ilustra a topologia de rede utilizada na rede protótipo experimental.

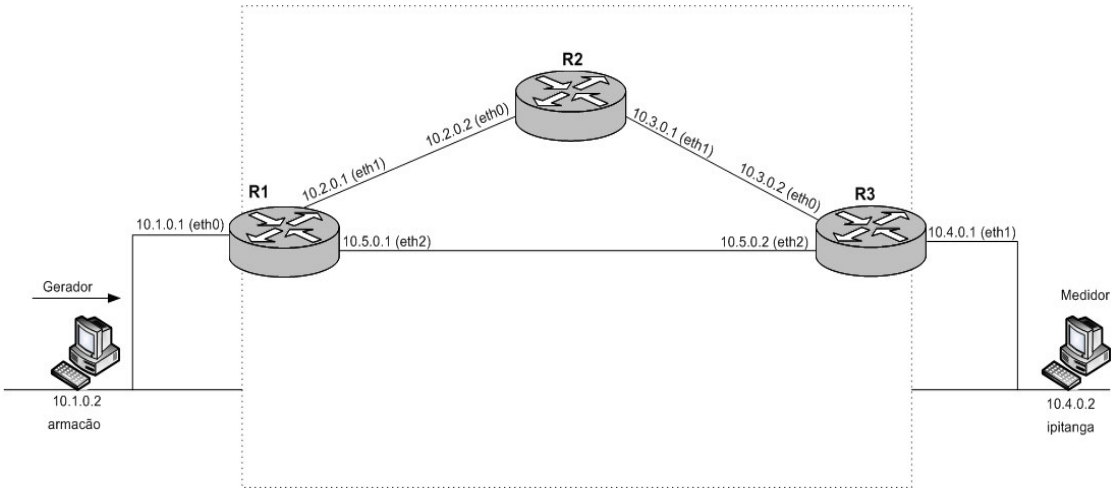


Figura 5-1: Topologia da rede protótipo experimental (OLIVEIRA, 2006).

A configuração da rede protótipo experimental permite um conjunto básico de configurações de campanhas de teste com geração de tráfego pela rota R1-R2-R3 ou pela rota direta R1-R3. Um hospedeiro atua como gerador de tráfego (configurável) e o outro atua na medição de tráfego.

A configuração de VLANs no comutador permite criar sub-redes independentes, reduzindo o domínio de difusão na camada *Media Access Control* (MAC) e, forçando a utilização dos roteadores para o encaminhamento dos pacotes entres as redes lógicas criadas.

Em cada roteador foi configurada uma tabela de rota de forma estática através de do comando *route* do GNU/Linux. Observe-se que uma rota padrão é configurada para permitir encaminhamento de pacotes cujo endereço de destino do pacote não pertença a nenhuma das rotas configuradas. Nos Quadros 5-1, 5-2 e 5-3 são apresentados os roteiros básicos de configuração de rotas para cada roteador usado na rede protótipo experimental (OLIVEIRA, 2006):

```
#deleta possíveis rotas existentes
route del –net 10.1.0.0 netmask 255.255.0.0
route del –net 127.0.0.0 netmask 255.0.0.0
route del –net 0.0.0.0 netmask 0.0.0.0
route del –net 169.254.0.0 netmask 255.255.0.0
#adiciona novas rotas
#adiciona rota 10.1.0.0/16 associada à interface eth0
route add –net 10.1.0.0 netmask 255.255.0.0 eth0
#adiciona rota 10.2.0.0/16 associada à interface eth1
route add –net 10.2.0.0 netmask 255.255.0.0 eth1
#define rota padrão
route add default gw 10.2.0.2
```

Quadro 5-1 Roteiro de configuração de rotas inicial do roteador R1 (10.1.0.1, 10.2.0.1, 10.5.0.1) (OLIVEIRA, 2006)

```
#deleta possíveis rotas existentes
route del –net 10.2.0.0 netmask 255.255.0.0
route del –net 10.2.0.0 netmask 255.255.0.0
route del –net 10.3.0.0 netmask 255.255.0.0
route del –net 10.3.0.0 netmask 255.255.0.0
route del –net 169.254.0.0 netmask 255.255.0.0
route del –net 127.0.0.0 netmask 255.0.0.0
#adiciona novas rotas
route add –net 10.3.0.0 netmask 255.255.0.0 eth1
route add –net 10.2.0.0 netmask 255.255.0.0 eth0
route add –net 10.1.0.0 netmask 255.255.0.0 gw 10.2.0.1
route add default gw 10.3.0.2
```

Quadro 5-2 Roteiro de configuração de rotas inicial do roteador R2 (10.2.0.2, 10.3.0.1) (OLIVEIRA, 2006)

```
#deleta possíveis rotas existentes
route del –net 10.4.0.0 netmask 255.255.0.0
route del –net 10.3.0.0 netmask 255.255.0.0
route del –net 169.254.0.0 netmask 255.255.0.0
route del –net 127.0.0.0 netmask 255.0.0.0
route del –net 0.0.0.0 netmask 0.0.0.0
#adiciona novas rotas
route add –net 10.3.0.0 netmask 255.255.0.0 eth0
route add –net 10.4.0.0 netmask 255.255.0.0 eth1
route add –net 10.1.0.0 netmask 255.255.0.0 gw 10.3.0.1
route add –net 10.2.0.0 netmask 255.255.0.0 gw 10.3.0.1
route add –net 10.3.0.0 netmask 255.255.0.0 gw 10.3.0.1
route add default gw 10.4.0.1
```

Quadro 5-3 Roteiro de configuração de rotas inicial do roteador R3 (10.3.0.2, 10.4.0.1, 10.5.0.2) (OLIVEIRA, 2006)

5.2 CENÁRIOS DE IMPLANTAÇÃO

A especificação e implantação da rede protótipo experimental tem o objetivo de validar as especificações de QoS das aplicações do projeto InfraVIDA, explicitadas nos Quadros 4-1, 4-2 e 4-3.

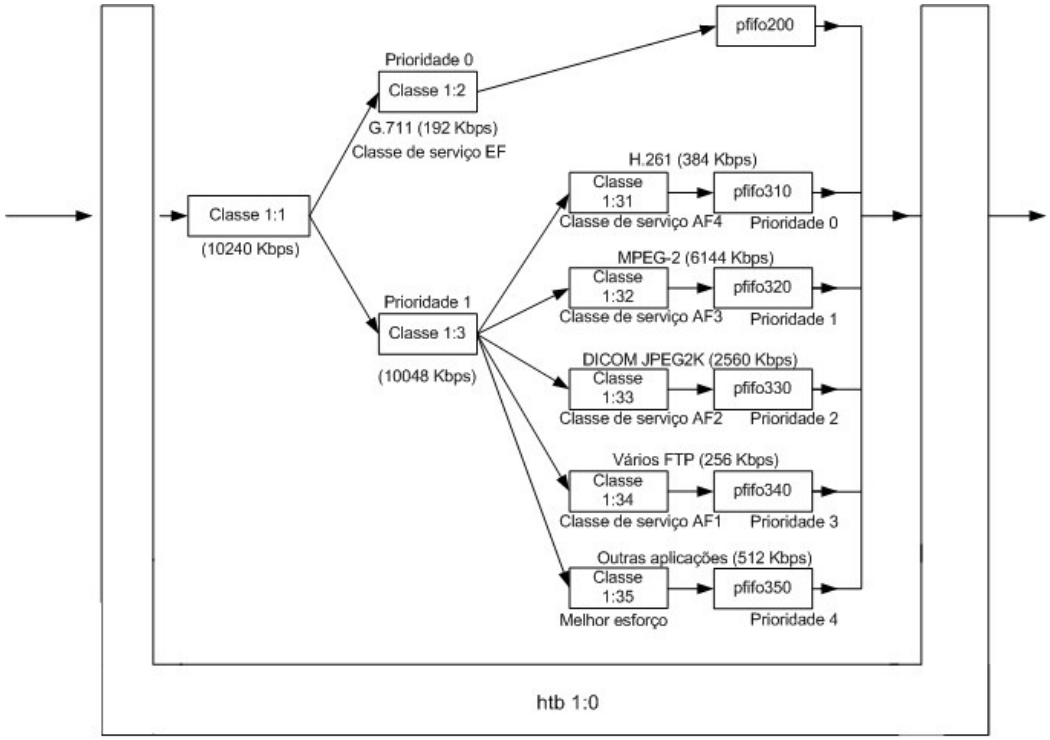


Figura 5-2: Algoritmo de escalonamento para o cenário 1 (OLIVEIRA, 2006)

Para cada uma das configurações de teste especificadas na tabela 4-3, foram criados dois tipos de roteiros com algoritmos de escalonamento diferentes para os roteadores no GNU/Linux utilizando a ferramenta *tc* do pacote *iproute2*. A Figura 5-2 ilustra ambos os algoritmos de escalonamento para o Cenário 1 da tabela 4-3.

O primeiro roteiro usa o algoritmo de escalonamento *Hierachical Token Bucket* (HTB), que implementa uma estrutura hierárquica de baldes de fichas (*token buckets*). O outro roteiro utiliza o algoritmo de escalonamento *Class-Based Queueing* (CBQ). Ambos os roteiros têm o mesmo propósito, que é atender aos requisitos de QoS das aplicações do projeto InfraVIDA, mas usam algoritmos de escalonamento diferentes para a mesma tarefa (OLIVEIRA, 2006).

A seguir é apresentado como exemplo o roteiro para a configuração do roteador R1, para o cenário 1 usando o algoritmo de escalonamento CBQ.

O roteiro cria uma disciplina de fila básica (*tc qdisc cbq*) que especifica o algoritmo CBQ e a banda de 10 Mbps a ser distribuída entre as classes de tráfego. Subordinadas hierarquicamente a esta fila são criadas disciplinas de fila para cada uma das classes *DiffServ* (*tc class*) com respectivas prioridades, bandas (e tamanhos médios de pacotes estimados).

A classe 1:1 tem prioridade máxima em relação às outras e é usada para a classe de serviço *DiffServ* EF mapeada para a aplicação de videoconferência, com banda mínima de 240 kbps. Ou seja, oferece garantia de largura de banda para 3 sessões G.711 de 80 kbps.

Foram criadas ainda mais 5 classes 1:2 mapeadas para as classes de serviço do *DiffServ* AF4, AF3, AF2, AF1 e melhor esforço respectivamente, às quais foram atribuídas porções da banda, de acordo com a tabela 4-3. A classe com maior prioridade é a classe 1:31 (AF4) e a de menor prioridade é a classe 1:35 (melhor esforço). Associa-se um buffer com tamanho de 10 pacotes para cada fila (*tc qdisc pfifo*). Por fim são especificados filtros (*tc filter*) que reconhecem o valor de DSCP nos pacotes e os direcionam para as devidas filas. O Quadro 5-4 ilustra o roteiro de configuração do roteador R1 usando CBQ.

```
#cria disciplinas de fila cbq
tc qdisc add dev eth1 root handle 1: cbq bandwidth 10Mbit avpkt 1000

#cria disciplinas de fila para cada uma das classes DiffServ
#EF
tc class add dev eth1 parent 1: classid 1:1 cbq bandwidth 10Mbit rate
240kbit avpkt 252 prio 1 bounded isolated allot 1514
#classe dos AF
tc class add dev eth1 parent 1: classid 1:2 cbq bandwidth 10Mbit rate
10048kbit prio 2 bounded isolated allot 1514
#AF4
tc class add dev eth1 parent 1:2 classid 1:21 cbq bandwidth 10Mbit rate
384kbit avpkt 1040 prio 1 bounded isolated allot 1514
#AF3
tc class add dev eth1 parent 1:2 classid 1:22 cbq bandwidth 10Mbit rate
4096kbit avpkt 1040 prio 2 bounded isolated allot 1514
#AF2
tc class add dev eth1 parent 1:2 classid 1:23 cbq bandwidth 10Mbit rate
2560kbit avpkt 1040 prio 3 bounded isolated allot 1514
#AF1
tc class add dev eth1 parent 1:2 classid 1:24 cbq bandwidth 10Mbit rate
512kbit avpkt 500 prio 4 bounded isolated allot 1514
#BE
tc class add dev eth1 parent 1:2 classid 1:25 cbq bandwidth 10Mbit rate
512kbit prio 5 bounded isolated allot 1514

#adiciona buffer para 10 pacotes para cada disciplina de fila criada
tc qdisc add dev eth1 parent 1:1 handle 200: pfifo limit 10
tc qdisc add dev eth1 parent 1:21 handle 310: pfifo limit 10
tc qdisc add dev eth1 parent 1:22 handle 320: pfifo limit 10
tc qdisc add dev eth1 parent 1:23 handle 330: pfifo limit 10
tc qdisc add dev eth1 parent 1:24 handle 340: pfifo limit 10
tc qdisc add dev eth1 parent 1:25 handle 350: pfifo limit 10
```

```
#cria os filtros de pacote
tc filter add dev eth1 parent 1:0 protocol ip prio 1 u32 match ip tos 0xb8
0xff flowid 1:1
tc filter add dev eth1 parent 1:0 protocol ip prio 1 u32 match ip tos 0x88
0xff flowid 1:21
tc filter add dev eth1 parent 1:0 protocol ip prio 1 u32 match ip tos 0x68
0xff flowid 1:22
tc filter add dev eth1 parent 1:0 protocol ip prio 1 u32 match ip tos 0x48
0xff flowid 1:23
tc filter add dev eth1 parent 1:0 protocol ip prio 1 u32 match ip tos 0x28
0xff flowid 1:24
tc filter add dev eth1 parent 1:0 protocol ip prio 1 u32 match ip tos 0x00
0xff flowid 1:25
```

Quadro 5-4: Roteiro de configuração de CBQ no roteador R1 (OLIVEIRA, 2006)

Para gerar o tráfego, de forma a simular os fluxos das aplicações especificados na tabela 4-3 do projeto InfraVIDA, foi usada uma ferramenta de medição ativa denominada *Real-time UDP Data Emitter* (RUDE) & *Collector for Real-time UDP Data Emitter* (CRUDE) (LAINE, J. ; SARITO, S. ; PRIOR, R., 2002). Essa ferramenta é composta de dois executáveis. Um deles, o RUDE, é utilizado para gerar o tráfego de forma totalmente configurável através de um arquivo de configuração. Pode-se montar um cenário de geração com diversos fluxos, configurando o destino desses fluxos, a taxa de geração, o código DSCP marcado no cabeçalho do pacote etc. O outro programa da ferramenta, o CRUDE, é executado na máquina de destino e serve para medir todo o tráfego gerado pelo RUDE. O CRUDE coleta e registra todo o tráfego transmitido pelo RUDE. Ele possui diversas opções, tais como medir métricas como vazão, atrasos, variação de atrasos e perdas referentes a apenas um fluxo do tráfego ou todos os fluxos.

O RUDE é instalado na ponta da rede fazendo o papel das aplicações no cenário da rede protótipo experimental. Todos os tráfegos foram gerados numa taxa constante com os tamanhos dos pacotes para cada fluxo especificados na tabela 4-3. Já o CRUDE é instalado no destino para a coleta das informações geradas pelo RUDE. A Figura 5-3 ilustra o posicionamento dessas ferramentas na rede protótipo experimental.

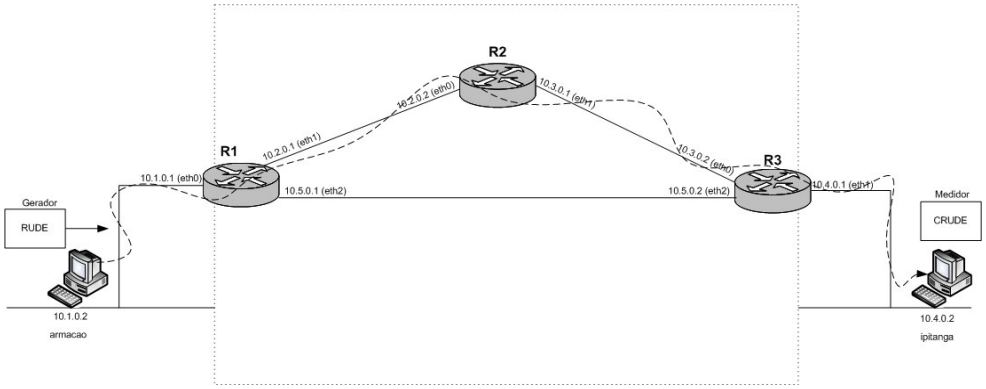


Figura 5-3: Posicionamento das ferramentas de geração e medição RUDE/CRUDE (OLIVEIRA, 2006)

As medições foram feitas considerando os parâmetros de QoS: vazão, atraso, variação de atraso (*jitter*) e perda.

Para medir o atraso na transmissão de um pacote de uma máquina para outra, é necessário que os relógios das mesmas estejam sincronizados, caso contrário, a medida do atraso não é calculada corretamente. O problema da sincronização é resolvido através do protocolo *Network Time Protocol* (NTP) (MILLS, 1992), que se baseia em um modelo de sincronização hierárquico. Para isto utiliza-se um relógio de alta precisão ou um *Global Position Sattelite* (GPS), chamado de *stratum 0*. Utiliza-se então um servidor, chamado de *stratum 1*, por estar ligado ao dispositivo *stratum 0*. É o servidor que vai atender às solicitações de sincronização das máquinas usadas na rede protótipo experimental, chamadas então de *stratum* 2. Com esta configuração o erro na medição do atraso é de aproximadamente 30 μs.

Na campanha dos testes realizada é feita uma comparação entre o desempenho de ambos os algoritmos com relação às aplicações. Foi adicionado tráfego de *stress* até a capacidade dos enlaces para melhor avaliar o impacto nos mecanismos de QoS. Analisando os resultados obtidos (OLIVEIRA, 2006), conclui-se que o algoritmo de escalonamento CBQ aplicado aos roteadores se mostrou o mais adequado em relação aos requisitos de QoS das aplicações do projeto InfraVIDA. Todos os parâmetros medidos utilizando essa configuração obtiveram resultados proveitosos, validando adequadamente os parâmetros de QoS especificados pelo projeto.

No entanto, a configuração de QoS utilizando o algoritmo de escalonamento HTB não atendeu adequadamente os requisitos de QoS das aplicações do projeto InfraVIDA. Alguns fluxos se comportaram de forma inesperada, particularmente fluxos que possuem um pequeno tamanho do pacote (252 *bytes*), característico do formato do *codec* de áudio G.711. Para

fluxos com essa característica foram observadas perdas além do esperado, alto atraso, além de alta variação de atraso (*jitter*).

5.3 PROTOTIPAÇÃO DO *BANDWIDTH BROKER*

O BB é um servidor com as seguintes funcionalidades:

- aceita conexões TCP iniciadas por clientes para o envio de requisições de recursos (RARs);
- faz o controle de admissão via autenticação simples;
- processa as requisições de recursos de clientes (RARs) e toma as decisões requeridas, baseando-se em informações dos contratos de serviço com os clientes (SLAs), obtidas no repositório de dados;
- envia uma resposta ao cliente indicando sucesso ou falha na alocação de recursos;
- aceita conexões dos roteadores para envio de configuração inicial dos mecanismos de escalonamento e de inclusão de rotas estáticas na tabela de rotas, opcionalmente através do protocolo COPS-PR;
- conecta-se e aceita conexões encaminhadas por BBs em domínios adjacentes opcionalmente através do protocolo SIBBS (*Simple Inter-Domain Bandwidth Broker Protocol*) de modo a processar requisições inter-domínios.

Existem diversas iniciativas de construção de BBs para ambientes *DiffServ* (SOHAIL, S.; JHA. S., 2002). Consideramos, para efeito da avaliação e prototipação, a implementação de um BB em Java (PHAM, K. B.; NGUYEN, R., 2003), baseado na arquitetura proposta pelo *QBone Signaling Design Team* da Internet2 e em desenvolvimento preliminar de código COPS pela Telia, estendido para uso do protocolo COPS-PR na Universidade New South Wales (HALIM, 2000).

Esta implementação do BB tira partido dos avanços em programação remota oferecida pelo *Java* [JAVA], uma linguagem de programação orientada a objeto, independente de plataforma, que possui um grande número de classes pré-definidas e facilidade de integração a banco de dados *MySQL* [MySQL]. O *Java* oferece grande facilidade de programação cliente/servidor remota através do uso de *TCP Sockets* e de classes *Sockets* parte de sua biblioteca padrão. O *Java* é orientado ao desenvolvimento de aplicações web, oferecendo suporte às especificações do XML [XML], uma linguagem de formatação de objetos de dados, que faz uso de *tags* pré-definidas para tornar acessível o seu

conteúdo e do SOAP [SOAP], um esquema de codificação de mensagens e protocolo para comunicação cliente/servidor remota. O uso do *Java Web Services Developer Pack* [JWSD] do Apache facilita o desenvolvimento de aplicações web. O JWSD exige o uso de *Java* no servidor e de *Java Virtual Machines* nos clientes que acessam aplicações web baseadas em Java.

A funcionalidade do BB segue o modelo cliente/servidor onde o BB responde a requisições de um cliente. A interação do BB com os roteadores também segue o modelo cliente/servidor. Os roteadores conectados ao BB recebem informação de configuração do BB, sendo a comunicação cliente/servidor oferecida pelo *Java* um facilitador ao seu funcionamento.

Os roteadores do domínio *DiffServ* são implementados usando máquinas Linux da rede protótipo experimental (no caso, em um único domínio). As configurações iniciais dos roteadores são feitas manualmente utilizando-se os roteiros mencionados na seção anterior. O BB poderia configurar opcionalmente os roteadores do seu próprio domínio usando o protocolo COPS-PR. A implementação suporta ainda comunicação entre BB de domínios *DiffServ* vizinhos, cujos endereços são acessíveis na base de dados do *MySQL*, mas esta funcionalidade não foi testada na rede protótipo experimental, já que implantamos um único domínio.

5.3.1 Repositório de Dados

A operação do BB depende do repositório de dados para processar adequadamente as requisições dos clientes. Nesta implantação, o repositório de dados faz uso de servidor *MySQL* e de diversas tabelas que apoiam o funcionamento do BB, descrevendo informações de clientes, do BB e da rede. As tabelas do banco de dados relevantes para a operação em um único domínio são descritas a seguir:

Nome do campo	**Formato**	**Descrição**
sla_id(PRIMARY KEY)	inteiro	Número do contrato ou SLA
service_type	alfanumérico	Classe de serviço *DiffServ* (EF, AF ou BE)
Startdate	data	Data do inicio do SLA
Starttime	hora	Hora do inicio do SLA
Enddate	data	Data do fim do SLA
Endtime	hora	Hora do fim do SLA
Rate	inteiro	Banda total alocada ao SLA (kbps)
AvailBW	inteiro	Banda ainda disponível para SLA (kbps)

Tabela 5-1 tabela SLA (AUTOR, 2008)

A Tabela 5-1 descreve os aspectos técnicos do contrato de serviço com o cliente (SLS). O *sla_id* identifica o contrato do cliente (SLA) e é associado uma classe de serviço *DiffServ*.

A Tabela 5-2 guarda as condições das requisições de alocação de recursos (RARs) aceitas pelo BB. Os campos da tabela RAR refletem os da tabela SLA, já que as condições das RARs dependem das especificações do SLA. A tabela contém os endereços de origem e destino dos fluxos cujas RARs foram aceitas.

Nome do campo	**Formato**	**Descrição**
rar_id(PRIMARY KEY)	inteiro	Número da RAR incrementado automaticamente
Startdate	data	Data do inicio do SLA
Starttime	hora	Hora do inicio do SLA
Enddate	data	Data do fim do SLA
Endtime	hora	Hora do fim do SLA
GivenBW	inteiro	Banda alocada ao RAR (kbps)
Source	alfanumérico	Endereço IP de origem no formato (w.x.y.z)
Destination	alfanumérico	Endereço IP de destino no formato (w.x.y.z)
sla_id	inteiro	Referência ao número do contrato ou SLA

Tabela 5-2 tabela RAR

A Tabela 5-3 contém as senhas associadas a cada SLA de cliente, para autenticação pelo BB.

Nome do campo	**Formato**	**Descrição**
sla_id(PRIMARY KEY)	inteiro	Número do contrato ou SLA
Password	alfanumérico	Senha associada ao contrato ou SLA

Tabela 5-3 tabela *passwords*

A Tabela 5-4 mapeia a classe de serviço associada a cada SLA de cliente, a um valor de DSCP usado para identificar o tipo de tratamento oferecido aos pacotes em uma rede *DiffServ*.

Nome do campo	**Formato**	**Descrição**
service_type (PRIMARY KEY)	alfanumérico	Classe de serviço *DiffServ* (EF, AF ou BE)
Dscp	alfanumérico	Campo de 6 *bits* correspondente ao DSCP daquela classe de serviço *DiffServ*

Tabela 5-4 tabela *codepoint*

O BB precisa saber qual a banda total e a banda ainda disponível na rede de um domínio. A premissa assumida aqui é que a banda total é a do enlace de menor capacidade na rede, por ser ele o gargalo onde pode ocorrer perda de pacotes que geram retransmissão. A banda total e a banda disponível de cada domínio constam da Tabela 5-5.

Nome do campo	Formato	Descrição
Domain (PRIMARY KEY)	alfanumérico	Identificador do domínio no formato (w.x.y.z)
Capacity	Inteiro	Capacidade total oferecida pelo domínio (kbps)
availCapacity	Inteiro	Capacidade ainda disponível no domínio (kbps)

Tabela 5-5 tabela *capacity*

A Tabela **5-6** mantém o registro de todos os fluxos válidos ativos. O BB faz uso dessa tabela para identificar quando o nível de recursos associado a um determinado RAR precisa ser alterado.

Nome do campo	Formato	Descrição
RAR (PRIMARY KEY)	Inteiro	Referência ao rar_id na tabela RAR
Domain	alfanumérico	Identificação de um dominio *DiffServ* (w.x.y.0 ou w.y.0.0)
Bandwidth	Inteiro	Banda alocada ao RAR (*kbps*)
Startdate	Data	Data do inicio do RAR
Starttime	Hora	Hora do inicio do RAR
Enddate	Data	Data do fim do RAR
Endtime	Hora	Hora do fim do RAR

Tabela 5-6 tabela *flows*

O BB tem o papel de gerenciador de recursos e servidor de políticas (PDP) e precisa conhecer todos os roteadores configuráveis (PEPs) no seu domínio, para enviar-lhes informações de configuração.

Nome do campo	Formato	Descrição
pepID (PRIMARY KEY)	alfanumérico	Identificação do PEP (roteador configurável)
Sindex	Inteiro	Valor inteiro (index) do TCP Socket correspondente à comunicação com este PEP
Domain	alfanumérico	Identificação do dominio ao qual o PEP pertence (w.x.y.0 ou w.y.0.0)
Address	alfanumérico	Endereço IP do PEP (w.x.y.z)
Neighbour	alfanumérico	Identificação do dominio adjacente ao qual o PEP pertence (w.x.y.0 ou w.y.0.0) [PEP é roteador de borda]
Status	Binário	Status (‘ON’ ou ‘OFF’)

Tabela 5-7 tabela PEP

Os PEPs devem ter uma identificação única em toda a rede, independente de domínio. A Tabela **5-7** mantém o registro de todos os **PEPs**. O campo ***sindex*** é requerido porque uma lista dos PEPs para um dado domínio é arquivado em um *array* nesta implementação de BB.

O acesso ao banco de dados *MySQL* é viabilizado com o uso de *Java Database Connectivity* (JDBC). É feita uma instanciação da classe *Connection* com a localização e o nome do banco de dados, para permitir qualquer acesso subseqüente ao banco de dados. O método Java que faz uma *query* no banco SQL é o *executeQuery("select .....")* tendo uma *query* SQL como parâmetro do método. O banco de dados retorna a resposta ao Java como uma classe *ResultSet*, que contém métodos para recuperar elementos individuais do resultado. Os métodos *getInt()* e *getString()* são usados com argumentos que são nomes dos campos retornados pela *query* SQL.

5.3.2 Mensagens entre o Cliente e o Servidor *Bandwidth Broker*

O BB é inicializado com a classe *BBServ* e aceita conexões de clientes através de *sockets TCPs.* O BB inicia uma nova instanciação do processo (*BBMultiServerThread*) a cada nova conexão de cliente. Para permitir a troca de informações entre o BB e o cliente, são criados canais distintos de dados de entrada e saída. Os dados são transferidos entre o servidor BB e a aplicação cliente, uma linha de cada vez. O método permite transferir um pequeno volume de informações e reduz o tráfego de informações simultâneas na rede.

Esta implementação de BB em *Java* fornece um cliente *BBClient* para interação com o servidor BB, que suporta as mensagens cliente/servidor. O formato textual das mensagens implementadas é apresentado no Quadro 5-5.

Msg	**Função**	**Formato e Exemplo**
Request Bandwidth	requisição de alocação de recursos	"request bw;sla;startdate;starttime;enddate;endtime;bw;src;dst" *"request bw;1;2003-05-10;00:00:01;2003-06-30;23:59:59;1000;129.94.231.23;129.94.232.41"*
SLA Info	informa identificador de SLA	"SLA info:sla" *"SLA:1"*
RAR Info	informa identificação de RAR associado ao SLA	"RAR INFO;sla;rar" *"RAR INFO;1;33"*
Delete RAR	deleta RAR	"delete RAR;sla;rar" *"delete RAR;1;33"*
Modify RAR	modifica condições de RAR	"modify RAR;sla;rar;startdate;starttime;enddate;endtime;bw" *"modify RAR;1;2003-05-11;10:00:00;2003-06-30;10:00:00;1500"*
Modify SLA	modifica especificações de SLA	"modify SLA;sla;service_type;total_bw;avail_bw;startdate;starttime;enddate;endtime"

		"modify SLA;1;EF;10000;9000;2003-04-25;09:00:00;2003-12-31;23:59:59"
Exit	fecha a conexão com o cliente	"exit
Shutdown	derruba o servidor BB	"shutdown"

Quadro 5-5 Mensagens da aplicação cliente-servidor do BB em Java

5.3.3 Comunicação entre os Roteadores e o *Bandwidth Broker*

Esta implantação de BB em Java (PHAM, K. B.; NGUYEN, R., 2003) usa uma implementação do COPS/COPS-PR [HALIM, 2000] para a comunicação entre um PDP (o BB) e diversos PEPs (roteadores Linux).

Uma questão de integração do código COPS ao BB, é que o COPS usa um *Policy Information Base* (PIB) para armazenar os dados e o BB usa um banco de dados *MySQL.* Para enviar mensagens COPS para um PEP, a classe *IpFilterEntry* é instanciada para prover os valores relevantes para o PIB. A classe *RARcops* é chamada quando uma requisição de recursos ou de eliminação de RAR é aceita pelo servidor BB e inicializa o procedimento de envio de mensagens com decisões COPS para o PEP. Os valores a serem enviados são recuperados do banco de dados *MySQL* e passados para a classe *IpFilterEntry* que os guarda no PIB antes de enviar mensagens COPS para o roteador.

Os roteadores Linux agem como PEPs de acordo com o protocolo COPS-PR. Quando é inicializado, o roteador requer que o BB supra a informação de configuração para que ele saiba como tratar os pacotes recebidos. A classe *PEPClient* é usada nesta implementação na aplicação cliente no roteador Linux . Uma instância dessa classe deve ser executada em cada máquina Linux configurada como um roteador.

Os comandos *tc* do Linux especificados em seção anterior poderiam ser executados para configurar cada roteador *DiffServ*. O código no arquivo *LinuxRoute.java*, em particular o método *setupDiff()* executam os comandos requeridos. O Java oferece acesso ao *shell* para execução dos comandos através do uso da classe *Runtime.* O comando *string diffserv* abaixo é executado da mesma maneira que se digitado no *shell* de comando do Linux.

```
String diffserv = "tc qdisc add dev eth0 handle 1:0 root dsmark indices 64 set_tc_index";
Runtime.getRuntime().exec(diffserv);
```

Usando a classe *Runtime*, o *PEPClient* pode acessar a tabela de roteamento ou a configuração de controle de tráfego requerida para a operação do *DiffServ*. O *PEPClient* vai chamar a classe *LinuxRoute* sempre que precisar fazer mudanças na tabela de roteamento, ou seja, instalar ou apagar uma rota.

O *PEPClient* faz uso do pacote COPS/COPS-PR, que fornece a classe *CopsprPepImpl*, usada para enviar e receber mensagens COPS para o BB que age como PDP. Quando o *PEPClient* é inicializado para operação *DiffServ*, ele tenta contatar o PDP que gerencia o seu domínio. Ele acessa o banco de dados *MySQL* para descobrir o endereço e porta do PDP, e estabelece uma conexão TCP usando estes parâmetros. Se o PDP aceita a conexão do PEP, o PDP tem a opção de enviar ao PEP toda a configuração inicial através de uma mensagem COPS *Decision*. Senão o PDP vai enviar mensagens *Decision* não-solicitadas ao PEP sempre que o BB achar necessário. Na inicialização, o *PEPClient* atualiza a base de dados de modo que o PDP saiba que ele está ativo e disponível para receber dados de configuração.

O método *processDEC()* do *PEPClient* extrai as decisões da mensagem *Decision* que chega ao PEP e lida com elas de acordo com o código de comando fornecido pela mensagem. O parâmetro *commandCode* determina se o PEP vai executar um comando de adição ou deleção da classe *LinuxRoute* de modo a alterar a tabela de roteamento. A ação e os parâmetros relevantes são então passados ao método *LinuxRoute* correspondente baseado na ação requerida pelo BB. Como o pacote COPS codifica os dados usando BER como uma seqüência de *bytes*, os valores de uma mensagem COPS *Decision* devem ser convertidos de volta a um formato utilizável. Um requisito particular é a criação de endereços IP no formato w.x.y.z.

No entanto, a única mensagem COPS usada dinamicamente por essa implementação de BB é a *Decision* e o uso da mensagem *Decision* no código oferece apenas a possibilidade de adicionar ou eliminar uma rota na tabela de roteamento de um roteador Linux.

5.3.4 Experimentação do *Bandwidth Broker* na Rede Protótipo

A implantação do BB em Java na rede protótipo experimental teve o objetivo de validar o tratamento de requisições de recursos de um fluxo entrante, admiti-lo e implantar uma alteração de configuração no roteador de borda, resultante de uma decisão do BB. Esta alteração de configuração é a implantação de uma rota estática na tabela de rotas.

Como visto, a aplicação cliente no roteador Linux complementa a aplicação servidor no BB no provimento da funcionalidade de gerenciamento por políticas de QoS *DiffServ*. A implantação do protocolo COPS-PR permite que o BB (PDP) envie informações de configuração ao roteador (PEP).

Para a implantação do BB em *Java* é necessária a instalação de *Java Virtual Machine* nos PCs da rede protótipo experimental, tanto no hospedeiro gerador de tráfego, quanto nos roteadores. O *PePClient* é inicializado nos roteadores, que podem ser customizados através da classe *LinuxRoute* para executar qualquer ação de controle de tráfego ou roteamento, que reflita uma decisão de alocação de recursos feita pelo BB.

Um teste simples permitiu validar uma mudança de configuração no roteador, em função de uma decisão do BB. Considere a Figura 5-1. O hospedeiro 10.1.0.2, no qual está instalado o gerador de tráfego, gera um tráfego equivalente a um *ping* destinado à interface eth2 do roteador R2, que agora é configurada com o endereço IP 10.6.0.1/16.

Para que o *ping* funcione, é necessária a instalação de uma nova rota no roteador R1. Para que essa rota seja obtida automaticamente por R1, o seguinte procedimento é verificado:

- o hospedeiro gerador de tráfego, onde está instalado o pacote *Java Virtual Machine* e o cliente *Java* fornecido na implantação do BB em *Java*, envia uma mensagem *request bw* com uma requisição de alocação de recursos (RAR) ao BB;

- o BB acessa seu repositório de dados, valida SLA e senha consultando as tabelas **SLA** e ***password*** e decide admitir o envio do novo fluxo pela aplicação, verificando a banda total e disponível na tabela **SLA**. O RAR aceito é adicionado à tabela **RAR** e à tabela ***flows*** (onde é mantido, enquanto ativo), enquanto o valor da banda disponível é atualizado na tabela **SLA**;

- O DSCP associado ao SLA é obtido pelo BB consultando a tabela ***codepoint***. O BB obtém a informação dos PEPs em seu domínio consultando a tabela **PEP**. Pode então consultar a configuração de controle de tráfego dos roteadores no repositório de dados para verificar se existem mecanismos de tratamento de tráfego para a CoS associada ao DSCP. Também pode informar à aplicação cliente, qual o DSCP a ser usado para marcar os pacotes do fluxo associado à RAR aceita;

- a informação de endereço IP de destino obtida na mensagem *request bw* e validada na tabela **RAR** é usada pelo BB para enviar uma mensagem *Decision* do COPS-PR com a configuração de nova rota a ser adicionada na tabela de roteamento do roteador R1. Para determinar automaticamente os parâmetros de configuração de novas rotas, seria preciso que o BB conhecesse toda a topologia da rede, e executasse, ele próprio, um protocolo de

roteamento como o OSPF. No teste realizado, o comando *tc* para adição da nova rota é configurado estaticamente no código;

- o roteador R1 recebe a mensagem *Decision* do COPS-PR, obtém o comando *tc* com a configuração da nova rota e executa o *Runtime* para adicioná-la à sua tabela de roteamento. No caso a rota 10.6.0.0/16 associada à interface eth1 é adicionada.

Roteador R1 (10.1.0.1, 10.2.0.1, 10.5.0.1)

```
route add –net 10.6.0.0 netmask 255.255.0.0 eth1
```

Como a rota padrão configurada usa o mesmo caminho que a nova rota configurada, a rota padrão do roteador R1 é redefinida para permitir validar o funcionamento da nova rota adicionada.

```
route add default gw 10.5.0.2
```

O *ping* é então realizado com sucesso, validando o procedimento como um todo.

O uso da mensagem *Decision* nesta implementação do BB em Java oferece apenas a possibilidade de adicionar ou retirar uma rota na tabela de roteamento de um roteador Linux. Para permitir a obtenção da configuração inicial do roteador automaticamente a partir do BB, o código precisaria ser estendido para permitir o envio da configuração do roteador, linha a linha, um comando *tc* de cada vez. A configuração inicial completa inclui a tabela de rotas inicial e a configuração dos mecanismos de escalonamento. Os comandos seriam implantados no roteador usando *Runtime*, como descrito anteriormente. No entanto, esta tarefa eventual pode ser executada manualmente por meio de roteiros, como fizemos na rede protótipo experimental.

CAPÍTULO 6 – TENDÊNCIAS EM GERENCIAMENTO DE POLÍTICAS DE QUALIDADE DE SERVIÇO EM REDES IP

O capítulo 6 apresenta a tendência de uso dos protocolos SOAP e NETCONF no campo de gerenciamento baseado em políticas e explicita especificamente o mapeamento de mensagens do protocolo COPS-PR no protocolo SOAP.

6.1 NOVOS PROTOCOLOS EM GERENCIAMENTO DE SERVIÇOS DE REDE

O protocolo COPS-PR foi definido pelo IETF dentro da arquitetura de gerenciamento de redes baseado em políticas *Policy-Based Network Management* (PBNM) para o provisionamento dinâmico de políticas, como anteriormente descrito na seção 2.3. Apesar do COPS-PR já ter suporte em dispositivos de mercado, existe uma tendência na indústria e no IETF em avaliar os protocolos NETCONF e SOAP como substitutos ao COPS-PR na área de gerenciamento de redes baseado em políticas (FRANCO et al., 2006).

À luz desta tendência e de forma a poder antecipar uma evolução da arquitetura de QoS proposta e avaliada, julgamos pertinente, no escopo do trabalho desenvolvido, indicar a extensão e adaptação desta arquitetura ao uso de SOAP e NETCONF.

O protocolo SOAP [SOAP] é parte da arquitetura de WS (CURBERA et al., 200)]. Apesar de ter sido concebido para permitir a comunicação entre processos de aplicações na Web, o SOAP está sendo especificamente adaptado para o gerenciamento de serviços de rede pela indústria. O protocolo de configuração NETCONF [ENNS, 2006] foi padronizado pelo IETF como solução de gerenciamento de configuração de dispositivos, sendo possível que as mensagens NETCONF sejam transportadas em SOAP (GODDARD, 2006).

Apesar do SOAP e do NETCONF terem sido originalmente definidos fora do mundo PBNM, estes protocolos poderiam substituir o COPS-PR no provisionamento de políticas, com a vantagem de permitir a comunicação entre entidades clientes e servidores de políticas localizados em diferentes domínios administrativos, tendo a Internet como rede intermediária. Esta comunicação é possível porque ambos os protocolos SOAP e NETCONF podem ser transportados por protocolos de aplicação como o HTTP, SMTP e FTP, os quais são freqüentemente permitidos por *firewalls* na fronteira entre domínios administrativos e a Internet, diferentemente do tráfego COPS-PR, que dificilmente recebe o mesmo tratamento e acaba confinado a cada domínio. Em contrapartida, questões de

segurança com o uso do SOAP ou NETCONF em gerenciamento de serviços na Internet precisam ser endereçadas.

O protocolo COPS-PR usa esquema de codificação binária, enquanto o SOAP e o NETCONF tem suas mensagens codificadas em XML [XML], como documentos de texto. Isto implica em um impacto decisivo no desempenho da arquitetura de gerenciamento, afetando o consumo de banda e o tempo de resposta quando novas políticas precisam ser implementadas em redes gerenciadas.

Vamos nos concentrar aqui no uso do SOAP em gerenciamento de serviços de rede, por duas razões: por nos parecer mais promissor que o NETCONF; e por mostrar afinidade com a arquitetura de suporte de gerenciamento de QoS proposta no escopo desta dissertação, onde o SOAP/XML já é adotado na interface dinâmica com as aplicações, com a função de encaminhamento das requisições de solicitações de recursos (RARs) ao BB.

6.1.1 Serviços Web em Gerenciamento de Serviços de Rede

O uso do protocolo SOAP e da arquitetura WS para gerenciamento de serviços de rede tem atraído o interesse da academia e da indústria. Diversos pesquisadores estudaram o impacto do uso de WS em gerenciamento considerando, por exemplo, o consumo de banda (FIORESE et al., 2005), o tempo de resposta (PRAS et al., 2004) e a sua integração ao SNMP (KLIE; STRAUSS, 2004).

Dois consórcios industriais lideram a iniciativa de gerenciamento com o uso de WS: o *Organization for the Advancement of Structured Information Standards* (OASIS) explora não apenas o uso de WS para gerenciamento, como o gerenciamento de WS; um segundo grupo, formado pelas empresas AMD, BMC, CA, Dell, Fujitsu, Intel, Microsoft, NEC, Novell, Sun, Symantec e WBEM Solutions, criou uma nova especificação chamada *WS-Management* (ARORA et al., 2005), que define como mensagens SOAP devem ser codificadas quando usadas para fins de gerenciamento de serviços de rede.

WS podem ser vistos como componentes programáveis independentes disponíveis na Web e acessíveis por aplicações em rede ou outros serviços web. A comunicação entre WS é possível graças ao SOAP, um protocolo leve baseado em XML que segue o modelo cliente-servidor e emprega o paradigma de procedimento controlado a distancia *Remote Procedure Control* (RPC). Apesar de suportar outros formatos de codificação de mensagens, o protocolo SOAP é usado freqüentemente com o formato XML. As mensagens SOAP podem ser transportadas por protocolos de aplicação da arquitetura TCP/IP como

HTTP, SMTP e FTP. As operações e formatos de dados do protocolo SOAP são bem definidos, o que permite rápido desenvolvimento e adoção de WS como soluções de gerenciamento de serviços.

6.1.2 Mapeamento do Protocolo Cops-Pr no Protocolo Soap

Se considerarmos a possibilidade de usar o protocolo SOAP para substituir o COPS-PR no provisionamento dinâmico de políticas, no escopo de uma arquitetura de gerenciamento baseado em políticas, devemos considerar mapear o protocolo COPS-PR no protocolo SOAP.

No COPS-PR, um PDP deve comunicar suas decisões e invocar operações no PEP (por exemplo, DEC/<*edit-config*>), mas o PDP também deve ser capaz de receber notificações do PEP (por exemplo, REQ/<*event*>). Do mesmo modo que o COPS-PR, o SOAP segue o modelo cliente-servidor: um servidor Web oferece serviços a clientes remotos. Mas diferentemente do COPS-PR, a implementação de um par cliente/servidor no SOAP só possibilita a comunicação em uma direção: só é possível invocar operações ou receber notificações. É preciso implementar outro par cliente/servidor para suportar comunicações na direção oposta.

O SOAP é genérico e flexível e suporta modelos de comunicação alternativos ao RPC. Mas apesar de sua flexibilidade, clientes e servidores SOAP são entidades assimétricas, em termos de operações de WS: só os clientes podem invocar operações nos servidores. Um servidor não pode invocar uma operação no cliente. Se isto for necessário, o WS deve ser também implementado no lado do cliente, e clientes e servidores trocam de papel para invocar novas operações na direção oposta.

Notificações SOAP (STEVE, G. HULL, D.; MURRAY, B., 2005) são operações WS especiais para as quais as respostas não contém nenhum valor, geradas apenas por clientes SOAP para servidores SOAP. Portanto, não é possível ter clientes invocando operações nos servidores em uma direção e sendo notificados na direção oposta.

No mapeamento de COPS-PR para SOAP, o PDP SOAP implementa simultaneamente um servidor SOAP e um cliente SOAP (FRANCO et al., 2006). O servidor SOAP no PDP é capaz de receber notificações SOAP, enquanto o cliente SOAP no PDP é usado para disparar operações no PEP remoto. De modo complementar, um PEP SOAP também implementa um servidor SOAP e um cliente SOAP. O cliente SOAP do PEP é

usado quando o PEP precisa notificar o PDP remoto, enquanto o servidor SOAP do PEP expõe as operações do PEP disponíveis para um dado PDP.

A Figura 6-1 mostra o mapeamento das mensagens do COPS-PR em mensagens SOAP. Aqui por simplicidade, assume-se que o SOAP é transportado diretamente no TCP, apesar de ser mais freqüente e adequado à comunicação na Internet o transporte do SOAP no HTTP.

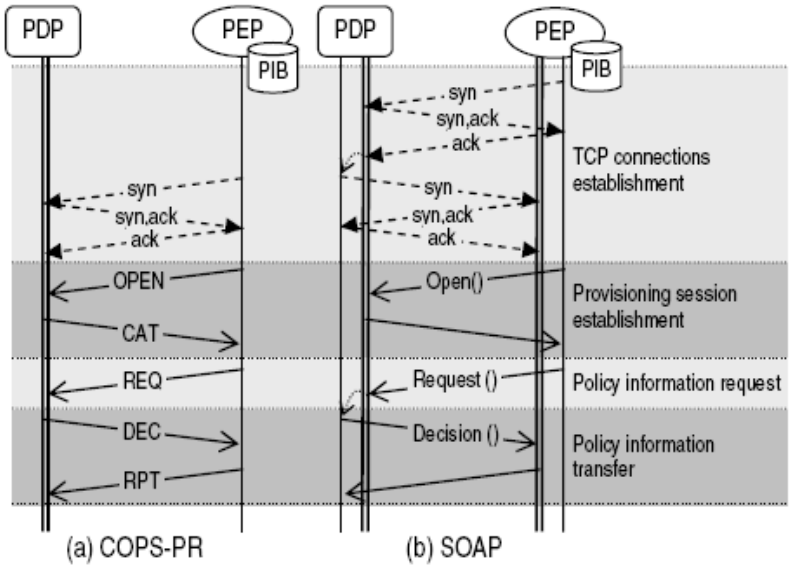


Figura 6-1: Mapeamento de mensagens COPS-PR em SOAP (FRANCO et al., 2006)

O gerenciamento do protocolo é mais complexo: por tratar com dois pares de cliente-servidor é preciso controlar duas conexões. Para tratar as duas conexões como uma única, define-se que se uma conexão é perdida, a outra deve ser imediatamente abortada. Portanto, se um cliente ou servidor SOAP de um lado (PEP ou PDP) detecta uma perda de conexão, o outro cliente ou servidor do mesmo lado deve ser internamente contactado para que derrube a outra conexão.

Os PEPs baseados em SOAP podem iniciar as sessões de provisionamento e PDPs baseados em SOAP podem gerenciar as re-conexões dos PEPs, do mesmo modo que no COPS-PR. Uma vez que as duas conexões cliente/servidor entre PEP e PDP sejam estabelecidas, o PEP invoca a operação *Open()* no PDP informando o seu tipo de cliente. A resposta à execução do *Open()* indica se a sessão de provisionamento foi aceita ou não. Se o tipo do PEP não for suportado pelo PDP, a sessão é negada.

Se o estado interno do PEP mudar, pode ser necessário obter novas políticas do PDP. Quando uma mudança de estado ocorre, o cliente SOAP no PEP notifica o servidor SOAP

no PDP invocando a operação de notificação *Request()*. Quando é notificado, o servidor SOAP no PDP contacta o cliente SOAP internamente. Se houver alguma nova política disponível para ser notificada ao PEP, ou se a política no PDP for alterada e requerer atualização do PIB no PEP, o cliente SOAP no PDP invoca a operação *Decision()* disponível no servidor SOAP do PEP. O resultado da execução da operação *Decision()* informa se a nova politica foi implementada com sucesso no PEP.

O protocolo SOAP suporta diversos formatos de codificação e pode-se adotar mecanismo SOAP RPC para aplicações distribuídas. No mapeamento COPS- PR para SOAP proposto, a informação de políticas originalmente definidas em PIBs pode ser traduzida em objetos de uma linguagem de programação orientada a objetos (C++), e então transferida entre PDPs e PEPs usando o formato de codificação suportado pelo SOAP.

O mapeamento proposto (FRANCO et al., 2006) mostra que, se em termos de modelagem o protocolo SOAP não é totalmente aderente ao modelo de comunicação do COPS-PR, o SOAP pode ser usado para provisionamento de políticas se suas mensagens forem usadas para lidar com os eventos relativos a políticas detectados ou ocorridos no PDP ou nos PEPs.

Uma avaliação comparativa de desempenho do uso do SOAP e do NETCONF em gerenciamento de políticas de QoS em relação ao uso de COPS-PR, considerando consumo de banda e atraso introduzido pelos protocolos, mostra que o SOAP ou o NETCONF podem substituir o COPS-PR com vantagens (PEREIRA&GRANVILLE, 2008). Apesar das mensagens codificadas em XML usadas pelo SOAP e NETCONF consumirem mais banda que as mensagens binárias do COPS-PR, versões comprimidas das mesmas mensagens XML acabam consumindo menos banda. Considerando-se o atraso introduzido pelos protocolos, o COPS-PR tem desempenho pior que o SOAP ou o NETCONF, devido ao processamento requerido pelas mensagens binárias ser maior que o requerido por mensagens XML, mesmo quando é introduzida a compressão das mensagens XML. Como o SOAP e o NETCONF permitem a comunicação entre entidades em domínios distintos na Internet, soluções para a questão de segurança devem ser estudadas e certamente impactarão o desempenho destes protocolos.

Em geral, a modelagem proposta e os resultados das avaliações obtidas por (FRANCO et al., 2006; PEREIRA; GRANVILLE, 2008] levam à conclusão de que o SOAP e o NETCONF podem vir a ser alternativas reais de substituição ao COPS-PR.

CAPÍTULO 7 CONCLUSÕES E TRABALHOS FUTUROS

Esse trabalho de dissertação propôs e avaliou uma arquitetura de suporte de Qualidade de Serviço para aplicações de Telemedicina, com suporte ao uso concorrente de Vídeo sob Demanda e Videoconferência sobre uma infra-estrutura de rede IP, dentro do escopo projeto InfraVIDA.

A arquitetura de suporte proposta contempla o gerenciamento baseado em políticas de Qualidade de Serviço e a integração a nível de plano de controle entre o *Bandwidth Broker*, elemento servidor de políticas e gerenciador de recursos da rede, e os respectivos elementos de gerenciamento do serviço de Vídeo sob Demanda na Rede de Vídeo Digital (*DynaVideo*) e do serviço de Videoconferência, o *Gatekeeper* na arquitetura de rede H.323. O *Bandwidth Broker* recebe solicitações de alocação dinâmica de recursos das aplicações de Vídeo sob Demanda e Videoconferência, as valida contra contratos de níveis de serviço *(*SLAs) acordados, e aloca recursos através de distribuição de configuração de mecanismos *DiffServ* em dispositivos de roteamento na rede.

Um resumo da arquitetura de suporte de gerenciamento de Qualidade de Serviço e a apresentação e discussão dos resultados obtidos em rede protótipo experimental foram reportados em artigos publicados nos congressos internacionais WebMedia/LA-Web 2004 (LAGE et al., 2004a) e IEEE IPOM 2004 (LAGE et al., 2004b) e discutidos na dissertação.

A arquitetura proposta se mostra factível, como demonstrado pela rede protótipo experimental, e, além disso, a especificação de operação efetiva de QoS no escopo da arquitetura proposta valida a operação efetiva do DiffServ na garantia de Qualidade de Serviço desejada para as aplicações de Telemedicina contempladas, como a Segunda Opinião Médica - que faz uso concorrente de sessões de Videoconferência, para a análise conjunta de prontuários de pacientes com um médico especialista à distância, e de sessões de Imagem ou Vídeo sob Demanda, que permitem a visualização de exames médicos, inclusive imagens radiológicas e vídeos de procedimentos cirúrgicos ou exames de ultra-som, com o objetivo de diagnóstico médico.

A implantação da arquitetura proposta é aplicável dentro de um domínio administrativo, como o da RNP, ou por uma operadora de telecomunicações, usando infra-estrutura de rede IP estadual ou nacional, para a oferta de serviço diferenciado voltado à Telemedicina. A oferta do mesmo serviço na Internet aberta, em um cenário no qual clínicas

remotas e grandes centros médicos sejam atendidos com acessos de diferentes operadoras, poderia se dar através de contratos de serviço de troca de tráfego e comunicação dinâmica entre os elementos gerenciadores de recursos de rede e QoS (BBs) dessas operadoras.

Como trabalho futuro, sugiro que a arquitetura de suporte proposta seja adicionada a implementação do protocolo SOAP, como alternativa ao COPS-PR na configuração dinâmica dos roteadores para o provisionamento de políticas, conforme mapeamento do protocolo COPS-PR em SOAP descrito nessa dissertação.

REFERÊNCIAS

ABRA. **Ambiente Brasileiro de Aprendizagem**. Disponível em: <http://www.abranet.ufba.br/piloto>. Acesso em: 02 maio/2004.

ARAÚJO, P. et al. **Backbone Simulator:** uma ferramenta de simulação de Rede para a definição de QoS em Aplicações Multimídia, 2003.

ARORA, A. et al. **Web Services for Management,** June 2005. Disponível em: <http://specs.xmlsoap.org/ws/2005/06/management/ws- management.pdf. >. Acesso em: 05 jun 2005

BARBOSA, K. **HealthNet:** um Sistema Integrado de Apoio ao Telediagnóstico e à Segunda Opinião Médica. Dissertação de Mestrado, Centro de Informática/UFPE, Nov. 2001.

BARRY, D. K. **Web Services and Service-Oriented Architectures***:* The Savvy Manager's Guide. Morgan Kaufmann, 2003.

BRAY, T.; PAOLI, J.; SPERBERG-MCQUEEN, C M.; MALER, E. **eXtensible Markup Language (XML)** ***1.0 (Fourth Edition)***. W3C Recommendation 16 August 2006. Disponível em:< http://www.w3.org/TR/xml>. Acesso em: 05 jul. 2008.

BERC, L. et al. RTP Payload Format for JPEG-compressed Video**. RFC 2435,** IETF Network Configuration WG, October 1998.

BLAKE, S. et al. ***An Architecture for Differentiated Services*****. RFC 2475**, Network Working Group, December 1998.

BLESS, R. et al. A Lower Effort Per-Domain Behavior (PDB) for Differentiated Services, **Internet informational RFC 3662,** December 2003.

BRADEN, R.; CLARK, D.; SHENKER, S. ***Integrated Services in the Internet Architecture: an Overview*****. RFC 1633,** June 1994.

BRADEN, R. et al. ***Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification*****. RFC 2205**, September 1997.

BRADEN, R. ***Recommendations on Queue Management and Congestion Avoidance in the Internet*****. RFC 2309**, Network Working Group, April 1998.

CHAN, K. et al. ***COPS Usage for Policy Provisioning (COPS-PR)*****. RFC 3084**, Network Working Group, March 2001.

CHARRIER, M.; SANTA CRUZ, D.; LARSSON, M. **JPEG2000,** the Next Millennium Compression Standard for Still Image. **Proceedings** … IEEE ICMCS´1999.

CHIMENTO et al. **QBone Signaling Design Team – Final Report.** October 2002. Disponível em : <http://qos.internet2.edu/wg/documents-informational/20020709-chimento-etal-qbone-signaling/> . Acesso em: 03 maio 2004.

CISCO. **DiffServ – The Scalable End-to-End QoS Model**, Cisco Systems White Paper, 2001.

CURBELA, F. et al. Unraveling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI, **IEEE Internet Computing,** vol. 6, no. 2, p. 86–93, 2002.

DAVIE, B. et al. ***An Expedited Forwarding PHB (Per-Hop Behavior).*** **RFC 3246**, Network Working Group, March 2002.

DEMERS, A.; KESHAV, S; SHENKER, S. et al. Analysis and Simulation of Fair Queueing Algorithm. **Journal of Internetworking Research and Experience**, p. 3-26, October 1990.

DICOM – Digital Imaging and Communications in Medicine. Disponível em: <http://medical.nema.org/>. Acesso em: 04 maio 2004.

DURHAM, D. et al. ***The COPS (Common Open Policy Service) Protocol.*** **RFC 2748**, Network Working Group, January 2000.

ELIAS, G. et al. ***Um Serviço de Distribuição de Vídeo Sob Demanda Baseado em uma Rede de Servidores D-VoD***, 2004.

ENNS, R. ***NETCONF Configuration Protocol.*** **RFC 4741**, IETF Network Configuration WG, December 2006.

EVEN, R. RTP Payload Format for H.261 Video Streams. **RFC 4587,** IETF Network Configuration WG, August, 2006 (obsoletes RFC 2032).

FERRAZ, C. **Infra-Estrutura de Video Digital para Aplicações de Telemedicina.** CNPq 10/2001 - ProTeM/RNP 01/2001.

FIORESE, T. et al. Comparing Web Services with SNMP in a Management by Delegation Environment. **Proceedings**… IM 2005 - 9th IFIP/IEEE International Symposium on Integrated Network Management, Nice, France, May 2005.

FLOYD, S.; JACOBSON, V. Random Early Detection Gateways for Congestion Avoidance. **IEEE/ACM Transactions on Networking,** Vol. 1 No. 4, pp. 397-413, August 1993.

FLOYD, S.; JACOBSON, V. Link-sharing and Resource Management Models for Packet ***Networks***. **IEEE/ACM Transactions on Networking,** Vol. 3 No. 4, pp. 365-386, August 1995.

FRANCO, T. et al., ***Substituting COPS-PR: An Evaluation of NETCONF and SOAP for Policy Provisioning***. **Proceedings**… 7th International Workshop on Policies for Distributed Systems and Networks (POLICY'06), 2006.

INTERNATIONAL TELECOMMUNICATIONS UNION - TELECOMMUNICATIONS STANDARDIZATION SECTOR, ***Recommendation* G.114:** One-way Transmission Time, May 2003.

GODDARD, T. ***Using NETCONF over the Simple Object Access Protocol (SOAP),* RFC 4743**, IETF Network Configuration WG, December 2006.

GROSSMAN, D. ***New terminology and Clarification for DiffServ*. RFC 3260**, Network Working Group, April 2002.

GUERIN, R; HEINANEN, J. ***A Two Rate Three Color Marker*. RFC 2698**, September 1999.

INTERNATIONAL TELECOMMUNICATIONS UNION - TELECOMMUNICATIONS STANDARDIZATION SECTOR, Recommendation **H.323**: Packet-based Multimedia Communications Systems, June 2006.

HALIM, 2000] H Halim, M Darmadi. ***Implementation of Bandwidth Broker using COPS-PR***. Honours thesis report, School Of Computing Science And Engineering, UNSW, November 2000.

HEINANEN, J. et al., ***Assured Forwarding PHB Group*. RFC 2597**, Network Working Group , June 1999.

HERSENT, J. P.; GURLE, D. **IP Telephony*: Deploying Voice-over-IP Protocols***. John Willey and Sons, 1999.

HOFFMAN et al. ***RTP Payload Format for MPEG1/MPEG2*** *Video*, **RFC 2250,** IETF Network Configuration WG, January 1998.

HUBERT, B. **Linux Advanced Routing & Traffic HOWTO**. Disponível em: <http://lartc.org/howto/index.html.>. Acesso em 03 maio 2004.

JACOBSON, V. Congestion Avoidance and Control. **Computer Communication Review**, Vol. 18 No. 4, pp. 314-329, August 1988.

JACOBSON, V.; NICHOLS, K.; ZHANG, L. ***A Two-Bit Differentiated Services Architecture For The Internet*. RFC 2638**, Network Working Group, July 1999.

JAVA ***2 Platform Standard Edition, v1.4.1 API Specification***. Sun Microsystems 2002. Disponível em: <http://java.sun.com/j2se/1.4.1/docs/api/>. Acesso em 06 maio 2004.

JDBC Java Database Connectivity. Disponível em: <http://java.sun.com/products/jdbc/overview.html>. Acesso em: 05 jul.2008

JHA, S.; HASSAN, M. ***Engineering Internet QoS***. Artech House, 2002.

JPEG Committee Home Page. Disponível em: <http://www.jpeg.org/>. Acesso em: 06 maio 2004.

JAVA Web Services Developer JWSD ***Pack v1.1.*** Sun Microsystems, 2003. Disponível em: <http://java.sun.com/webservices/webservicespack.html.>. Acesso em: 06 maio 2004.

KLIE, T.; STRAUSS, F. ***Integrating SNMP agents with XML-based management systems*****. IEEE Communications Magazine,** vol. 42, no. 7, pp. 76–83, July 2004.

KOREN et al. ***Enhanced Compressed RTP (CRTP) for Links with High Delay, Packet Loss and Reordering.*** *RFC 2545*, IETF Network Configuration WG, July 2003.

KUROSE, J.; ROSS, K. **Computer Networking*****: A Top-Down Approach Featuring the Internet***, 2. ed., São Paulo: Addison Wesley, 2003.

LAGE, A. L. **Qualidade de Serviço em Redes IP para Aplicações de Tele-Saúde.** Relatório Técnico apresentado no Workshop InfraVIDA, Gramado, SBRC 2004.

LAGE, A. L. et al. ***A Quality of Service Framework for Tele-Medicine Applications*****. Proceedings**… WebMedia/LA-Web Joint Conference - 2nd Latin American Web Congress and 10th Brazilian Symposium on Multimedia and the Web, Ribeirão Preto, 2004a.

LAGE, A. L. et al. ***A Quality of Service Approach for Managing Tele-Medicine Multimedia Applications Requirements*****.** In: Proceedings of the IEEE Workshop on IP Operations and Management - IPOM, Vol. 1, Beijing, 2004b, p. 186-190.

LAINE, J. ; SARITO, S. ; PRIOR, R. **Introduction to RUDE & CRUDE**. 2002. Disponível em: <http://rude.sourceforge.net/#intro>. Acesso em: 05 maio 2004.

LEITE, L.; SOUZA FILHO, G.; BATISTA, T. **DynaVideo - A Dynamic Video Distribution Service.** 6 Eurographics Workshop in Multimedia. 2001. **Proceedings**… EGMM2001, 2001, Manchester.

LINUX DIFFSERV. ***Differentiated Services On Linux***. 2001. Disponível em: <http://diffserv.sourceforge.net/>. Acesso em: 04 maio 2004.

MANTAR et al. ***A Scalable Model for Inter-Bandwidth Broker Resource Reservation and Provisioning***, **IEEE Journal on Selected Areas in Communications (JSAC**), Vol.22, No.10, December 2004.

MANTAR et al. ***A Bandwidth-Broker Based Inter-domain SLA Negotiation***. In : A. Helmy et al. (Eds.): MMNS 2006, LNCS 4267, pp. 134–140, 2006. IFIP International Federation for Information Processing 2006.

MARTINS et al. **Valcarenghi, Managing IP Networks*****: Challenges and Opportunities***. Wiley-IEEE Computer Society, 2003.

MILLS, D. ***Network Time Protocol (Version 3: Specification, Implementation and Analysis.*** **RFC 1305**, 1992.

MIRAS, D. **Network QoS Needs for Advanced Internet Application:** ***A Survey***, 2002. Disponível em: <http://qos.internet2.edu/wg/apps/fellowship/Docs/Internet2AppsQoSNeeds.pdf>. Acesso em: 05 maio 2004.

MPEG Home Page. Disponível em: <http://www.chiariglione.org/mpeg/>. Acesso em: 05 maio 2004.

MySQL Reference Manual. Disponível em: <http://www.mysql.com/documentation/mysql/bychapter/index.html>. Acesso em: 04 maio 2004.

NAGLE, J. ***On Packet Switches with Infinite Storage.*** **RFC 970,** December 1985.

NICHOLS et al. ***Definition Of The Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*****. RFC 2474**, Network Working Group, December 1998.

NICHOLS, K.; CARPENTER, B. ***Definition Of Differentiated Services Per-Domain Behaviors and Rules for their Specification*****. RFC 3086**, Network Working Group, April 2001

ORGANIZATION FOR THE ADVANCEMENT OF STRUCTURED INFORMATION STANDARDS (OASIS), 2005. Disponível em: <http://www.oasis-open.org>. Acesso em: 05 jul. 2008.

ODOM, W. ***Deploying Cisco QoS***. [S.l]: Cisco Press, 2003.

OLIVEIRA FILHO, J. L. ***Tecnologias Diffserv como Suporte para a Qualidade de Serviço (QoS) de Aplicações Multimídia - Aspectos de Configuração e Integração.*** Dissertação de Mestrado, UNIFACS 2006.

PAREKH, A. K.; GALLAGER, R. G. A Generalized Processor Sharing approach to Flow control in Integrated Services Networks: The Single Node Case. **Proceedings**…IEEE Infocom., 1992.

PEREIRA, R. C.; GRANVILLE, L. Z. ***On the Performance of COPS-PR and NETCONF in an Integrated Management Environment for DiffServ-enabled Networks***. **Proceedings**… 15th International Conference on Telecommunications (ICT 2008), 2008.

PHAM, K. B.; NGUYEN, R. **Implementation Of A Bandwidth Broker In Java,** June 2003.

PRAS, A. et al. ***Comparing the Performance of SNMP and Web Services-Based Management***. **IEEE eTNSM - eTransactions on Network and Service Management**, vol. 1, no. 2, p. 11, Dec. 2004.

RABELO, H. ***Utilizando ACME para Descrever o Código Aberto do Projeto Open H.323.*** SBMÍDIA2001, 2001, Florianópolis. **Anais**... VII Simpósio Brasileiro de Sistemas Multimídia e Hipermídia. Sociedade Brasileira de Computação, 2001.

RAMAKRISHNAN, K.; FLOYD, S. ***A Proposal to add Explicit Congestion Notification (ECN) to IP***. **RFC 2481**, Network Working Group, January 1999.

RAMAKRISHNAN, K.; FLOYD, S.; BLACK, D. ***The Addition of Explicit Congestion Notification (ECN) to IP***. **RFC 3168**, Network Working Group, September 2001

SCHAFER, R. SIKORA, T. ***Digital Video Coding Standards and Their Role in Video Communications***. **Proceedings**… IEEE Vol. 83, pp. 907-923, 1995.

SCHULZRINNE et al. ***RTP: A Transport Protocol for Real-Time Applications.*** **RFC 3550,** IETF Network Configuration WG, July, 2003.

SIKORA, T. ***Digital Video Coding Standards and Their Role in Video Communications***. In: Signal Processing for Multimedia, pp. 225-252, IOS Press, 1999.

SILVA, O.; ELIAS, G.; LEMOS, G. ***Serviço de Gerência e Seleção de Servidores de Vídeo***, 2004

SHREEDHAR, M.; VARGHESE, G. Efficient Fair Queueing using Deficit Round Robin", **Procedings**… ACM SIGCOMM 95, 1995.

BOX. D. et. al. ***Simple Object Access Protocol (SOAP) 1.2*** . WC3 Recommendation 27 April 2007. Disponível em: < http://www.w3.org/TR/SOAP12/>. Acesso em: 06 jul. 2008.

SOHAIL, S.; JHA. S. **The Survey Of Bandwidth Broker.** Technical Report UNSW CSE TR 0206, School of Computer Science and Engineering, University of New South Wales. May 2002.

SOLLAUD, A. ***RTP Payload Format for the G.729.1 Audio Codec. RFC 4749***, IETF Network Configuration WG, October 2006.

STEVE, G. HULL, D.; MURRAY, B. ***Web Services Base Notification (WS-BaseNotification)***, 1st ed., IBM, Tibco, Hewlett-Packard Company, July 2005. Disponível em : <http://www.oasis-open.org/committees/download.php/13488/wsn-ws-base notification-1.3-spec-pr-01.pdf> . Acesso em: 05. jul. 2005.

WORLD Wide Web Consortium - Web Services Activity [W3C]. Disponível em: <http://www.w3.org/2002/ws/> . Acesso em: 05 jul. 2008.

WENGER et al. ***RTP Payload Format for H.264 Video.*** **RFC 3984,** IETF Network Configuration WG, February 2005.

YAVATKAR, R.; PENDARAKIS, D.; GUERIN, R. ***A Framework for Policy Based Admission Control.*** **RFC 2753**, January 2000.

LAGE PEREIRA, Ana Lúcia. *Uma Proposta de Arquitetura de Suporte para a Qualidade de Serviço Orientada para Aplicações de Telemedicina*. 2008. xxx f. Dissertação (Mestrado Profissional em Redes de Computadores) – Universidade Salvador – UNIFACS. Salvador, Bahia.