FERNANDO LUIZ DE ALMEIDA

PBQoS – Uma Arquitetura de Gerenciamento baseado em Políticas para
distribuição otimizada de conteúdo multimídia com controle de QoS em
redes Overlay

São Paulo
2011

FERNANDO LUIZ DE ALMEIDA

PBQoS – Uma Arquitetura de Gerenciamento baseado em Políticas para
distribuição otimizada de conteúdo multimídia com controle de QoS em
redes Overlay

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para a obtenção
do título de Doutor em Engenharia.

Área de Concentração: Sistemas Digitais

Orientadora: Profa. Dra. Graça Bressan

São Paulo
2011

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 28 de janeiro de 2011.

Assinatura do autor _____

Assinatura do orientador _____

FICHA CATALOGRÁFICA

Almeida, Fernando Luiz de

PBQoS – uma arquitetura de gerenciamento baseado em políticas para distribuição otimizada de conteúdo multimídia com controle de QoS em redes Overlay / F.L. de Almeida. -- ed.rev. -- São Paulo, 2011.

187 p.

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1. Redes de computadores (Gerenciamento) I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II. t.

DEDICATÓRIA

Dedico este trabalho aos meus pais Jorge e Vera, aos meus irmãos Leandro e Belisa e a minha esposa Francielli que sempre estiveram presentes nos momentos mais difíceis.

AGRADECIMENTOS

A Deus pela coragem de facear as dificuldades criadas por nós mesmos, pelas provas que nos aperfeiçoam o raciocínio e pela sabedoria nos momentos de incertezas. Agradeço pela fé que me mantém vivo.

Eu não teria chegado até aqui sem o amor incondicional, a paciência, o apoio, o incentivo e compreensão dos meus pais Jorge de Almeida e Vera Lucia Bavia de Almeida, meus irmãos Leandro Luiz de Almeida e Belisa Bavia de Almeida e minha esposa Francielli Almeida.

A Profa. Dra. Graça Bressan pela orientação e constante estímulo transmitido durante todo o percurso. Agradeço pela oportunidade e confiança em meu trabalho durante o doutorado e projetos de pesquisa nos quais trabalhei. A parceria orientada e orientadora serviu de base para fortalecer e preencher ainda mais os conhecimentos adquiridos no decorrer desta etapa.

Ao Prof. Dr. Denis Gabos pelas “rápidas” e pertinentes discussões a respeito do tema, conceitos e direções a seguir.

Ao Prof. Dr. Wilson Ruggiero agradeço pelas oportunidades a mim concedidas em seus projetos de pesquisas e espaço em seu laboratório para desenvolvimento do trabalho.

Ao Prof. Dr. Ademar Cardoso pela amizade, compreensão e pelas incansáveis discussões que tivemos durante o processo de validação da proposta aqui apresentada.

Agradeço aos amigos do LARC que contribuíram direta ou indiretamente para o desenvolvimento do trabalho, pela amizade e incentivo.

A todos que, de uma forma ou de outra, contribuíram para a realização deste trabalho, minha profunda gratidão e meu sincero reconhecimento.

Meu muito Obrigado!

*O progresso do mundo moderno nos oferece facilidade de conforto e prazer,
de luxo e beleza motivando-nos à conquista do sucesso.*

*A natureza é rica em sabedoria em abundância, mostrando-nos que Deus nos
criou para usufruirmos todas as coisas da vida.*

*Contudo, todo progresso só é positivo quando inclui em suas metas o sucesso
do seu meio social. Infelizes daqueles que, na ânsia de prosperar, atropelam os
direitos dos outros, violando a lei da integridade universal achando que cortam o
caminho.*

*Perceberão que o egoísmo e a ganância não são funcionais para se manter o
fluxo da abundância.*

*Divina a seu favor, pois a Natureza responderá com violência corretiva
ensinando-os que só o verdadeiro bem é capaz de proporcionar a prosperidade
completa.*

*O caminho da verdadeira vitória é sempre árduo e cheio de surpresas
desafiadoras que determinarão o desenvolvimento de nossos potenciais inatos,
garantindo a evolução do nosso espírito eterno.*

*A cada novo minuto você tem liberdade e responsabilidade de escolher para
onde quer seguir, mas é bom lembrar que TUDO NA VIDA TEM SEU PREÇO.*

Zíbia Milane Gaspareto

RESUMO

Avanços nas tecnologias de comunicação e processamento de sinais além de mudar a forma de como realizar negócios em todo o mundo, têm motivado o surgimento de serviços e aplicações multimídia na Internet de forma crescente. Como consequência, é possível conceber, desenvolver, implantar e operar serviços de distribuição de vídeo digital na Internet, tanto na abordagem sob-demanda quanto ao vivo. Com o aumento das aplicações multimídia na rede, torna-se cada vez mais complexo e necessário definir um modelo eficiente que possa realizar o gerenciamento efetivo e integrado de todos os elementos e serviços que compõe um sistema computacional. Pensando assim, este trabalho propõe uma arquitetura de gerenciamento baseado em políticas aplicada à distribuição de conteúdo multimídia com controle de QoS (*Quality of Service*) em redes de sobreposição (*overlay*). A arquitetura é baseada nos padrões de gerenciamento por políticas definida pela IETF (*Internet Engineering Task Force*) que, através de informações contextuais (rede e clientes) administra os serviços disponíveis no sistema. Faz uso dos requisitos de QoS providos pela rede de distribuição e os compara com os requisitos mínimos exigidos pelos perfis das aplicação previamente mapeados em regras de políticas. Dessa forma é possível controlar e administrar os elementos e serviços do sistema, afim de melhor distribuir recursos aos usuários deste sistema.

Palavras-chave: Gerenciamento por políticas (PBNM – *Policy Based Network Management*). Redes *Overlay*. Qualidade de Serviço (QoS – *Quality of Service*). Roteamento.

ABSTRACT

Advances in communication technologies and signal processing have not only changed the way business is conducted around the world, but have also driven the development of services and multimedia applications on the Internet. As a result, it is possible to design, develop, deploy and operate services for digital video distribution on the Internet, both according to an on-demand approach and live. Because of the increase in multimedia applications on the network, it has become increasingly more complex and necessary to define an efficient architecture that can achieve the effective and integrated management of all the elements and services that compose a computer system. With this in mind, this study proposes developing a robust and efficient architecture based on IETF (Internet Engineering Task Force) policy management standards applied to multimedia distribution content with QoS (Quality of Service) control in Overlay Networks. This architecture makes use of QoS requirements provided by the distribution network and compares them to the minimum requirements demanded by each type of application previously mapped in the policy rules. This system makes it possible to control and manage system information and services and also to distribute resources to system users better.

Keywords: Policy Based Network Management. Overlay Network. Quality of Service. Routing.

LISTA DE ILUSTRAÇÕES

Figura 1 – Arquitetura de Políticas definida pela IETF	26
Figura 2 – Modelo de gerenciamento baseado em Política	27
Figura 3 – Modelo de política	28
Figura 4 – Causa e efeito - execução de ações	29
Figura 5 – Exemplos simples de políticas	29
Figura 6 – Elementos arquiteturais do modelo de gestão por políticas	30
Figura 7 – Rede sobreposta para P2P	33
Figura 8 – Exemplo de uma rede sobreposta	34
Figura 9 – Categorização baseada em tarefas das aplicações	57
Figura 10 – Efeito da variação do atraso nas aplicações	66
Figura 11 – Vazão e interatividade para serviços comuns de vídeo	75
Figura 12 – Qualidade do vídeo x taxa de codificação	79
Figura 13 – Qualidade do vídeo x taxa de codificação	81
Figura 14 – Limiares aceitáveis de qualidade entre áudio e vídeo	81
Figura 15 – Perfil <i>Voz Convencional</i>	86
Figura 16 – Perfil <i>Mensagem de voz</i>	86
Figura 17 – Perfil <i>Streaming de Áudio de Alta Qualidade</i>	87
Figura 18 – Perfil <i>VideoPhone</i>	87
Figura 19 – Perfil <i>Web-Browsing</i>	89
Figura 20 – Perfil <i>Bulk DataTransfer</i>	89
Figura 21 – Perfil <i>Aplicação de alta prioridade</i>	90
Figura 22 – Perfil <i>CommandControl</i>	90
Figura 23 – Perfil <i>Imagens estáticas</i>	91
Figura 24 – Perfil <i>Jogos Interativos</i>	91
Figura 25 – Perfil <i>Telnet</i>	91
Figura 26 – Perfil <i>EmailServerAccess</i>	92
Figura 27 – Perfil <i>FaxStoreForward</i>	92
Figura 28 – Perfil <i>Transação de baixa Prioridade</i>	93
Figura 29 – Perfil <i>EmailServerToServerTransfer</i>	93
Figura 30 – Perfil <i>Usenet</i>	94
Figura 31 – Mapeamento dos requisitos de QoS centrados no usuário	95
Figura 32 – Modelo de categorias de QoS centrado no usuário	95
Figura 33 – Adaptação da Arquitetura TCP/IP para o sistema PBQoS	98
Figura 34 – Detalhamento das camadas BOS e SON	100
Figura 35 – Elementos arquiteturais – PBQoS	103
Figura 36 – Representação da MIB do sistema	104
Figura 37 – Representação abstrata do Repositório de Políticas	105
Figura 38 – Perfis de aplicações	111
Figura 39 – Cenário de uso – Arquitetura PBQoS	116
Figura 40 – Solicitação de rota para um aplicação VoIP	117
Figura 41 – Representação do estado de tráfego dos enlaces da rede	118
Figura 42 – Configuração inicial do Sistema	119
Figura 43 – Manutenção de uma rota	120
Figura 44 – Manutenção de uma rota	121
Figura 45 – Manutenção e Controle – “alerta amarelo” ($\alpha = 70\%$)	121
Figura 46 – Manutenção e Controle – “alerta vermelho” ($\beta = 85\%$)	122
Figura 47 – Abstração da definição de uma rota pelo PDP	123
Figura 48 – Cenário composto por 2 nós	128
Figura 49 – Cenário composto por 4 nós	128
Figura 50 – Cenário composto por 6 nós	129
Figura 51 – Representação abstrata de uma rede	133

Figura 52 – SLA para uma Aplicação VoIP	134
Figura 53 – Representação dos enlaces para requisito vazão	135
Figura 54– Representação dos enlaces para o requisito atraso.....	136
Figura 55 – Definição de variação de atraso.....	138
Figura 56 – Representação dos enlaces para o requisito variação do atraso	140
Figura 57 – Cálculo da perda em um enlace.....	142
Figura 58 – Cálculo de perda para uma rota com diversos enlaces.....	142
Figura 59 – Representação dos enlaces para o requisito perdas.....	142
Figura 60 – Grafos resultantes após a iteração do sistema.....	143
Figura 61 – Fluxograma do Algoritmo PBQoS	146
Figura 62 – Representação da rede em forma de Grafo	151
Figura 63 – Aplicação PBQoS	152
Figura 64 – Política <i>Maximizar Vazão</i> – Requisição de rota ao sistema PBQoS.....	153
Figura 65 – <i>log</i> gerado pelo sistema – Política <i>Maximizar Vazão</i>	154
Figura 66 – Política <i>Minimizar Perdas</i> – Requisição de rota ao sistema PBQoS.....	155
Figura 67 – Política <i>Minimizar Perda</i> – <i>log</i> gerado pelo sistema	155
Figura 68 – Política <i>Minimizar variação do atraso</i> – Requisição de uma rota ao sistema PBQoS	156
Figura 69 – <i>log</i> gerado pelo sistema – Política <i>Minimizar Perda</i>	157
Figura 70 – Requisição de rota ao sistema PBQoS – Política <i>Minimizar Atraso</i>	158
Figura 71 – <i>log</i> gerado pelo sistema – Política <i>Minimizar Atraso</i>	158
Figura 72 – Política <i>Visão Global de QoS</i> – Requisição de rota ao sistema PBQoS.....	159
Figura 73– <i>log</i> gerado pelo sistema – Política <i>Visão Global de QoS</i>	160
Figura 74 – Política <i>Visão Global de QoS</i> – Processo de Manutenção de rota	161
Figura 75 – <i>log</i> do sistema PBQoS – Políticas de Manutenção	162
Figura 76 – Política <i>Visão Global de QoS</i> – Processo de Manutenção de uma rota	162
Figura 77 – <i>log</i> gerado pelo sistema PBQoS – Processo de manutenção	163
Figura 78 – Política <i>Visão Global de QoS</i> – Processo de Manutenção de uma rota	164
Figura 79 – <i>log</i> do sistema PBQoS – Política de manutenção	164
Figura 80 – Topologia da rede – Cenário 2.....	172
Figura 81 – Execução da Aplicação PBQoS	173
Figura 82 – Política <i>Maximizar Vazão</i> – Requisição de rota (1)	174
Figura 83 – <i>log</i> gerado pelo sistema PBQoS após o processo de admissão	174
Figura 84 – Política <i>Maximizar Vazão</i> – Requisição de rota (2)	175
Figura 85 – Política <i>Maximizar Vazão</i> – Requisição de rota (30)	176

LISTA DE TABELAS

Tabela 1 – Comparativo entre as propostas estudadas	51
Tabela 2 – Relação entre a classificação dos requisitos e a necessidade da aplicação.....	54
Tabela 3 – Requisitos mínimos exigidos por uma aplicação de VoIP.....	59
Tabela 4 – Requisitos mínimos exigidos para orquestração de Áudio de alta qualidade.....	60
Tabela 5 – Diretrizes de atraso para VoIP	70
Tabela 6 – Diretrizes de variação de atraso para VoIP	70
Tabela 7 – Efeito de perdas de pacotes na qualidade de voz	71
Tabela 8 – Padrões de codificação de áudio	72
Tabela 9 – Relação entre a qualidade de áudio e Taxa de transmissão	73
Tabela 10 – Requisitos típicos de banda para os formatos de vídeo mais comuns.....	74
Tabela 11 – Requisitos de desempenho para aplicações de áudio e vídeo	88
Tabela 12 – Objetivos de desempenho das aplicações de dados.....	94
Tabela 13 – Representação do estado atual da rede – MIB	134
Tabela 14 – Cálculo da perda fim-a-fim para uma aplicação VoIP.....	143
Tabela 15 – Critérios definidos pelas Políticas de Controle para o processo de definição do fator de otimização de uma rota	144
Tabela 16 – Cálculo do fator para rotas encontradas.....	145
Tabela 17 – Controle do processo de admissão de um serviço	147
Tabela 18 – Exemplo da MIB geral da rede	150
Tabela 19 – Rotas encontradas pelo sistema PBQoS.....	150
Tabela 20 – Cálculo do QoS para cada rota encontrada na rede.....	151
Tabela 21 – Configuração requisitada ao sistema PBQoS.....	152
Tabela 22 – Política <i>Maximizar Vazão</i> – Rotas encontradas entre os nós 1 e 2	153
Tabela 23 – Política <i>Minimizar Perdas</i> – Rotas encontradas entre os nós 1 e 2.....	154
Tabela 24 – Política <i>Minimizar Variação do atraso</i> – Rotas encontradas entre os nós 1 e 2.....	156
Tabela 25 – Política <i>Minimizar Atraso</i> – Rotas encontradas entre os nós 1 e 2	157
Tabela 26 – Política <i>Visão Global de QoS</i> – Rotas encontradas entre os nós 1 e 2	159
Tabela 27 – SLA requerido – Processo de manutenção de rota	161
Tabela 28 – Conjunto de perfis de Aplicações	165
Tabela 29 – MIB geral da rede.....	165
Tabela 30 – Requisição de recursos – Política <i>Maximizar Vazão</i>	166
Tabela 31 – Requisição de recursos – Política <i>Minimizar Atraso</i>	166
Tabela 32 – Requisição de recursos – Política <i>Minimizar Variação do Atraso</i>	167
Tabela 33 – Requisição de recursos – Política <i>Minimizar Perdas</i>	167
Tabela 34 – Requisição de recursos – Política <i>Visão Global de QoS</i>	168
Tabela 35 – Comparativo das rotas encontradas pelo sistema PBQoS	169
Tabela 36 – Resumo dos resultados gerados pelas políticas ativas	169
Tabela 37 – SLA utilizado para gerar carga na rede	169
Tabela 38 – Rotas selecionadas para cada requisição	170
Tabela 39 – MIB geral da rede – Cenário 2	171
Tabela 40 – Requisição de recursos – Política <i>Maximizar Vazão</i> (1)	173
Tabela 41 – Informações de uma requisição ao sistema PBQoS (2)	175
Tabela 42 – Resumo dos resultados gerados – Processo de configuração	177
Tabela 43 – SLA utilizado para gerar carga na rede	177
Tabela 44 – Rotas selecionadas para cada requisição	178

LISTAS DE ABREVIATURAS E SIGLAS

α – *Alpha*
 β – *Beta*
 λ – *Lâmbda*
ARQ – *Automatic Repeat reQuest*
ATM – *Asynchronous Transfer Mode*
BOS – *Basic Overlay Network*
CDSP – *Content Distribution Service Provider*
COPS – *Common Open Policy Service Protocol*
DFS – *Depth-First Search*
DiffServ – *Differentiated Services*
DVD – *Digital Video Disc*
FEC – *Forward Error Correction*
FM – *Frequency Modulation*
FTP – *File Transfer Protocol*
HDTV – *High Definition TV*
HTTP – *Hypertext Transfer Protocol*
ICMP – *Internet Control Message Protocol*
IETF – *Internet Engineering Task Force*
IntServ – *Integrated Services*
IPv4 – *Internet Protocol version 4*
ISP – *Internet Service Provider*
ISPs – *Internet Service Providers*
MATLAB – *Matrix Laboratory*
MIB – *Management Information Base*
MPEG – *Moving Picture Experts Groups*
MPLS – *Multiprotocol Label Switching*
NFS – *Network File System*
OPEP – *Overlay Policy Enforcement Point*
OPMP – *Overlay Policy Monitoring Point*
OSPF – *Open Shortest Path First*
PBMN – *Policy Based Management Network*
PBQoS – *Policy Based Quality of Service*
PDP – *Policy Decision Point*
PEP – *Policy Enforcement Point*
PSTN – *Public Switched Telephone Network*
QoE – *Quality of Experience*
QoS – *Quality of Service*
RSVP – *Resource Reservation Protocol*
SDI – *Serial Digital Interface*
SDTV – *Standard Definition TV*
SLA – *Service Level Agreement*
SLM – *Service Level Management*
SMS – *Simple Message System*
SNMP – *Simple Network Management Protocol*
SSON – *Specific Service Overlay Networks*
TED – *Traffic Engineering Database*
VB – *Visual Basic*

VCR – *Videocassette Recorder*
VHS – *Video Home System*
VoIP – *Voice over IP*

|

SUMÁRIO

Capítulo 1	17
Introdução	17
1.1 – Motivação	18
1.2 – Justificativa	19
1.3 – Objetivos	21
1.4 – Metodologia	22
1.5 – Contribuições	22
1.6 – Estrutura da Tese	23
Capítulo 2	25
Conceitos básicos	25
2.1 – Gerenciamento Baseado em Políticas	25
2.1.1 – Políticas	27
2.1.2 – Benefícios do gerenciamento baseado em Políticas	31
2.2 – Redes Sobrepostas	31
2.3 – Protocolo SNMP	35
2.4 – Protocolo COPS	36
2.5 – Considerações finais	37
Capítulo 3	39
Trabalhos Relacionados	39
3.1 – Policy based Management of Content Distribution Networks	39
3.2 – CORDENA – Um Framework para gestão de Redes baseadas em Políticas	40
3.3 – Policy based resource management for QoS aware application in heterogeneous network environments	42
3.4 – OverQoS: Offering Internet QoS Using Overlays	43
3.5 – Automating Overlay Networks Management	44
3.6 – OVERSEA – Towards a Scalable and Effective Architecture for Overlay Networks	46
3.7 – QoS-Aware Overlay Routing with Limited Number of Alternative route candidates and Its Evaluation	47
3.8 – Comparativo entre as propostas estudadas	48
Capítulo 4	52
Requisitos de QoS exigidos pelas Aplicações Avançadas na Internet	52
4.1 – Introdução	52
4.2 – Aplicações Avançadas	53
4.3 – Classificação dos requisitos das Aplicações Avançadas	54
4.3.1 – Aplicações flexíveis e Aplicações adaptativas	57

4.3.2 – Classes de Aplicações de alto nível	59
4.4 – Comportamento das aplicações avançadas e requisitos de QoS	63
4.4.1 – Requisitos de QoS de rede	64
4.4.2 – Requisitos de QoS das Aplicações	66
4.5 – Aplicações que necessitam de QoS	68
4.5.1 – Aplicação de Voz Interativa	68
4.5.1.1 – Efeito de Atraso	69
4.5.1.2 – Efeito da variação do atraso	70
4.5.1.3 – Efeito de Perdas de Pacotes	70
4.5.2 – Aplicações de Áudio	73
4.5.3 – Aplicação de vídeo	73
4.5.3.1 – Vídeo Interativo	74
4.5.3.2 – Vídeo Streaming	76
4.6 – Requisitos que alteram a qualidade	78
4.6.1 – Taxa de Transmissão	79
4.6.2 – Perda de pacotes	80
4.7 – Considerações Finais	82
Capítulo 5	84
Categorias de QoS das Aplicações multimídia	84
5.1 – Requisitos de desempenho	84
5.2 – Considerações de desempenho para diferentes aplicações	85
5.2.1 – Áudio	85
5.2.2 – Vídeo	87
5.2.3 – Dados	88
5.3 – Classificações dos requisitos de desempenho do usuário em categorias de Qualidade de Serviço	94
Capítulo 6	97
Arquitetura do Sistema de Gerenciamento - PBQoS	97
6.1 – Arquitetura	97
6.1.2 – Arquitetura PBQoS	101
6.2 – Cenário de uso – Arquitetura PBQoS	115
6.2.1 – Configuração do Sistema	118
6.2.2 – Manutenção e Controle de uma rota	120
6.2.3 – Escalabilidade da arquitetura PBQoS	124
Capítulo 7	130
Algoritmo de cálculo de rota – PBQoS	130
7.1 – Processo de definição da melhor rota	130

7.2 – Definição do algoritmo PBQoS.....	133
Capítulo 8	149
Implementação Computacional do Sistema PBQoS	149
8.1 – Cenário 1	149
8.1.1 – Processo de Configuração	149
8.2 – Processo de Manutenção e Controle de uma rota	160
8.1.3 – Análise dos resultados gerados pelo sistema PBQoS.....	165
8.2 – Cenário 2	170
8.2.1 – Processo de Configuração	171
8.2.2 – Processo Manutenção	177
8.3 – Considerações Finais.....	179
Capítulo 9	180
Conclusão.....	180
Referências Bibliográficas	183

Capítulo 1

Introdução

Com o grande crescimento das aplicações multimídia nos últimos anos e o aumento expressivo dos usuários que fazem uso dos serviços de rede, é cada vez maior a necessidade de soluções que provê escalabilidade e utilize as redes de comunicação para fornecer serviços de forma eficiente aos usuários. Percebe-se também um grande crescimento voltado para o oferecimento de serviços multimídia interativos, integrando voz, dados, vídeo e imagens em diferentes áreas de atuação.

Avanços nas tecnologias de comunicação e processamento de sinais têm motivado o surgimento de serviços e aplicações multimídia, e como consequência, é possível conceber, desenvolver, implantar e operar serviços de distribuição de conteúdo digital na Internet, tanto na abordagem sob demanda quanto ao vivo. Com o aumento das aplicações multimídia, torna-se cada vez mais complexo e necessário definir um modelo eficiente que possa realizar o gerenciamento efetivo e integrado de todos os elementos e serviços que compõe um sistema computacional. Pensando assim, este trabalho propõe uma arquitetura de gerenciamento baseada em políticas aplicada na distribuição de conteúdo multimídia em redes sobrepostas. A arquitetura faz uso dos requisitos de QoS providos pela rede de distribuição e os compara com os requisitos mínimos exigidos pelos perfis de cada aplicação. Dessa forma é possível controlar e administrar os elementos e serviços do sistema, afim de melhor utilizar os recursos da rede gerenciada.

O conceito de redes sobrepostas (*overlay*) visa minimizar os problemas de flexibilidade existentes nas redes IP, provendo serviços que não são possíveis ou de difícil implementação nessas redes. Através deste conceito, é possível trabalhar com os elementos de rede e realizar o controle na camada de aplicação, abstraindo assim a complexidade de gerenciar os elementos presentes nas camadas inferiores, principalmente no que tange a QoS (*Quality of Service*).

A falta de uma arquitetura de gerenciamento que permita além da administração dos equipamentos, serviços e aplicações, controlar a admissão de requisições poderá limitar a exploração das aplicações e comprometer de forma considerável o desempenho do sistema. Visando distribuir conteúdo multimídia de

forma personalizada, é extremamente importante definir em um nível de abstração superior (políticas de gerenciamento) um modelo de gerenciamento que melhor atenda à necessidade dos usuários.

Uma arquitetura de gerenciamento baseado em Políticas pode ser considerada uma evolução nos princípios de como gerenciar redes e serviços de distribuição, pois favorece a visão global do sistema, de modo a controlar eficazmente as infra-estruturas de uma forma integrada com todas as políticas previamente definidas. Esta arquitetura permite mudar a forma de como as redes são mantidas hoje em dia, pois se baseia num conjunto de entidades de rede que dinamicamente fazem associações de alguns requisitos de QoS e os mapeiam em um nível mais alto de abstração, onde o administrador da rede não precisa se preocupar com detalhes técnicos de configuração.

Nesse contexto, o presente trabalho visa propor uma arquitetura de gerenciamento baseado em políticas que forneça meios para melhor controlar os requisitos de QoS exigidos pelas aplicações multimídia e controle de admissão de novos serviços em um cenário de rede sobrepostas. Para isto, é necessário conhecer o ambiente de configuração mais favorável para distribuição do fluxo de dados entre os clientes que fazem uso do sistema. A estratégia adotada dispõe de elementos que monitoram a rede física e faz o controle de admissão de novas requisições. Dessa forma, é possível fazer a gestão da rede e das aplicações nela contida através do controle do QoS.

1.1 – Motivação

A falta de uma arquitetura de gerenciamento voltada à administração dos serviços de distribuição nos dias atuais pode limitar a exploração das aplicações e comprometer de forma considerável o desempenho do sistema. Algumas técnicas estão sendo estudadas para melhorar o gerenciamento dos componentes das redes, aplicações e serviços, dentre elas está o Gerenciamento Baseado em Políticas (*Policy Based Network Management – PBNM*) (Strassner, 2007) (Teixeira, 2004).

O interesse em uniformizar o gerenciamento dos recursos de forma personalizada às características das redes e aplicações é extremamente importante, pois permite definir em um nível de abstração superior, os requisitos mínimos

exigidos por cada perfil de aplicação e assim, definir políticas de gerenciamento para melhor acomodá-las ao sistema. Dessa forma, é possível definir de forma concreta os serviços pretendidos, ocultando assim, detalhes técnicos prescindíveis aos administradores de sistemas.

Partindo do princípio que as redes sobrepostas foram construídas com intuito de migrar parte da complexidade de roteamento, viabilizar aplicações e serviços na Internet é imprescindível que haja uma arquitetura que opere de forma inteligente e personalizada para administrar os requisitos de QoS de acordo com o perfil das aplicações.

1.2 – Justificativa

Mecanismos inteligentes de distribuição e transmissão de dados multimídia são imprescindíveis no contexto atual de serviços multimídia interativos. Esses mecanismos devem gerenciar a distribuição do fluxo de dados gerados pelas aplicações entre provedores e consumidores de conteúdo. No entanto, quanto maior o tráfego de informações gerado na rede pelas aplicações, mais complexo torna-se o seu gerenciamento. No caso de aplicações envolvendo mídias contínuas (como áudio e vídeo) essa problemática é ainda mais crítica. Nessas aplicações, além da geração de grande volume de dados, existe uma exigência temporal para transmissão das mídias. Quando essa exigência não é respeitada, o atendimento ao cliente pode ser sensivelmente comprometido.

O desenvolvimento de estratégias para distribuição eficiente de dados é uma tendência fortemente verificada nas aplicações que utilizam dados multimídia. Essas estratégias, além de identificar a rota de distribuição (geradores e consumidores de conteúdo), devem considerar aspectos como proximidade dos usuários aos provedores de conteúdo e principalmente capacidade de transmissão das redes. Dessa forma, o desenvolvimento de estratégias responsáveis pelo gerenciamento deste cenário é primordial para o sucesso de aplicações e serviços que envolvam distribuição e transmissão de dados multimídia.

A arquitetura proposta neste trabalho prevê a manutenção de uma rede que seja, ao mesmo tempo, uma infra-estrutura de alto desempenho para comunicação

entre instituições de ensino e pesquisa, laboratórios para testes, desenvolvimento de aplicações e tecnologias de rede avançadas.

Dentro deste contexto, foi proposto o desenvolvimento de uma arquitetura de gerenciamento baseada em políticas, no intuito de reduzir os custos de provisionamento de serviços e melhorar a utilização dos recursos da rede. Esta arquitetura tem como base os requisitos de QoS e através das regras impostas pelas políticas de gerenciamento, realiza o controle operacional do sistema.

Uma rede sobreposta permite aos provedores de conteúdo construir arbitrariamente topologias virtuais sobre a topologia física da rede, onde é realizado o mapeamento da rede física na rede sobreposta (virtual). Dessa forma, é possível construir uma malha completa de conexões lógicas entre todos os nós de borda. Estas conexões lógicas podem ser configuradas como rotas explícitas MPLS (*Multiprotocol Label Switching*) através dos protocolos de sinalização. Os objetivos da Engenharia de Tráfego podem ser alcançados mapeando estas rotas sobre a rede física de tal forma que a distribuição do tráfego seja balanceada para toda a rede (Wang, 2001).

Embora o modelo de rede sobreposta seja simples, é largamente implementado nos roteadores (*backbones*) da Internet e deve suportar algum tipo de escalabilidade, pois com o aumento do número de nós na rede, aumentará drasticamente o número de rotas, adicionando uma complexidade considerável de gerenciamento. Pensando assim, foi escolhido um cenário de rede sobreposta com intuito de migrar parte da complexidade de roteamento das camadas inferiores para a camada de aplicação, facilitando o provisionamento de serviços que não são suportados ou difíceis de serem implementados na rede física. Dessa forma, não há a necessidade de modificar a estrutura existente nas redes atuais, nos protocolos existentes, além de fornecer um sistema totalmente escalável.

Um sistema de gerenciamento baseado em políticas permitirá distribuir os recursos da rede de forma mais eficiente e personalizada para cada tipo de aplicação. O modelo arquitetural é baseado nos padrões IETF (*Internet Engineering Task Force*) de gerenciamento e apresenta um novo modelo de política na construção de rotas com base nas informações contextuais das aplicações, usuários e rede.

1.3 – Objetivos

Os sistemas de gerenciamento, de forma geral, realizam a gestão dos elementos físicos da rede (roteadores, *switches* e *gateways*). Este trabalho propõe uma arquitetura de gerenciamento baseado em políticas que possa realizar o controle de admissão de serviços e administrar o QoS das redes sobrepostas. As Políticas podem ser geradas dinamicamente através de um contexto de informação de usuários, rede, ou então através de provedores de conteúdo, alcançando assim um nível de automação que diminui consideravelmente a interação humana. O trabalho apresenta três modelos de políticas: *Políticas de Configuração*, que serão usadas para configurar a rede e deixá-la pronta para o uso dos serviços; *Políticas de Manutenção* que serão utilizadas para manter o sistema operacional, e de forma reativa realizará o roteamento do fluxo multimídia em função das necessidades exigidas por cada perfil de aplicação, representando o menor custo para o sistema. Por fim, as *Políticas de Controle* são responsáveis por determinar as regras internas que definirão o comportamento do sistema.

A arquitetura busca assegurar que as redes sobrepostas se comportarão de acordo com as necessidades específicas de cada aplicação. Para isto, utiliza técnicas de Engenharia de Tráfego para reduzir o congestionamento e definir critérios para uma melhor utilização dos recursos da rede. Para isto, o ambiente de configuração deve ser conhecido de modo a torná-lo mais favorável para distribuição de recursos entre provedores de conteúdo e usuários. O sistema monitora a rede antes de tomar uma decisão, para que de acordo com o estado atual da rede, o sistema possa determinar quais recursos serão utilizados para melhor atender as requisições exigidas por cada perfil de aplicação.

Como o sistema conhece a topologia de rede, a distribuição dos provedores de conteúdo e localização dos clientes, no momento em que todos os requisitos forem atendidos, o roteamento do fluxo é realizado através dos elementos que possam suportar as necessidades de QoS exigidas pelas aplicações, garantindo assim, um sistema balanceado que opere de acordo com a disponibilidade de recursos para atender as necessidades exigidas pelos perfis das aplicações.

1.4 – Metodologia

Após estudos e análises dos trabalhos relacionados (Verma, 2002), (Teixeira, 2004), (Hetzer, 2007), (Subramanian, 2003), (Oqily, 2007), (Uchoa, 2009), (Ushida, 2006) voltados ao gerenciamento por políticas, foram levantados os requisitos mínimos de QoS que uma aplicação necessita para transmitir conteúdo multimídia de forma eficiente. Estes requisitos foram mapeados em um modelo de requisitos que represente o perfil de uma aplicação. Através destas informações, é possível utilizar a rede de forma personalizada para cada tipo específico de aplicação. Posteriormente foram definidas e mapeadas as políticas que representam as regras e definem o comportamento do sistema através da realização do cálculo da rota que melhor atenda uma determinada solicitação. Para a realização e validação deste trabalho, as seguintes etapas foram contempladas de acordo com a seqüência a seguir:

- Definição dos elementos arquiteturais e suas principais funções;
- Levantamento dos requisitos de QoS presentes na rede e os exigidos pelas aplicações multimídia;
- Mapeamento dos requisitos de QoS em contratos que representam o SLA (*Service Level Agreement*);
- Definição e criação das políticas de configuração, manutenção e controle do sistema PBQoS;
- Elaboração de um algoritmo que seja capaz de encontrar a melhor rota entre dois nós (levando em consideração a imprevisibilidade da rede física - Internet);
- Criação de um protótipo através da ferramenta MATLAB capaz de validar a arquitetura e os algoritmos envolvidos;
- Criação de diferentes cenários que possam ilustrar o uso da arquitetura proposta;

1.5 – Contribuições

A realização do presente trabalho trouxe as seguintes contribuições:

- Definição de uma arquitetura de gerenciamento em redes sobrepostas através de gestão por Políticas (PBNM);
- Integração dos conceitos de gestão por políticas, QoS e redes sobrepostas em um único modelo de informação;
- Definição de um algoritmo capaz de realizar a escolha de uma rota com base nos requisitos de QoS da rede, aplicações e políticas ativas;
- Automatização do processo de geração e construção de rotas;
- Administração dos requisitos de QoS (vazão, atraso, *variação do atraso* e perdas) da rede sobreposta e das aplicações envolvidas;
- Definição de Políticas (Configuração, Manutenção e Controle) para realizar a gestão do sistema;
- Elaboração personalizada dos contratos SLAs (perfil das aplicações);
- Definição de um modelo que realiza o SLM (*Service Level Management*) automático.

Este modelo apresenta algumas vantagens como, aumento de produtividade, reduções significativas do tempo de desenvolvimento, simplicidade de configuração de diferentes equipamentos e aplicações, desenvolvimento de novos serviços, prioridade de tráfego, consistência, entre outros.

1.6 – Estrutura da Tese

Esta Tese está organizada da seguinte forma: O Capítulo 1 apresenta a introdução do trabalho. O Capítulo 2 apresenta os conceitos básicos de temas onde o trabalho está inserido. O Capítulo 3 descreve alguns trabalhos relacionados, procurando descrever o estado da arte em distribuição de conteúdo através do gerenciamento baseado em política em redes sobrepostas. O Capítulo 4 descreve os principais requisitos de QoS exigidos por uma aplicação na Internet avançada. O Capítulo 5 descreve e classifica as aplicações multimídia em categorias com base nos requisitos de QoS exigido por cada perfil de aplicação. O Capítulo 6 apresenta a arquitetura e seus principais elementos e funcionalidades. No Capítulo 7 será apresentada a definição do algoritmo de cálculo de rota do sistema PBQoS. O Capítulo 8 descreve a implementação computacional do sistema PBQoS.

Finalmente, as considerações finais e algumas perspectivas serão apresentadas no Capítulo 9.

Capítulo 2

Conceitos básicos

Este Capítulo apresentará uma breve descrição dos principais conceitos no qual este trabalho está inserido, dentre eles estão: Gerenciamento Baseado em Políticas (PBNM), Redes sobrepostas, Requisitos de Qualidade de Serviço (QoS) e Distribuição de conteúdo multimídia (ao vivo e sob-demanda).

2.1 – Gerenciamento Baseado em Políticas

Nos dias atuais, as redes IP estão se tornando sistemas complexos por serem constituídas por uma heterogeneidade de dispositivos de diferentes fabricantes, protocolos, aplicações e serviços e fazer a interoperabilidade desses dispositivos não é uma tarefa trivial. Dessa forma, algumas tecnologias surgiram em busca de solucionar algumas limitações dos protocolos tradicionais das redes IP, e consequentemente tem adicionado complexidade nas infra-estruturas de rede (Strassner, 2004).

Dessa forma, torna-se claro a necessidade de novas soluções com objetivo de tornar o gerenciamento desses dispositivos mais simples e amigável aos administradores da rede. Pensando nisso, a IETF (*Internet Engineering Task Force*) apresenta o *Policy Framework*, o qual define uma arquitetura básica para implementação do gerenciamento baseado em políticas e modelos para representação dessas políticas.

O *Framework* definido pela IETF surge com intuito de tornar simples o processo de gerenciamento dos recursos e elementos de rede. A simplificação e automação do processo de gerenciamento de rede é uma das aplicações principais do *Policy Framework* (IETF, 2008). Este *Framework* consiste basicamente de quatro elementos básicos conforme ilustra a Figura 1.

Figura 1 – Arquitetura de Políticas definida pela IETF
(IETF, 2008)

- A Ferramenta *Policy Management Tool* é utilizada pelo administrador para definir diferentes políticas que serão aplicadas aos dispositivos da rede;
- *Policy Enforcement Point (PEP)* é uma entidade lógica responsável por aplicar as políticas aos elementos de rede;
- *Policy Repository* é a entidade responsável pelo armazenamento das políticas disponíveis no sistema de gerenciamento. O repositório de políticas é usado para armazenar políticas geradas pela ferramenta *Policy Management Tool*.
- *Policy Decision Point (PDP)* é a entidade responsável por recuperar as políticas do repositório, interpretá-las e então decidir qual política será aplicado pelo PEP aos elementos da rede.

Um modelo de gerenciamento baseado em Política pode ser considerado uma evolução nos princípios de como gerenciar as redes e serviços de distribuição, pois favorece a visão global do sistema de modo a controlar eficazmente as infra-estruturas de uma forma integrada com todas as políticas previamente definidas no sistema. A arquitetura de gerenciamento PBNM (*Policy Based Network Management*) proposto pela IETF apresenta mudanças na forma de como as redes são mantidas hoje em dia. O sistema baseia-se num conjunto de entidades de rede que dinamicamente fazem associações de alguns requisitos de QoS e os mapeiam em um nível de abstração, onde o administrador da rede não precisa se preocupar com detalhes técnicos de configuração dos elementos da rede.

A arquitetura PBNM apresenta algumas vantagens, como o aumento de produtividade, reduções significativas do tempo de desenvolvimento, simplicidade de

configuração de diferentes serviços e aplicações, desenvolvimento de novos serviços, prioridade de tráfego, consistência, dentre outros. A uniformização do gerenciamento desses recursos permite definir políticas de gerenciamento de rede baseados nos requisitos mínimos exigidos por cada tipo de serviço (perfil de aplicação), fazendo com que as requisições por recursos da rede sejam sempre satisfatórios aos usuários do sistema.

2.1.1 – Políticas

Políticas são definidas como um conjunto de regras que podem ser aplicadas em uma entidade de rede para controlar seu comportamento (Oqily, 2007). Estas regras podem ser traduzidas para componentes específicos de políticas que são armazenados em um repositório, podendo ser recuperados e aplicados aos elementos da rede. O objetivo das políticas é estabelecer um método de administração mais amigável à linguagem humana, promovendo o gerenciamento de todo o ambiente, ao invés de tratar da configuração individual de cada elemento ou segmento da rede.

O gerenciamento baseado em políticas apresenta uma nova proposta de redes autônomicas, surgindo como um novo paradigma de administração das entidades de rede e serviços existentes nos dias atuais. O administrador dispõe de mecanismos próprios que o permitem realizar o gerenciamento de forma unificada, deixando de lado a tarefa árdua de administrar individualmente os serviços existentes em uma rede de distribuição. A Figura 2 ilustra a essência de um sistema que possui um modelo de gerenciamento baseado em políticas.

Figura 2 – Modelo de gerenciamento baseado em Política

O administrador de rede se responsabiliza por definir políticas, que irão determinar a forma com que os recursos e serviços poderão ser acessados ou

utilizados. Uma política pode associar informações dos usuários, da rede, dos serviços e aplicações, características de segurança, prioridade de negócios e características de rede. São utilizadas, por exemplo, para determinar se um usuário está autorizado a executar uma aplicação particular, níveis de prioridade, quantidade de recursos de rede que deverão ser alocadas a um serviço, dentre outras aplicações. Um sistema de Gerenciamento baseado em políticas tem como principal objetivo transformar uma política em mudanças de configuração aos recursos da rede (dispositivos e serviços).

Desta forma, o conceito de política pode ser definido como uma representação de um negócio que tem como objetivo ser implementada em um domínio gerenciável e ser aplicada através de um conjunto de regras e ações como ilustra a Figura 3.

Figura 3 – Modelo de política

Um grupo de políticas é composto por uma ou diversas regras, onde cada regra é composta por um conjunto de condições e um correspondente conjunto de ações (lei da causa e efeito). Esta combinação define uma seqüência de ações a serem iniciadas quando o conjunto de condições correspondentes for satisfeito ou não. Cada regra de política consiste de uma expressão booleana que descreve uma situação que será aplicada pela política. Quando a expressão for verdadeira, um conjunto de ações será executado, e quando a expressão for falsa, diferentes ações serão iniciadas, podendo estas ser priorizadas dependendo do contexto e condições em que as mesmas se encontraram. A Figura 4 ilustra um exemplo de como as ações são executadas dependendo da condição pré-estabelecida.

Figura 4 – Causa e efeito - execução de ações

Cada regra de política consiste em dois componentes: (i) um *conjunto de condições* que determinará uma ou mais ações, incluindo alguns parâmetros como, nome de usuário, endereço IP, protocolos e tipo de aplicações e (ii) um *conjunto de ações* que são aplicadas em consequência de uma condição ser satisfeita ou não. Dentre estas ações estão, reserva de banda, configuração de serviços, permissão de acesso, uso do sistema, dentre outras. De acordo com as pré-condições estabelecidas, uma ação pode mudar ou manter o estado de uma determinada entidade ou objeto. O uso de políticas torna o sistema programável, flexível e a modificação ou cancelamento de antigas políticas ou a inclusão de novas políticas permite que o sistema seja facilmente adaptado para atender a mudanças de requisitos. A Figura 5 ilustra alguns exemplos simples do modelo de gestão por políticas (*causa e efeito*).

<p>(a)</p> <p>Causa e efeito 1 (exemplo)</p> <p>IF(condição) THEN Ação)</p>	<p>(b)</p> <p>Causa e efeito 3 (exemplo)</p> <p>IF(sourceAddress = 143.167.111.5 THEN setPriority = 2</p>
<p>(c)</p> <p>Causa e efeito 2 (exemplo)</p> <p>IF(sourceAddress = 143.167.111.5) AND (destinationAddress = 143.167.111.10) THEN setRate = 256 bits</p>	

Figura 5 – Exemplos simples de políticas

Segundo (Beller, 2005), a grande maioria das redes utiliza o protocolo de gerenciamento SNMP (*Simple Network Management Protocol*), que teve sua especificação inicial feita pelo IETF no final da década de 80. O SNMP vem sendo

constantemente atualizado e encontra-se na sua terceira versão (SNMPv3). Na arquitetura de gerenciamento com SNMP, os dispositivos da rede normalmente são configurados individualmente. Este processo demanda tempo e torna o sistema propício a erros, principalmente quando o número de dispositivos cresce. Além disso, a reconfiguração do sistema é infreqüente e executada manualmente.

A estrutura de gerenciamento baseado em políticas representa uma evolução conceitual em relação ao SNMP, pois favorece a visão global do sistema, de modo a controlar eficazmente as infra-estruturas de uma forma integrada com todas as políticas previamente definidas no sistema. A utilização deste conceito possibilita que a rede adapte-se facilmente para atender novos objetivos, estabelecendo um processo adequado para a constante necessidade de atualizações dos sistemas (Beller, 2005).

Os sistemas de gerenciamento usualmente são aplicados aos elementos físicos da rede como, roteadores, *switches* e *gateways*. A Figura 6 ilustra um exemplo em alto nível dos elementos arquiteturais do modelo de gestão por políticas proposto pela IETF.

Figura 6 – Elementos arquiteturais do modelo de gestão por políticas
Adaptado de (IETF, 2008)

Pretende-se neste trabalho apresentar uma arquitetura de gerenciamento que permite realizar as tarefas de administração em redes sobrepostas e seus elementos lógicos como aplicação de distribuição de conteúdo. As Políticas podem ser geradas dinamicamente de um contexto de informação de usuários, redes, ou então através de provedores de conteúdo, alcançando assim um nível de automação que requer pouca interação humana. Isto assegura que as redes sobrepostas se comportarão de acordo com as necessidades específicas de cada aplicação.

2.1.2 – Benefícios do gerenciamento baseado em Políticas

Gerenciar uma rede através de políticas (que descrevem as restrições e as regras que o sistema deve respeitar) trouxe diversos benefícios e facilidades aos administradores de rede. Dentre estes benefícios (Strassner, 2004) destacam-se:

- Alternativa inteligente para distribuir e controlar serviços de rede;
- Provê segurança de forma eficiente;
- Gerenciamento de dispositivos complexos;
- Gerenciamento de serviços e tráfegos complexos;
- Redução de intervenção humana;
- Facilidade de reconfiguração do sistema;
- Distribuição de tráfego de forma mais inteligente;

2.2 – Redes Sobrepostas

Uma rede sobreposta (*overlay*) é uma rede de computador construída sobre uma rede IP com intuito de migrar parte da complexidade de roteamento das camadas inferiores para a camada de aplicação. Este tipo de rede é usado comumente para viabilizar aplicações e serviços na Internet. Como exemplo, as redes P2P (*Peer-to-Peer*) que são provavelmente as redes sobrepostas mais difundidas que existem operam sobre a Internet. Por outro lado, a Internet quando utilizada para serviços tipo *dial-up* também é considerada uma rede sobreposta, pois opera sobre a rede telefônica. Essas redes podem ser construídas para permitir roteamento de pacotes sem especificar o endereço IP do destinatário, isto pode ser realizado através de tabelas *hash* distribuídas onde apenas o endereço lógico

específico é conhecido (Coutinho, 2009). O conceito de redes sobrepostas visa minimizar os problemas de flexibilidade existentes nas redes IP. Através deste conceito, é possível trabalhar com os elementos de rede em nível de aplicação, trazendo além de flexibilidade, meios consistentes que permitam realizar o gerenciamento antes não atingido pelas camadas inferiores.

A estrutura é baseada em uma ou mais funções matemáticas que determinam como os nós são interconectados. A estrutura da rede contribui para um funcionamento eficiente, pois quando um nó falha, os algoritmos da rede recuperam ou recriam outra estrutura apropriada (Doval, 2003).

As redes infra-estruturadas (como Internet e as demais redes IP), por serem definidas para suportar escalabilidade e devido à simplicidade em sua arquitetura, não são capazes de prover *Qualidade de Serviço* (garantia de controle sobre atrasos, variação de atrasos, vazão e taxa de perdas) para as aplicações em um sistema multi-domínio. Como a rede sobreposta forma uma rede lógica sobre a rede física e nesta rede, os nós exercem enlaces lógicos com outros nós através de conexões virtuais, do ponto de vista da rede física subjacente, são caminhos com vários saltos pelos quais o tráfego é roteado de forma transparente para a aplicação. Redes sobrepostas também estão sendo utilizadas para possibilitar o uso de protocolos não suportados pelos roteadores da Internet atual, como a criação de rede IPv6 sobre rotas que suportam somente o IPv4 (Oqily, 2007).

Os nós de uma rede sobreposta podem ser conectados por enlaces virtuais ou lógicos, correspondendo um caminho entre eles. Um enlace lógico não precisa corresponder a um enlace físico da rede física. A Figura 7 ilustra um exemplo de rede sobreposta sendo usada em um cenário P2P.

Figura 7 – Rede sobreposta para P2P
(Coutinho, 2009)

As redes sobrepostas são consideradas fáceis de implementar, se comparadas com a dificuldade de mudanças na infra-estrutura IP, além de permitir o oferecimento de novos serviços de comunicação aos usuários. O modelo de redes sobrepostas foi proposto para melhorar o roteamento na Internet e através de controle de QoS, é possível alcançar altas qualidades em fluxos de mídia nessas redes. As primeiras propostas como o *IntServ*, *DiffServ* e *IP Multicast* não tiveram uma larga aceitação no mercado, pois exigiriam modificações em todos os roteadores da rede. Por outro lado, uma rede *overlay* pode ser desenvolvida gradualmente através de equipamentos que rodam o protocolo *overlay*, sem a cooperação de um ISP (*Internet Service Provider*). Uma rede sobreposta não tem controle de como os pacotes são roteados na rede física, por outro lado, pode controlar, por exemplo, a seqüência dos nós em que a mensagem irá atravessar antes de chegar ao destino (Nascimento, 2009) (Subramanian, 2003).

O conceito de rede sobreposta não é uma idéia nova; Conforme já definido anteriormente, a Internet pode ser considerada uma rede sobreposta por ser desenvolvida sobre a rede telefônica. Diversos nós virtuais tem sido designados na Internet para vários propósitos, incluindo prover conectividade OSI (*Open Systems Interconnection*) na camada de rede, o que facilita, por exemplo, o desenvolvimento serviços como o *IP multicast* (Nakao, 2003).

As redes sobrepostas têm sido projetadas para vários propósitos, e tornam possíveis implementar serviços não disponíveis na rede nas camadas inferiores, por

exemplo, *MBone*, *6-bone*, *X-Bone*, *End System Multicast*, *Overcast*, distribuição de conteúdo, redes *Peer-to-Peer*, detecção e recuperação de falha de forma eficiente, melhorar o desempenho fim-a-fim, facilitar o gerenciamento de serviços, dentre outros (Andersen, 2001b) (Nascimento, 2009). A Figura 8 definida por (Nascimento, 2009) ilustra um exemplo de rede sobreposta.

Figura 8 – Exemplo de uma rede sobreposta
(Nascimento, 2009)

Assim como numa rede IP, as redes sobrepostas apresentam algumas vantagens e desvantagens que serão descritas a seguir:

Vantagens

- Não necessita de adição/modificação na rede física (hardware ou *driver*);
- Não necessita de alteração em software existente, apenas construção de novo software sobre um já existente (em alguns casos);
- Não é necessário modificar nenhum protocolo já existente – *Ethernet Protocol*;
- Provê escalabilidade;
- Desenvolvimento de serviços difíceis de serem desenvolvidos nas camadas inferiores;
- Facilidade de realizar o gerenciamento de serviços;

Desvantagens

- Redes sobrepostas sondam a Internet com frequência, através dos comandos *ping* e *traceroute*, visando mensurar continuamente os atributos dinâmicos como largura de banda, taxa de perda, atrasos e variações de atraso;

- Quando um nó participa de múltiplas redes overlays, o tráfego gerado por um determinado nó pode ser alto;
- Mudanças dinâmicas na rede (entradas e saídas de nós na rede);
- Mudanças nas condições da rede e topologia (atrasos na comunicação entre nós podem variar como o tempo, devido a mudanças de rota, congestionamento, dentre outros);
- Cada nó presente na rede sobreposta deve dispor de mecanismos inteligentes que possam determinar as condições atuais da rede.

2.3 – Protocolo SNMP

O protocolo SNMP (*Simple Network Management Protocol*) é um protocolo de gerência de redes IP, responsável por facilitar o intercâmbio de informação entre os dispositivos de rede. O SNMP possibilita aos administradores gerenciarem o desempenho da rede, encontrar e resolver seus eventuais problemas, e fornecer informações para o planejamento de sua expansão, dentre outras operações.

O modelo assumido pelo SNMP é relativamente simples. Todos os nós (roteadores e servidores) da rede executam uma aplicação conhecida como agente (*SNMP agent*). Este agente escuta a porta 161, espera por requisições de um SNMP manager e envia informações dos nós para o gerente (*SNMP manager*). O gerente é um processo que é executado em um nó da rede e é usado para obter informações de gerenciamento. O gerente e o agente trocam informações entre si usando o protocolo SNMP. Uma rede pode ter mais de um gerente.

Um SNMP manager troca informação de gerenciamento com os agentes SNMP. As informações de gerenciamento dependem do tipo de elementos que estão sendo gerenciados. Para gerenciar uma interface *ethernet*, por exemplo, o administrador deve conhecer algumas informações como, o endereço MAC, endereço IP, tamanho máximo da unidade de transmissão, número de pacotes enviados, taxa de erros, dentre outros.

No SNMP a informação básica de gerenciamento é chamada de objeto. De forma geral, um objeto descreve um atributo variável ou um conjunto de atributos que é usado para propósitos de gerenciamento de rede. Estes objetos são coletados em uma base de informação de gerenciamento denominada MIB (Verma, 2000).

2.4 – Protocolo COPS

O protocolo COPS (*Common Open Policy Service Protocol*) é um protocolo usado para distribuir políticas para diferentes dispositivos. Foi originalmente desenvolvido para ser usado dentro do contexto de controle de admissão RSVP (*Resource Reservation Protocol*), que de certa forma, aloca prioridade de tráfego para aplicações que possuem requisitos de QoS como fluxo de vídeo, ao vivo ou sob-demanda. Extensões deste protocolo têm sido propostas, e pode ser usado também para prover políticas para *DiffServ* ou segurança IP.

O protocolo COPS está sendo proposto como protocolo padrão para troca de informações de políticas entre as entidades PDP (*Policy Decision Point*) e PEP (*Policy Enforcement Point*). O PDP é uma entidade de rede que determina quais políticas serão aplicadas de acordo com o tipo de tráfego (dados, vídeo, áudio, videoconferência, etc), enquanto que o PEP é uma entidade que implementa as políticas determinadas pelo PDP.

O protocolo COPS teve sua origem no trabalho realizado pelo IETF com o objetivo de criar uma nova norma para efetuar a comunicação entre os PDPs e os PEPs. Trata-se de um protocolo simples de pergunta e resposta que permite a troca de informações entre o gerente de política (PDP) e seus clientes (PEPs). Este protocolo é baseado em estados, o que significa que o servidor guarda o estado dos clientes, reagindo sempre que necessário de modo apropriado, mesmo que não exista uma solicitação para o efeito. Utiliza o protocolo TCP como protocolo de transporte para troca confiável de mensagens, fornecendo assim mecanismos de segurança para as mensagens. Além de empregar o modelo cliente-servidor, o protocolo COPS permite o envio de informações de configuração e administração dos elementos (Strassner, 2004).

Cada mensagem trocada com o protocolo COPS consiste de vários tipos de objetos COPS. Estes objetos são estruturas de dados que incluem itens detalhados de uma requisição, identificando assim as requisições de clientes, decisões de políticas, ou outros tipos de informações necessárias dentro de diferentes mensagens (Verma, 2000).

2.5 – Considerações finais

No intuito de dar suporte aos leitores, este Capítulo apresentou os conceitos relacionados ao gerenciamento baseado em políticas (PBNM), redes sobrepostas e os protocolos de gerenciamento SNMP e COPS.

A IETF descreve modelos genéricos para representação de políticas, sugere uma arquitetura para implementação PBNM e especifica os protocolos adequados para transportar as informações de políticas. Um sistema de gerenciamento baseado em política apresenta uma forma de definir regras de negócios em um nível mais alto (camada de aplicação) assegurando que a rede possa fornecer os serviços que os usuários necessitam. O conceito de políticas vem sendo amplamente definido e adotado nos sistemas de gerenciamento de redes, bem como no uso para melhor distribuição de conteúdo. Como desafio, tem-se uma prescrição fim-a-fim de como uma determinada política pode ser usada para gerenciar, controlar os serviços de rede, bem como distribuir conteúdo multimídia por caminhos otimizados de forma eficiente.

As redes sobrepostas estão se tornando cada vez mais populares devido a sua flexibilidade e suas habilidades de oferecerem novos serviços que são difíceis de serem providos pelas estruturas atuais de rede IP. Recentes pesquisas (Verma, 2002), (Subramanian, 2003), (Oqily, 2007), (Uchoa, 2007), (Ushida, 2006) na área têm focado no modelo apresentado pelas redes sobrepostas, principalmente no que tange o processo de distribuição de mídia em ambientes heterogêneos.

Muito difundido na área de gerencia de rede, o protocolo SNMP tem como principal responsabilidade facilitar o intercâmbio de informação entre os dispositivos de rede. Possibilita aos administradores gerenciarem o desempenho da rede, encontrar e resolver seus eventuais problemas, e fornecer informações para o planejamento de sua expansão.

O protocolo COPS vem sendo amplamente usado para distribuir políticas para diferentes dispositivos de uma rede e tem sua especificação passando por um processo de padronização quando o assunto é carregar informações de políticas de gerenciamento. Nos dias atuais, o protocolo COPS está sendo proposto como protocolo padrão para troca de informações de políticas entre as entidades PDP e PEP.

Os níveis de QoS que serão garantidos em uma rede depende do perfil da aplicação que fará uso dos recursos desta rede, pois uma rede pode disponibilizar poucos recursos, podendo assim, impactar a qualidade de uma aplicação. Ouvir um áudio pela Web (aplicações em tempo real) pode necessitar de mais recursos se comparado com o processo de um *download* de um arquivo texto (aplicações sob-demanda). Diferentes aplicações presentes na rede podem exigir diferentes níveis de garantias de recursos, o que são determinados pelos requisitos mínimos exigidos por cada perfil de aplicação. Dessa forma, a arquitetura proposta neste trabalho, tem o propósito de controlar os recursos de uma rede de acordo com as necessidades exigidas por cada perfil das aplicações.

Capítulo 3

Trabalhos Relacionados

Este Capítulo apresenta os principais trabalhos relacionados referentes à distribuição de conteúdo através de Políticas de Gerenciamento com base nos requisitos de QoS. Alguns trabalhos apresentam arquiteturas de gerenciamento baseado em políticas em um contexto de redes IP, enquanto outros trabalhos apresentam modelos de gerenciamento baseado em políticas aplicadas em rede sobreposta.

3.1 – Policy based Management of Content Distribution Networks

O trabalho proposto por Verma (2002) apresenta uma arquitetura baseada em políticas para controlar e gerenciar a distribuição de conteúdos em redes formando uma rede de sobreposição de nós (*proxys*) sobre a rede física. A arquitetura estende o *Policy Framework* usado para controlar o QoS e segurança em serviços de distribuição de conteúdo pela rede. Verma (2002) afirma que o uso de um *framework* baseado em políticas trás como vantagem um mecanismo para gerenciar vários dispositivos a partir de um simples ponto de controle, e analisam diferentes componentes que compõe as regras impostas pelo sistema (condições e ações). O artigo descreve a arquitetura e demonstra como atualizar dinamicamente as políticas de distribuição de conteúdo. Explora e avalia se o paradigma usado pelo *framework* é adequado para controlar a distribuição de conteúdo. Por fim, faz uma análise dos custos envolvidos nessa distribuição.

O sistema apresenta dois repositórios, o de políticas, onde as políticas são armazenadas e um repositório de estados, onde um monitor é responsável por coletar informações sobre o estado atual da rede e armazená-las nesse repositório, sendo ele acessível por vários subsistemas. O servidor fonte configura o sistema de distribuição para atuar em acordo com as políticas de distribuição. Um servidor de borda configura seu ambiente para assegurar que as políticas sejam aplicadas de acordo com o tipo de aplicação. A arquitetura faz uso de um elemento denominado

PEP para aplicar políticas, enquanto o agente de políticas faz o papel de um PDP, ajudando o sistema de distribuição a determinar o conjunto de políticas que deve ser aplicada em cada subsistema. A ferramenta de gerenciamento de políticas é um mecanismo utilizado pelo administrador, em conjunto com as informações de estado da rede, para definir as políticas de distribuição para o sistema. Visando entender o potencial benefício de um sistema baseado em políticas, o trabalho faz uma análise do custo economizado que pode ser obtido em um ambiente de distribuição de conteúdo quando incluso o uso de políticas dinâmicas.

Verma (2002) explorou como as políticas podem ser aplicadas em um sistema de distribuição de conteúdo em redes. Fez uma extensão do *framework* de políticas proposto pela IETF neste novo domínio e propôs uma arquitetura para distribuição de conteúdo baseado em políticas. Além de prover uma taxonomia para especificar políticas de distribuição de conteúdo, fez uma análise do impacto causado por cada decisão no que tange o tempo de resposta de cada usuário no sistema. Para isto, dispõe de um servidor centralizado denominado CDSP (*Content Distribution Service Provider*) para distribuir conteúdo através de políticas previamente definidas. Faz uso de políticas para determinar se um conteúdo pode ou não ser acessado ou distribuído pelos CDSPs. Verma (2002) considera um ambiente simétrico, onde o conteúdo distribuído pela rede consiste de um servidor central de um círculo e um número de *proxies* distribuídos em sua volta com distâncias equivalentes. Embora cada *proxy* sirva a um conjunto de clientes com distâncias equivalentes entre eles, o sistema não faz o controle de QoS na rede gerenciada.

3.2 – CORDENA – Um Framework para gestão de Redes baseadas em Políticas

O trabalho CORDENA proposto por (Teixeira, 2004), é um *framework* de gestão de redes baseado em políticas, concretizando uma abordagem PBNM. O *framework* busca a auto-suficiência e a inteligência das redes, capacitando-as para se auto-configurar de acordo com os níveis de serviços definidos e traduzidos em políticas transversais. Estes níveis de serviços são divulgados a priori à rede por mecanismos de provisionamento¹ (*Policy Provisioning*) com o qual potencializa a

¹**Provisionamento** - adaptação ou implantação de um serviço para um cliente em particular.

autonomia nos nós da rede de acordo com as políticas definidas. Embora a implementação se desenvolvesse com o intuito de ser o mais abrangente possível, para efeitos de definição de escopo, o trabalho se concentra no gerenciamento de serviços diferenciados (*DiffServ*). O trabalho visa melhorar a qualidade de serviço prestado pela rede e otimizar a utilização dos recursos disponíveis.

Teixeira (2004) apresenta um modelo baseado na delegação de funções para os elementos físicos da arquitetura PBNM, como forma de diminuir o tempo de resposta na aplicação das políticas aos elementos de rede e tornar os componentes principais da arquitetura mais independentes. Dessa forma, o sistema garante uma diminuição de alguns riscos inerentes à diminuição da dependência das comunicações com os PDPs, repositórios e busca por melhorias na eficiência da rede, pois realiza a automatização dos PEPs, juntamente com o serviço de monitoração.

Nesta proposta, os PDPs são retirados do intercâmbio com os PEPs nos casos em que estes não apresentam valor agregado ao processo. Os PDPs têm foco nas funções de “*Policy Proxy*”, compatibilizando políticas com equipamentos que não apresentam suporte nativo a protocolos de controle de políticas (ex. protocolo COPS), resolução de conflitos quando ocorrer, aplicações divergentes e realizar a integração das funções de monitoração e controle da rede.

Tratando-se de um trabalho não concluído, os autores destacam as linhas mais importantes incluídas nas especificações técnicas e funcional do projeto. O *framework* CORDENA tem como objetivo abordar questões como organização de base, definição da infra-estrutura, fluxos de processos, objetos sobre os quais o *framework* deve permitir a criação de políticas, modelos de informação, dentre outras funcionalidades. O *framework* CORDENA é um ambiente para o gerenciamento de redes baseadas em políticas, no qual foram destacados os requisitos técnicos e funcionais mais relevantes na sua definição e atual implementação. Neste sentido, a proposta visa capacitar cada vez mais os elementos, para que os mesmos possam agir de forma autônoma ao sistema.

3.3 – Policy based resource management for QoS aware application in heterogeneous network environments

O trabalho proposto por (Hetzer, 2007) é baseado no projeto de pesquisa *European NETQoS*, onde descreve um *toolkit* de gerenciamento baseado em política destinado a configuração e adaptação de recursos existentes em uma infra-estrutura de rede heterogênea (UMTS, WIMAX, WLAN, DVD-T, DVB-H) através de mecanismos de QoS. O foco principal é permitir de forma dinâmica que se possa especificar, adaptar e otimizar a alocação de recursos em um ambiente heterogêneo, considerando os elementos de políticas em diferentes níveis (operadores de rede, provedores de serviço e usuários) derivados dos contratos SLAs. De forma particular, o *toolkit* proposto pretende dar suporte a aplicações multimídia e serviços *multicast* através da Internet (*Wireless* e *Broadcast*).

Embora os autores foquem em mecanismos de políticas de QoS aplicados em diferentes elementos em ambientes de redes heterogêneas, os mesmos consideram as dependências hierárquicas de políticas e leva em consideração as informações baseadas em contexto. O trabalho tem por objetivo configurar e adaptar os recursos de forma otimizada através de aplicações dependentes das preferências de rede em um contexto particular e define uma ontologia para especificar as políticas de diferentes atores.

As políticas são adicionadas no repositório através do gerente de preferências para um determinado usuário e cenário. O gerente de decisão valida alterações no repositório de contexto, visando invocar as políticas automatizadas de adaptação e aplicação dependendo de cada cenário. A adaptação é disparada pelo gerente de decisão baseadas nas mudanças específicas de contexto e informação de repositório de políticas. O *toolkit* permite que usuários possam especificar políticas de QoS, que podem ser dinamicamente adaptadas e melhoradas de acordo com as mudanças contextuais detectadas pelo componente de aprendizado. Dentro deste contexto, o componente de aprendizado usa duas bases de dados, de monitoração e de medidas, ambas alimentadas pela ferramenta de monitoração.

O repositório armazena todas as informações relevantes (relação com o contexto, descrições de SLAs, perfis de aplicações, atores de políticas, etc.) para que o sistema possa determinar qual política que será aplicada. Todas as

informações são mapeadas em ontologias, que definem “classes” que representam os objetos (conceitos) e “propriedades” que expressam o relacionamento entre as classes. O sistema confronta algumas informações previamente relacionadas para melhor atender uma determinada requisição.

3.4 – OverQoS: Offering Internet QoS Using Overlays

O trabalho proposto por (Subramanian, 2003) define uma arquitetura para prover QoS usando redes sobrepostas (OverQoS); Para isto, busca prover serviços de rede aos clientes através da noção de um *Controlled Loss Virtual Link (CLVL)*, que por sua vez tem a responsabilidade de medir a taxa de perda existente nos tráfegos das redes *overlay*. A arquitetura OverQoS utiliza este mecanismo para prover taxas diferenciadas, largura de banda e garantia de perdas, e permite também o uso explícito de algoritmos de controle de congestionamento.

Quando o mecanismo de QoS for provido pela camada IP, o roteador tem total controle sobre os pacotes (*buffers*) e banda, podendo diretamente alocar recursos. Por outro lado, um nó OverQoS pode controlar largura de banda ou perdas na rede IP. Entretanto, se cada roteador OverQoS manipular um fluxo agregado, transmitindo-o em alguma taxa através de uma conexão virtual (camada IP), conectando dois roteadores OverQoS, então ele poderá alocar recursos disponíveis para o fluxo, controlando assim as taxas de cada fluxo na conexão virtual.

A arquitetura define uma conexão virtual formando um caminho em uma rede IP que conecta dois nós virtuais (rede sobreposta). Trata-se de uma conexão unidirecional e transporta tráfego entre esses nós. De forma geral a conexão virtual é caracterizada pela sua capacidade e taxa de perda. A capacidade representa a taxa máxima de entrada que um nó OverQoS pode enviar tráfego sobre a conexão virtual, enquanto que a taxa de perda representa a probabilidade que um pacote é retirada da conexão virtual devido aos congestionamentos.

O sistema OverQoS é baseado em dois princípios fundamentais: controle de perda e controle de recursos agregados. Controle de perda capacita o OverQoS a obter uma qualidade mínima de serviço independente das condições de rede. Para os fluxos agregados, OverQoS pode realizar um controle completo na distribuição de recursos disponíveis (largura de banda, e perdas) entre fluxos individuais.

As redes *overlay* apresentam mecanismos mais simples de desenvolvimento, se comparado com o modelo real presente nas infra-estruturas das redes IP, o que permite a provisão de serviços aos clientes não alcançados pelas redes físicas. O sistema OverQoS permite as empresas construírem suas próprias VPNs para prover serviços de comunicação oferecidos pelos ISPs (*Internet Service Provider*) tradicionais. A chave para prover serviços de forma eficiente no OverQoS é a abstração CLVL, o que permite as aplicações proverem diferenciação de largura de banda por fluxo, assegurar taxa de forma estatística, e implementar algoritmos de controle de congestionamento sem nenhum suporte de QoS nos roteadores. Existem duas propriedades de CLVL que torna possível implementar estes serviços em uma rede sobreposta: controle da taxa de perda e agregação de tráfego. Um importante aspecto da arquitetura OverQoS é que ele não realiza nenhuma alteração em dados ou planos de controle entre um roteador IP e nós OverQoS, ao invés disso, aplica os requisitos de QoS diretamente nos nós OverQoS.

3.5 – Automating Overlay Networks Management

O trabalho proposto por (Oqily, 2007) define uma arquitetura baseada em política guiada por informações contextuais, com objetivo de automatizar o gerenciamento de redes através de uma camada de políticas criada sobre a rede *overlay*. Esta camada é composta por um conjunto de elementos PEPs (*Policy Enforcement Point*) e PDPs (*Policy Decision Point*) que trabalham em conjunto para adaptar dinamicamente o comportamento das redes sobrepostas de acordo com as preferências dos usuários, recursos de rede e provedores de conteúdo. Dentro deste contexto, as políticas são geradas dinamicamente em tempo de execução através de informações contextuais e então aplicadas *on the fly*². O trabalho descreve a arquitetura, suas vantagens e resultados. Tem como meta, automatizar o gerenciamento das redes sobrepostas de forma dinâmica, visando preservar a flexibilidade e benefícios propostos pelas redes sobrepostas.

Trata-se de um trabalho inspirado pelo projeto SMART (Oqily, 2007), desenvolvido para otimizar o serviço de distribuição de mídias. Suporta um modelo de distribuição de conteúdos variados como: *Peer-to-Peer*, *Content Distributed*

² **On-the-fly** – algo criado, ou modificado dinamicamente somente quando requisitado.

Networks, Unicast e Multicast e apresenta o conceito de SSON (*Specific Service Overlay Networks*), sendo ele criado para cada tipo de serviço de entrega de mídia ou grupos de serviços. Este modelo customiza a rede através de requisitos particulares de um serviço (QoS, formato da mídia, concorrência e custos).

Como o projeto SMART não especifica como os SSONs são construídos e gerenciados, nem mesmo como um SSON pode ser adaptado de acordo com as informações contextuais dos usuários, a proposta apresentada por Oqily (2007) visa suprir essa lacuna. O sistema consiste de um plano de controle e um plano de usuário, onde o plano de controle é responsável pela criação, roteamento, adaptação e finalização de um SSON e o plano de usuário contém uma camada denominada *Overlay Support Layer* que recebe e envia pacotes para a rede, e os encaminham para o *overlay*. O plano de controle é composto pelos PEPs, que tem como objetivo controlar os recursos, funcionalidade, aplicar mudanças de configuração baseadas em informações de contexto e usar estas informações para gerar novas políticas dinamicamente. O elemento PDP é instalado em cada domínio para tomar as decisões apropriadas para dar suporte a criação, adaptação e finalização de um SSON. O protocolo COPS (*Common Open Policy Service*) é usado na troca de objetos de políticas entre o PDP e PEP.

Este trabalho apresenta uma arquitetura flexível de gerenciamento baseado em política, onde as políticas são dirigidas por informações contextuais, visando automatizar o gerenciamento de redes sobrepostas. A criação, adaptação e finalização de um SSON são controladas através de políticas. As políticas são geradas dinamicamente através das informações contextuais dos usuários, redes e provedores de serviços e as aplicam de forma *on the fly* nos nós virtuais. Cada nó virtual pode simultaneamente fazer parte de muitos SSON (que representa diferentes serviços), as políticas são organizadas em grupos, onde cada grupo representa diferentes SSON, trazendo assim escalabilidade ao sistema e reduz o *overhead* de gerenciamento.

3.6 – OVERSEA – Towards a Scalable and Effective Architecture for Overlay Networks

O trabalho descrito por Uchoa (2009) propõe uma arquitetura efetiva e escalável voltada para redes overlay, visando prover serviços de rede que não possíveis de serem aplicados diretamente nas redes IP. Em sua arquitetura, o trabalho destaca a camada de serviços básicos BOS (*Basic Overlay Network*), destinados as aplicações e serviços da rede overlay. Esta camada de serviços básicos visa evitar a redundância de serviços e diminuir o controle de overhead na rede. Dessa forma, as aplicações avançadas ganham escalabilidade e provem suporte para serviços avançados, como multicast, QoS, segurança e gerenciamento baseado em políticas. Como prova de conceito, foi desenvolvido um Sistema Multimídia Escalável (SMS) em conformidade com a arquitetura Oversea.

A arquitetura segue dois princípios básicos: custos e o desenvolvimento de serviços especializados devem seguir algumas primitivas. Baseado nesta arquitetura foi designado um SMS para oferecer suporte avançado a aplicações multimídia. Visando prover funcionalidade multicast na camada de aplicação, alguns módulos tiveram que ser desenvolvidos e algumas características mínimas foram consideradas para tornar possível o desenvolvimento de outros serviços overlay, como por exemplo, QoS e segurança. A arquitetura consiste de uma plataforma de gerenciamento para distribuição multimídia com capacidade de oferecer três funcionalidades principais: recuperação de mídia através de agendamento e sob-demanda para múltiplos usuários simultâneos na rede; administração de conteúdo e usuários; e gerenciamento da entrega destes conteúdos.

As funcionalidades são organizadas de uma forma bem definida visando prover escalabilidade para os serviços da rede overlay. Os autores descrevem como que a camada de serviços básicos (BOS) sobre a rede física deve prover algumas primitivas para os serviços da rede overlay (SON), buscando diminuir possíveis redundâncias em informações de controle e garantindo eficiência nas decisões de roteamento.

3.7 – QoS-Aware Overlay Routing with Limited Number of Alternative route candidates and Its Evaluation

O trabalho descrito por (Ushida, 2006) foca no controle de roteamento em redes *overlay* para melhorar o controle de QoS fim-a-fim, pois a chave deste controle usando redes sobrepostas é encontrar uma rota alternativa que possa evitar rotas congestionadas, mesmo existindo uma rota padrão no sistema.

Os autores estimam o quão eficiente pode ser o roteamento em uma rede sobreposta, quando o número de rotas alternativas (candidatas) foi limitado para reduzir custo. Foram utilizados dados de medições reais, demonstrando que os nós intermediários (em trânsito) oferecem ótimas rotas alternativas e não estão distribuídos uniformemente. Um pequeno número de nós intermediários pode oferecer rotas alternativas para um grande número de pares fim-a-fim. Dessa forma, foi estimado o quanto QoS fim-a-fim pode ser melhorado quando propositalmente, limita o número de nós intermediários distribuídos de forma não uniformes. Foi desenvolvido um algoritmo de roteamento em rede sobreposta que é capaz de encontrar uma rota sob a restrição do número de rotas alternativas serem limitada. Neste trabalho cada nó foi conectado diretamente com um dos 18 nós presentes na simulação, espalhados geograficamente, por diversos ISPs. Foram realizadas medições de atraso entre cada possível par de nós através do envio de um ping/s (pacotes ICMP) durante 3 minutos, ou seja, 180 pacotes em uma hora.

Dessa forma, o trabalho mostra algumas avaliações relacionadas ao controle efetivo do roteamento, usando uma rede sobreposta. Os autores avaliaram o QoS fim-a-fim, quando um caminho ótimo possa ser escolhido e como resultado, afirmam que o QoS fim-a-fim é melhor do que uma caminho padrão de uma rede IP.

Um ponto importante a ser considerado, é a limitação do número de nós candidatos. Trata-se de uma importante discussão, pois conforme afirmam os autores, há uma redução no custo do processo de seleção das rotas alternativas. Também foi avaliado o melhoramento do QoS fim-a-fim, quando o número de nós candidatos for limitado. Como resultado, Ushida (2006) afirma que o desempenho encontrado nas sub-rotas ótimas é praticamente o mesmo, no momento em que o caminho ótimo for escolhido de forma apropriado. Com base nestas informações, o trabalho descreve o custo-benefício do algoritmo de roteamento aplicado em redes

sobrepostas, que busca por nós válidos para o processo de encontrar as melhores rotas com a restrição do número de nós candidatos ser limitado.

Como pontos positivos, destacam-se: utilização de roteamento em redes sobrepostas visando melhorar o QoS fim-a-fim sem mudar a arquitetura da rede IP, otimização de QoS fim-a-fim em uma rota visando apresentar melhores condições do que as definidas pela rota padrão. De forma similar, dentre os pontos negativos, os que mais se destacam são: medição apenas de atraso, número limitado de rotas alternativas, número limitado de nós intermediários e o algoritmo desenvolvido não trata as modificações no tráfego ou mudanças no roteamento da camada IP. Quando os nós selecionados sofrerem alterações, leva-se muito tempo para o sistema possa encontrar novos nós.

3.8 – Comparativo entre as propostas estudadas

Um ponto crítico para gerenciar e distribuir conteúdo em arquiteturas distribuídas é a utilização de estratégias que permitam selecionar rotas válidas para atender as necessidades de cada aplicação. E dentro de um determinado contexto, o sistema possa definir uma rota que além de atender as necessidades de uma aplicação, possa também configurar, manter e controlar o sistema com base em requisitos de QoS. Neste sentido, nenhuma das propostas apresentadas (Verma (2002), Hetzer (2007), Teixeira (2004), Subramanian (2003), Oqily (2007), Ushoa (2009), Ushida (2006)), contempla satisfatoriamente esse aspecto, pois não utilizam mecanismos de configuração, manutenção e controle das rotas admitidas pelo sistema de gerenciamento.

Embora Verma (2002) faça uso de políticas para gerenciar redes sobrepostas, o trabalho não utiliza requisitos de QoS para realizar o gerenciamento da rede. Por outro lado, o sistema PBQoS além de propor uma arquitetura de gerenciamento, também define políticas de configuração, manutenção e controle e tem como base técnicas de otimização e leva em consideração os requisitos de QoS mais difundidos como vazão, perdas, variação do atraso e atraso.

O trabalho proposto por Teixeira (2004) define um *framework* para gestão de redes por políticas e opera através de regras de provisionamento, porém, se limita na exploração de técnicas que possam diminuir a comunicação entre os elementos

da rede, resolução de conflitos e concentra esforços na organização de base e infraestrutura. Em adição a este *framework*, o sistema PBQoS poderia ser utilizado no processo de administração e operação de cada elemento, definindo assim o comportamento e uso dos recursos da rede.

Hetzer (2007) apresenta um modelo de gerenciamento por políticas em redes heterogêneas através de requisitos de QoS, porém, o trabalho se concentra no processo de configuração e adaptação de recursos nessas redes. Opera com políticas em diferentes níveis de SLA e utiliza informações baseadas em contexto. O trabalho apresenta grandes semelhanças quando comparado com o sistema PBQoS, pois opera através de políticas e requisitos de QoS e define ontologias que caracterizam diversos contextos em redes heterogenias. Por outro lado, o sistema PBQoS uma vez configurado, também opera em diferentes tipos de rede e tem como principal diferencial uma arquitetura que segue o padrão definido pela IETF e dispõe de mecanismos de otimização para melhor definir uma rota entre dois pontos na rede.

Embora Subramanian (2003) defina uma arquitetura capaz de prover QoS em redes sobrepostas, o trabalho adota a técnica CLVL para criar um tunelamento entre dois nós e o controle de QoS (apenas vazão e atraso) é realizado efetivamente na rede IP. O trabalho PBQoS não só realiza o controle de QoS dos requisitos vazão e atraso, mas também gerencia a variação do atraso e perdas da rede IP e as replicam na camada de aplicação para o efetivo funcionamento da rede sobreposta.

O trabalho proposto por Oqily (2007) define uma arquitetura baseada em política onde as mesmas são geradas dinamicamente através de informações contextuais. Entretanto, o trabalho foca a definição na criação, adaptação e finalização de um serviço denominado SSON. Por outro lado, o sistema PBQoS foca na definição de uma rota ótima para atender as necessidades de uma aplicação. Faz uso de políticas de gerenciamento para administrar o comportamento do sistema. Enquanto Oqily (2007) define a criação de políticas dinâmicas, o trabalho PBQoS foca na construção de rotas dinâmicas visando atender as exigências impostas pelos perfis das aplicações avançadas.

O trabalho proposto por Uchoa (2009) define uma arquitetura escalável voltada para redes sobrepostas e provê meios para que o sistema possa tomar decisões de roteamento. Em adição, o trabalho proposto por Ushida (2006) define um mecanismo capaz de controlar o roteamento fim-a-fim das redes sobrepostas

através de rotas alternativas limitadas. O trabalho tem como foco principal a definição de um algoritmo de roteamento e realiza a medição apenas do requisito atraso. Por outro lado, o trabalho PBQoS não define ou utiliza algoritmos de roteamento, mas constrói rotas com base nas técnicas de otimização, levando em consideração os requisitos de QoS exigidos pela aplicação e o QoS disponível na rede. Para administrar todo o funcionamento do sistema, conta ainda com as políticas de configuração, manutenção e controle.

Como alternativa as propostas aqui apresentadas, o trabalho PBQoS propõe uma arquitetura de gerenciamento baseada em políticas, no qual, através de requisitos de QoS, define uma rota ótima para atender as necessidades de cada aplicação. Além disso, a arquitetura contempla mecanismos de configuração, otimização, manutenção e controle dessas rotas realizando o SLM (*Service Level Management*) de forma automática.

Visando realizar uma breve comparação de algumas das soluções existentes atualmente, no que tange o estado da arte em pesquisa e desenvolvimento voltados para sistemas de gerenciamento em redes de computadores, a Tabela 1 apresenta de forma resumida as principais diferenças entre as propostas estudadas. Os trabalhos abordam arquiteturas de gerenciamento baseado em políticas, redes de distribuição, QoS, roteamento e redes sobrepostas.

Tabela 1– Comparativo entre as propostas estudadas

Trabalhos	Overlay	Framework IETF	Controle Multicast/QoS	PBNM dinâmicas	Distribuição de conteúdo	Taxonomia	Arquitetura centralizada
3.1 [Verma, 2002]	Sim	Sim	Não / Sim	Sim	Sim	Sim	Sim
3.2 [Teixeira, 2004]	Não	Não	Não / Sim	Não	Sim (<i>Genérico</i>)	Não	Sim
3.3 [Hetzer, 2007]	Não	Sim	Sim / Sim	Sim	Sim (<i>Recursos</i>)	Não (<i>ontologia</i>)	Sim
3.4 [Subramanian, 2003]	Sim	Não	Não / Sim	Não	Sim	Não (<i>autônomo</i>)	Não
3.5 [Oqily, 2007]	Sim	Sim	Sim / Sim	Sim (<i>on the fly</i>)	Sim	Não (<i>contexto</i>)	Sim
3.6 [Uchoa, 2009]	Sim	Não	Sim / Sim	Não	Sim	Não	Sim
3.7 [Ushida, 2006]	Sim	Não	Não/Sim	Não	Sim	Não (<i>limite de nós</i>)	Não
PBQoS	Sim	Sim	Não/Sim	Sim	Sim	Sim	Não

Capítulo 4

Requisitos de QoS exigidos pelas Aplicações Avançadas na Internet

Este Capítulo apresenta os principais requisitos de QoS exigidos por uma aplicação avançada na Internet. Busca identificar as aplicações que não podem ser acomodadas pelo modelo de serviço melhor esforço (*best effort*) e analisa a natureza de cada aplicação e seu funcionamento dentro da rede. O estudo aqui realizado tem como foco primário aplicações de áudio e vídeo, onde será feita uma análise dos requisitos das aplicações fim-a-fim como áudio-vídeo conferência, voz sobre IP e fluxo de áudio e vídeo de alta qualidade.

4.1 – Introdução

Durante os últimos anos, a Internet tem crescido de forma considerável e tem mudado de alguma forma a vida das pessoas. Como é conhecida, iniciou-se puramente na academia como uma grande rede de pesquisa, e é agora usada extensivamente para educação, entretenimento, comércio eletrônico, comunicação, entre outras. Embora seja baseada no modelo de serviço *Best-effort* (melhor esforço), sua simplicidade operacional tem acomodado milhões de usuários nos dias atuais e com desempenho aceitável. Neste mesmo período, ocorreu surgimento de novas aplicações e serviços exigindo uma demanda maior pelos recursos de rede.

Embora diversas propostas já estejam em produção e muitas redes dão suporte a QoS, ainda existe uma lacuna a ser resolvida quando se diz respeito aos requisitos precisos que essas novas aplicações necessitam para funcionar com alto ou aceitável nível de qualidade. Uma rede não pode garantir um bom desempenho sem ser monitorada, e através de mecanismos de controle deve realizar um efetivo gerenciamento desses requisitos para adaptar o comportamento das aplicações de acordo com as condições da rede. Dessa forma, é possível definir uma arquitetura de gerenciamento que seja capaz de

melhor compreender como o QoS da rede e das aplicações refletirão na usabilidade de uma aplicação real.

4.2 – Aplicações Avançadas

Nos dias atuais, a Internet é predominantemente usada por serviços convencionais orientados a TCP e as aplicações como Web, FTP e email são enriquecidas com tipos de mídias estáticas como imagens, animações, entre outras. Nos últimos anos, a Internet tem sido usada para transportar conteúdo áudio-visual com qualidade modesta, e está sendo usada também como mídia interativa de baixo custo para comunicação através de áudio e vídeo. No entanto, as pessoas estão começando a concretizar o potencial das aplicações já existentes, sendo elas utilizadas em diferentes disciplinas e contextos através da Internet. É visível o surgimento de uma nova geração de aplicações que irá revolucionar a forma com que as pessoas realizam pesquisa, trabalhos colaborativos e comunicação na rede.

As aplicações avançadas podem oferecer novas oportunidades para comunicação e colaboração, alavancando o ensino e aprendizagem, e melhorar significativamente a forma com que os grupos de pesquisa se reúnem para compartilhar idéias e dados científicos. O uso dessas aplicações facilitará a exploração de novas fronteiras, e buscará resolver os problemas complexos de pesquisa, permitindo a colaboração e experimentação em larga escala, acesso a dados distribuídos, ambientes virtuais de pesquisa (Miras, 2002).

Dados e fluxo de mídia das aplicações avançadas da Internet fazem uso de uma grande demanda de recursos (componentes e dispositivos) presentes em um caminho entre origem e destino, compondo uma rota fim-a-fim entre eles. Hoje em dia existem requisitos para dar suporte ao sistema operacional em tempo-real, novas estratégias computacionais, recursos distribuídos, banco de dados, melhorias nas capacidades de hardware, desenvolvimento de *middlewares* e sistemas de gerenciamento, e com isso, há um aumento da complexidade e capacidades de infra-estrutura de rede.

As aplicações visuais de alta qualidade como, vídeo-conferência, TV de alta definição (HDTV) e mídias ricas, por exemplo, demanda uma quantidade considerável de recursos, principalmente, disponibilidade de banda. Por outro

lado, existem as aplicações que são altamente sensíveis a perda de dados e exigem um tratamento específico de rede através dos requisitos de QoS (vazão, atraso, variação do atraso, perdas, dentre outras). Como o modelo de serviço melhor esforço não suporta nenhum tipo de diferenciação de tráfego, não se pode garantir Qualidade de Serviço no nível desejado por essas aplicações.

Desde que as aplicações avançadas começaram a ser desenvolvidas em larga escala, os requisitos de QoS tem se tornado um mecanismo muito utilizado, visando garantir a qualidade exigida por cada perfil de aplicação. É uma das mais importantes metas que um administrador de rede deve possuir: garantir que uma rede opere com ótima precisão, tenha desempenho e atenda as necessidades dos usuários corretamente (Beuran, 2003) (Filsfils, 2007). A Tabela 2 ilustra a relação básica entre os requisitos de QoS e o tipo de aplicação específica.

Tabela 2 – Relação entre a classificação dos requisitos e a necessidade da aplicação

Requisitos de QoS	Voz Interativa	FTP	E-mail	Vídeo Broadcast	Vídeo Interativo
Exigência de largura de banda	Média	Baixa	Baixa	Alta	Alta
Sensibilidade a perda de pacotes	Média	Média	Média	Média	Média
Sensibilidade ao atraso	Alta	Baixa	Baixa	Baixa	Alta
Sensibilidade a variação do atraso	Alta	Baixa	Baixa	Média	Alta

4.3 – Classificação dos requisitos das Aplicações Avançadas

A classificação dos requisitos das aplicações é uma atividade importante para um sistema de gerenciamento, pois define as necessidades de uma aplicação, levando em consideração a capacidade e limitação da rede que será usada para transmitir seus dados. As aplicações que não fazem uso adequado de acordo com as condições da rede podem causar congestionamentos ou grandes colapsos, reduzindo assim a sua utilização, e sofrer as consequências do seu próprio comportamento.

Entender as necessidades de desempenho das aplicações avançadas é essencial tanto para os engenheiros de rede, como para os desenvolvedores de

aplicações, pois determinam como as aplicações podem explorar os recursos de rede de forma eficiente. Para isto, é necessário que todo o comportamento operacional das aplicações seja cuidadosamente explorado e convertido para uma adaptação própria ou então fazer uso de políticas de gerenciamento para melhor distribuir seus dados. Os mecanismos de controle (adaptação e políticas) são importantes para a administração eficiente das aplicações, uma vez que irá permitir seu funcionamento em uma ampla gama de ambientes de rede, aumentando assim sua aceitação ou comercialização, e conseqüentemente preservar o funcionamento nativo das redes atuais.

Visando mapear as necessidades básicas de uma aplicação através de contratos de serviço, foram investigados os requisitos de QoS exigidos pelas aplicações avançadas da Internet, juntamente com os requisitos de desempenho da rede destinada a estas aplicações para que possam ser operáveis ou possuir qualidade aceitável. É importante conhecer qual comportamento uma aplicação deve ter para que os recursos de rede sejam alocados de forma eficiente e otimizada para melhor acomodá-las e transmitir seus dados. Havendo esse equilíbrio entre os engenheiros de rede e os desenvolvedores de aplicação, haverá sempre um serviço de qualidade para os usuários do sistema.

Para Miras (2002) as aplicações podem ser classificadas em categorias baseada em tarefas que elas executam, por exemplo, tipo de mídia, tipo de acesso, situação operacional (dispersão de usuários geograficamente), características de comportamento de usuários (expectativas, habilidades). Miras (2002) define grupos e classifica as aplicações da seguinte forma:

- Aplicações de dados em primeiro plano (*Foreground Teledata*) – incluem tarefas que requerem acesso direto ou indireto dos usuários, monitor e manipulação de dados, sem haver comunicação interpessoal ou entre homem-máquina.

- Aplicações de dados em segundo plano (*Background teledata*) – são aplicações que manipulam dados sem haver interação direta pelo usuário, elas existem para criar sensação de presença, efeitos, etc. Web câmeras e som de fundo de um ambiente são exemplos deste tipo de aplicação onde os requisitos de qualidade são relativamente baixos.

- Aplicações de tele presença em primeiro plano (*Foreground Telepresence*) – são aplicações que requerem interação e algum tipo de

comunicação entre usuários. Esta comunicação pode ser realizada por dados sensoriais como áudio e vídeo (videoconferência), por representação de avatar, ou pela combinação da ambos. Devido aos requisitos interativos dessas aplicações, elas são um caso particular de sensibilidade a atraso fim-afim e variação de atraso. A taxa de perda da rede deve ser baixa, pois, embora existam técnicas sofisticadas para minimizar o problema de perda de pacotes como FEC (*Forward Error Correction*) ou retransmissões, pode ainda existir atraso extra na rede.

- Aplicações de tele presença em segundo plano (*Background telepresence*) – são aplicações que utilizam poucos recursos do sistema, pois o grau de interação não é um requisito. Por exemplo, alguns usuários de uma videoconferência podem ter participações passivas em determinadas atividades de outros usuários, o que significa baixa taxa de quadros por segundo desses participantes.

- Aplicações interativas – uma atividade interativa é composta por uma ação, seguida de uma resposta apropriada. Esta interação pode ser interpessoal e/ou entre máquinas. Determinar os níveis de interação em uma atividade é extremamente importante, pois estes níveis podem indicar os níveis de tolerância de atraso de uma aplicação. Normalmente este tipo de aplicação exige mais requisitos do que uma aplicação não interativa devido ao nível de prontidão das respostas que são exigidas. Em alguns casos, dependendo do número de usuários, uma atividade interativa pode ser dividida em novos grupos. As aplicações interativas não suportam atrasos elevados e requer mecanismos de controle da variação do atraso e perdas de pacotes, pois podem ter um custo significativo na degradação da qualidade.

- Aplicações não interativas – como este tipo de aplicação faz uso de poucos recursos do sistema e requer requisitos mínimos para funcionar, os requisitos como atraso e variação de atraso podem ser mais flexíveis. Em alguns casos pode haver atrasos sem comprometer a qualidade da aplicação e sem causar incômodo ao usuário. Por outro lado, existe a expectativa de uma mídia de alta qualidade, principalmente devido à natureza dessas aplicações (músicas, vídeos, entretenimentos, dentre outros).

- Atividades entre máquinas – são as aplicações que usam pouca ou nenhuma intervenção humana como parte de sua operação. Dentro deste

contexto está a criação, manipulação, processamento e transmissão de dados entre computadores. Este tipo de aplicação geralmente não requer altos níveis de serviços de rede, os requisitos de qualidade são dedicados não por fatores humanos, mas sim por propriedades exatas de uma aplicação, como sensibilidade a atraso, tempo de entrega (aplicações críticas), ou mesmo, volume de dados a serem trocados entre nós alocados remotamente. É importante ressaltar que atrasos fim-a-fim e a variação do atraso são os parâmetros de desempenho mais cruciais para as aplicações que controlam a transferência de dados, enquanto que a vazão é importante para aplicações que exigem grandes transferências de dados (Miras, 2002). A Figura 9 ilustra a categorização baseada em tarefas de cada aplicação.

Figura 9 - Categorização baseada em tarefas das aplicações

4.3.1 – Aplicações flexíveis e Aplicações adaptativas

Existem aplicações que são mais flexíveis e podem tolerar uma variação significativa de vazão e atraso sem afetar sua qualidade. Essas aplicações são conhecidas como as tradicionais aplicações de transferência de dados como email, HTTP, FTP, dentre outras. Na presença de atraso e falta de vazão, estas aplicações podem sofrer degradação de desempenho, mas a transferência de dados não será afetada pelas condições desfavoráveis da rede. As aplicações flexíveis podem ser categorizadas pelos requisitos atraso e vazão conforme descrição a seguir:

- *Tráfego Assíncrono* – são aplicações como email, onde os requisitos vazão e atraso são bem flexíveis, porém não são consideradas aplicações avançadas.

- *Tráfego interativo (em rajadas)* – são as aplicações como *telnet* ou NFS (*Network File System*). Estas aplicações requerem uma interação *near-real-time*, e de acordo com [Miras, 2002], o atraso aceitável seria entre 200-300 ms.

- *Tráfego interativo (a granel)* – são as aplicações que permitem realizar a transferência de arquivos como FTP e/ou HTTP (transferência Web). Exigem requisitos similares para atraso, mas geralmente optam por alta vazão. Na maioria dos casos, os usuários toleram atraso proporcionalmente ao volume de dados transferidos. As aplicações Web são interativas, mas não são classificadas como serviço em tempo real.

As aplicações como aplicações de tempo real não dispõem das flexibilidades mencionadas anteriormente, são intolerantes aos requisitos de QoS, pois geralmente suportam algum tipo de mídia sensível a QoS. No momento em que estes requisitos não forem atendidos, a qualidade de uma aplicação pode se tornar inaceitável e assim, perder sua utilidade. Por outro lado, existem as aplicações que suportam ou toleram certa degradação dos requisitos de QoS, pois operam dentro de uma faixa pré-definida, garantindo qualidade aceitável e satisfatória. Uma aplicação de vídeo pode tolerar certa quantidade de perda sem resultar em incômodos aos usuários. Para [Miras, 2002], estas aplicações podem ser classificadas da seguinte forma:

- *Adaptativas* – são aplicações que aceitam certa variação de atraso através da criação de *buffers* ou realiza uma adaptação sobre alguns parâmetros como: taxa disponível, taxa de perdas, congestionamentos e redução na taxa de codificação ou taxa de transmissão. De forma geral, estas aplicações são capazes de ajustar suas demandas por recursos dentro de uma faixa aceitável de valores.

- *Não adaptativas* – são aplicações que não suportam nenhum tipo de adaptação, mas podem ser tolerantes a algumas variações de QoS. A qualidade de um fluxo de áudio e vídeo pode ser degradada por perdas, e ainda ser inteligível pelo usuário, ouvinte ou leitor, por exemplo.

4.3.2 – Classes de Aplicações de alto nível

As classes de aplicações de alto nível se dividem basicamente em dois grandes grupos: aplicações de áudio e aplicações de vídeo. Estes dois grupos dispõem de aplicações com caráter interativo e não interativo e serão detalhados a seguir.

4.3.2.1 – Aplicações de Áudio

De acordo com Miras (2002) as aplicações de áudio podem ser classificadas em aplicações interativas e não interativas:

Interativa – comunicação contínua por voz sendo o tipo dominante de comunicação remota. Pode ser caracterizado como *foreground*, interativo e aplicação de tele-presença. Um exemplo deste tipo de aplicação é a telefonia IP, que recentemente tem se tornado uma alternativa competitiva para as redes PSTN (*Public Switched Telephone Network*), principalmente devido a sua simplicidade, viabilidade econômica e o valor agregado pela junção de um computador com o telefone. Tanto na indústria como no meio acadêmico a tecnologia VoIP (*Voice over IP*) tem se tornado o meio mais comum nas comunicações por voz através das intranets corporativas. Como a interatividade é o principal requisito dessas aplicações, níveis baixos de atraso e variação de atraso são requisitos importantes para manter a qualidade da conversação, enquanto que níveis baixos de perdas são necessários para manter a qualidade do sinal do áudio. A Tabela 3 ilustra os requisitos mínimos exigidos por uma aplicação VoIP de acordo com Alves (2004) e (Miras (2002).

Tabela 3 – Requisitos mínimos exigidos por uma aplicação de VoIP

Parâmetro de QoS	Requisito
Atraso	<ul style="list-style-type: none"> • 150 ms – entre ouvir e falar • 150-400 ms – aceitável • Acima de 400 ms – não aceitável
Variação do atraso	<ul style="list-style-type: none"> • <20 ms • >50 ms – inaceitável
Perdas de pacotes	<ul style="list-style-type: none"> • <10%
Vazão	<ul style="list-style-type: none"> • 64 kbps – voz e dados

Manter um sistema de áudio de alta qualidade impõe exigências rigorosas em relação aos requisitos dos fluxos de áudio. Isso ocorre devido algumas aplicações estarem dispersas geograficamente, sendo necessário haver a sincronização entre canais (ex. tele-concerto). Este tipo de aplicação pode necessitar de diferentes requisitos para que o fluxo de áudio mantenha a qualidade desejada. Devido às exigências, requisitos como atraso e variação de atraso são fatores cruciais para este tipo de aplicação, pois alta qualidade significa que vários canais de fluxo de áudio devem ser transmitidos sem compressão, o que aumenta a demanda dos serviços de gerenciamento de rede para sustentar a taxa exigida pela aplicação. Além disso, fatores humanos indicam que os usuários são menos tolerantes a degradação de qualidade quanto ao entretenimento (som, música, vídeos, jogos, dentre outros). A Tabela 4 ilustra os requisitos mínimos exigidos pelo perfil da aplicação de áudio de alta qualidade.

Tabela 4 – Requisitos mínimos exigidos para orquestração de Áudio de alta qualidade

Aplicação	Taxa	Taxa de Perda
<i>Reprodução MP3- estéreo</i>	≥ 128 Kbps	Baixa
<i>AC-3 Dolby sound</i>	1,5 Mbps	Baixa

Não Interativa – em um cenário com fluxo de áudio com qualidade profissional, ter alta qualidade significa ter alta amostragem (16-24 bits de amostragem), ou então, múltiplos canais de áudio (10 ou mais). Para ser equivalente a um CD (*Compact Disc*) a taxa de amostragem deve ser 96 KHz. Para manter a qualidade do sinal original, por exemplo, gravação remota de música ou cenário de distribuição de música, o fluxo deve ser transmitido sem perdas de compressão (*losslessly compressed*). Entretanto, fluxos de áudio podem ter requisitos de aplicações completamente diferentes. Uma vez que a interatividade não é o foco principal, a aplicação pode ser capaz de criar *buffers* (*de-jittering*) e então ser tolerante ao atraso e variação de atraso. Além disso, menor sensibilidade a atraso significa que algoritmos de correção de erros podem ser aplicados para aumentar a robustez no que tange a perda de pacotes em um fluxo. Robustez com relação a perdas pode ser alcançado através de retransmissões. Isto mostra que o requisito mais importante para conseguir fluxos com alta qualidade é sustentar a taxa de transmissão (Miras, 2002).

4.3.2.2 – Aplicações de Vídeo

Assim como as aplicações de áudio, as aplicações de vídeo também recebem duas classificações: Interativa e não Interativa, conforme descrição a seguir:

Interativa – Representam as aplicações com algum tipo de interação entre usuários e/ou sistema. Dentre as diversas aplicações de vídeo interativo, destacam-se: (i) Conferência áudio-visual com alta qualidade. Nos dias atuais, é possível realizar conferências através de computadores pessoais, com custos mais discretos se comparado com o modelo tradicional. Tecnologia baseada em IP e H.323 faz com que as salas, sessões e configurações possam ser criadas e mantidas mais facilmente. Atualmente uma aplicação de videoconferência baseada em Internet acaba sendo restrita a taxas modestas, relativamente baixas, variando de 300 a 500 kbps (Miras, 2002), permitindo assim haver uma comunicação com áudio e vídeo, aplicações de ensino a distância, entretenimentos, tele-medicina e colaboração entre usuários alocados remotamente; (ii) Videoconferência colaborativa com alta qualidade pode ser usada para colaboração entre usuários alocados remotamente, apresentações, reuniões de projeto e reuniões colaborativas. Envolve transmissão de áudio e vídeo em alta definição para oferecer uma experiência colaborativa avançada. (iii) Orquestração de áudio e vídeo é uma aplicação similar a orquestração de áudio com a inclusão de vídeo em tempo real. Estas aplicações podem apresentar eventos artísticos, como danças e orquestras, gravação de músicas através de fontes distribuídas, dentre outras. Por demandar altas qualidades, essas aplicações apresentam controle de atraso fim-a-fim e variação de atraso relativamente baixo para alcançar o desempenho, coordenação e sincronismo esperado. (iv) Aplicações médicas e tele-diagnósticos podem ter um propósito educacional. Uma videoconferência em alta definição poderia ser transmitida com áudio e vídeo, onde um cirurgião ensinaria técnicas médicas de forma remota aos estudantes de medicina. Visando alcançar um bom sincronismo com os equipamentos do cirurgião, os requisitos para tais aplicações são críticos para alcançar uma boa fidelidade e para isto, os requisitos de atraso e variação de atraso devem apresentar valores baixos.

Não Interativa – representam as aplicações sem nenhum tipo de interação entre usuários e/ou sistema. Dentre as diversas aplicações de vídeo não interativo, destacam-se: (i) *Broadcast, streaming* e vídeo sob-demanda – transmissão de vídeo armazenado e ao vivo é muito comum hoje na Internet. Entretanto, eles são confinados com qualidade modesta, baixa resolução e taxas de frames restritas devido às grandes necessidades de recursos que as aplicações de alta definição necessitam. São aplicações que requerem quantias significantes de banda, mas tem requisitos de atraso mais flexíveis. Para Miras (2002) fluxos de vídeo são divididos em duas categorias: Disseminação de eventos ao vivo em tempo real (*broadcasting* ou *Webcasting*), e Fluxo de conteúdo sob-demanda.

Um atributo comum destes serviços é que eles não realizam ações extras, exceto as tradicionais como: iniciar, pausar, retroceder ou avançar e envolve um alto nível de interação ou comunicação interpessoal. Isto significa demandas flexíveis de requisitos de QoS como, atraso, variação de atraso, podendo ser aliviado através dos algoritmos de buferização. As aplicações de tele-dados requerem alta qualidade, assim certa quantia de banda deve estar disponível para este tipo de aplicação. Além disso, baixas taxas de perdas são importantes para manter a distorção da qualidade baixa. Aplicações tolerantes a latência significa que técnicas de perda de proteção, como FEC, ou retransmissão (*Automatic Repeat ReQuest* (ARQ)), pode ser utilizadas para manter a alta qualidade, mesmo na presença de alta taxa de perda de pacotes na rede. (ii) HDTV (*High Definition TV*) é uma tecnologia capaz de atingir as velocidades limites das redes atuais. Um sistema HDTV pode prover altas resoluções no aspecto 16x9, 1920x1080 (1080i) com 60 Hz entrelaçados, ou seja, imagens com qualidade semelhante ou maiores que um DVD (*Digital Vídeo Disc*). Dependendo do algoritmo de compressão utilizado, um sinal HDTV pode ser enviado a uma taxa de 200 Mbps (ITU-T, 2002). A tecnologia HDTV é uma das maiores consumidoras de banda hoje em dia devido a sua alta necessidade de recursos para garantir a qualidade. Esta tecnologia é disponibilizada em diferentes níveis de qualidade para diferentes públicos como a *Consumer-grade (broadcast) quality* (19,2 Mbps), a *Contributor Quality* (40 Mbps MPEG-2), Qualidade de estúdio HDCAM (compressão de 200 Mbps), *Standard Definition TV* (SDTV) (Qualidade semelhante ao DVD a uma taxa de 270 Mbps com equipamentos baseados em

SDI – *Serial Digital Interface*), Raw HDTV vídeo (descomprimido) (taxas de 1.5 Gbps usando o padrão SMPTE-292) e por fim, o formato MPEG (*Moving Picture Experts Groups*) (família de padrões para codificar áudio e vídeo. O MPEG-1 prove qualidade near-VCR³ com taxa típica de 1.5 e 3 Mbps, enquanto o MPEG-2 alcança broadcast TV com alta qualidade com taxas entre 5-30 Mbps. MPEG-4 é um novo padrão que prove altas qualidade e taxas elevadas de transferências de dados. Inicialmente teve um foco comercial em fluxo de vídeo sob-demanda com qualidade maior que MPEG-1, mas com taxa de 300-400 Kbps).

4.4 – Comportamento das aplicações avançadas e requisitos de QoS

Esta seção descreve as propriedades das aplicações avançadas destacando-se os seus requisitos fim-a-fim para que possam operar com alta qualidade (ou qualidade aceitável). Serão descritos os principais requisitos de QoS relacionados ao desempenho da rede, como *vazão*, *atraso*, *variação de atraso* e *perdas* em relação ao comportamento qualitativo das aplicações avançadas (Miras, 2002).

Os requisitos de QoS de rede permitem definir métricas de qualidade através de uma variedade de parâmetros. Entretanto, os modelos de QoS são desenvolvidos através de requisitos de qualidade centrado na rede (vazão, atraso, variação de atraso e perdas). Os desenvolvedores de aplicações avançadas e usuários dessas aplicações requerem modelos de qualidade que são gerados de acordo com suas necessidades (QoE – *Quality of Experience*), e que são expressos por diferentes características de desempenho como tempo de resposta, previsibilidade, percentual de qualidade consistente, expectativa e experiência dos usuários, tarefas de aplicações, entrega com nível de qualidade, dentre outros (Harroud, 1990) (Harman, 1990).

Algumas aplicações podem de forma agressiva tentar adquirir recursos compartilhados de uma rede que não impõe nenhuma penalidade para o “mau uso” desta rede. Dessa forma, este mau comportamento pode comprometer sua

³ *Near-VCR* – Qualidade semelhante ao vídeo-cassete

própria qualidade se comparado com aplicações que são mais cautelosas como as aplicações baseadas em TCP, por exemplo. Inicialmente pode parecer que não ocorre nenhuma influência direta, mas pode causar implicações indesejadas no que tange o desempenho e bom funcionamento da rede. O uso de recursos de forma excessiva e irresponsável pelas aplicações pode causar condições de condicionamentos severos, que inevitavelmente afetará também as aplicações que estão fazendo uso da rede de forma indevida. Existem algumas características importantes com relação ao comportamento das aplicações estando fortemente atrelados a percepção de qualidade, conforme classifica (ITU-T, 2002):

Adaptatividade – aplicações que dispõe de mecanismos que permite sondar as condições de um ambiente de rede e alterar suas próprias características de transmissão de acordo com o ambiente em que a mesma opera. Para isto, reduz seus requisitos de transmissão em períodos de alta demanda (ou congestionamento), sendo capazes de rapidamente perceber a disponibilidade de recursos e os utilizar para aumentar sua qualidade. Dessa forma, estas aplicações não somente utilizam os recursos de forma eficiente, ou mantém um balanço entre os recursos disponíveis e qualidade, como também contribuem para o bem-estar das aplicações concorrentes e estabilidade da rede, partilhando de forma eficiente os recursos de rede disponíveis.

Tolerância – são as aplicações que se fazem presentes em uma ampla gama de condições de recursos oferecidos. Elas podem aparecer quando a banda disponível for variável, ou então quando houver perda de dados. Sob estas circunstâncias, as aplicações devem utilizar os mecanismos que são capazes de amenizar os efeitos desses eventos. Devido à natureza dos fluxos de dados transmitidos por cada aplicação (áudio ou vídeo), existe uma quantia inerente de tolerância para a variação das condições de rede, ou seja, uma faixa em que essas condições podem variar sem afetar a qualidade fim-a-fim.

4.4.1 – Requisitos de QoS de rede

Sabe-se que a caracterização de desempenho de uma rede IP é realizada através dos requisitos de QoS, o qual representam os fatores mais significantes

que influenciam diretamente a qualidade fim-a-fim de uma aplicação, dentre esses requisitos destacam-se:

Vazão – capacidade disponível de banda, ou seja, capacidade de uma rota fim-a-fim ser acessível pela aplicação ou fluxo de dados. Conseqüentemente, o número de bits que é injetado na rede por vários fluxos de uma aplicação deve ser ajustado de acordo com a largura da banda disponível.

Atraso – corresponde ao tempo decorrido na transmissão de um pacote de informação ser transmitido entre origem (transmissor) e destino (receptor). Quanto maior for o atraso, maior será o esforço para o protocolo de transporte trabalhar eficientemente. É causado pela combinação do atraso de propagação da rede, atrasos de processamento e atrasos variados nas filas dos roteadores intermediários entre o nó de origem e nó de destino do pacote. Atraso elevado pode causar indisponibilidade de dados e deixar uma interação em tempo real comprometida, frustrando assim o usuário da aplicação. Os projetistas de aplicações avançadas devem buscar mecanismos que sejam capazes de preservar a boa qualidade dos dados mesmo na presença de atraso.

Variação de atraso (jitter) – variação de atraso pode ser entendida como a variação no tempo e na seqüência de entrega das informações devido à variação dos atrasos da rede. Em termos matemáticos é medido com o valor absoluto do primeiro diferencial da seqüência de medidas de atrasos individuais (Miras, 2002). Em outras palavras, a variação do atraso é um dos principais responsáveis pela degradação de desempenho em sistemas de transmissão de alta velocidade. Quando a variação do atraso é excessiva, os dados podem ser perdidos durante o processo de transmissão, acarretando no aumento da taxa de bits errados. De acordo com (ITU-T-2000), as possíveis fontes de variação do atraso em sistemas de telecomunicações podem ser: interferência intersimbólica, diafonia (*crosstalk*) e o ruído nos meios de transmissão, ruído térmico, imperfeição em circuitos de recuperação de relógio, entre outros. Para resolver este problema, uma aplicação pode realizar um rápido atraso na apresentação de forma suficiente para encher o *buffer* e diminuir ou até mesmo anular a variação do atraso envolvida. A Figura 10 ilustra o efeito da variação do atraso entre a entrega de pacotes na origem e o seu processamento no destino. É importante notar que o efeito da variação do atraso não somente causa uma entrega com

periodicidade variável (*Packet-Delay Variation*) como também a entrega de pacotes fora de ordem.

Figura 10 – Efeito da variação do atraso nas aplicações
(Lima, 2004)

Perda de pacotes – Perda de pacotes pode ser definida como uma fração (ou porcentagem) de pacotes de dados retirados do número total de pacotes transmitidos, que são perdidos de alguma forma ao longo do caminho entre a origem e o destino. Para a maioria das aplicações (especialmente aplicações multimídia) o percentual de pacotes perdidos não é suficiente para medir o impacto causado no quesito qualidade (Lima, 2004) (Miras, 2002).

A probabilidade de perda é muito importante em um sistema de comunicação e é definida por Lima (2004) da seguinte forma:

$$\text{BER} = \frac{\text{número de bits recebidos com erro}}{\text{número de bits transmitidos}}$$

4.4.2 – Requisitos de QoS das Aplicações

De acordo com a especificação definida pela ITU-T (2002), os requisitos de QoS de rede não coincidem necessariamente com os requisitos de QoS relacionados à percepção das aplicações. Da perspectiva do usuário, o desempenho de uma aplicação não está relacionado com detalhes de como o serviço de rede foi desenvolvido e qual o seu desempenho. A deteriorização do desempenho de uma aplicação deverá ser expressa em termos focados na percepção dos usuários finais (QoE), e não na sua origem ou caminho fim-a-fim que a aplicação fará uso. Os usuários finais de uma aplicação observam apenas o atraso, e não conseguem distinguir a causa do mesmo – se esta foi gerada pela rede ou pelo processamento do sistema. Em sua perspectiva, o desempenho da rede e os requisitos de QoS estão embarcados no desempenho da aplicação. Dentre os principais requisitos de desempenho, destacam-se:

- **Vazão** – corresponde o efetivo compartilhamento de banda que a aplicação consome da rede (vazão de dados).
- **Atraso** – corresponde ao atraso em todo caminho (fim-a-fim) que uma aplicação apresenta. O atraso causa um impacto direto na satisfação do usuário. A aplicação não determina seu requisito de atraso quantitativo, nem qualitativo.
- **Disponibilidade e continuidade do serviço** – descrevem os requisitos para um serviço ser ininterrupto com qualidade aceitável. Existem diversos fatores que podem interromper a continuidade de um serviço:
 - **Perda de dados** – nível de percepção do usuário e/ou perda de dados não necessariamente coincidem com perda de pacotes em nível de rede. Muitas vezes a perda está em nível de aplicação (perda da fidelidade do sinal devido à codificação), ou então ser completamente específica de cada usuário.
 - **Variação de atraso** – a variação do atraso pode ser observada por uma aplicação através do uso de *buffer*. A variação excessiva de atraso fim-a-fim pode produzir inconsistências temporais na apresentação do conteúdo ao usuário, como congelamento de um vídeo (vídeo *freezing*) e reinício, causando assim irritação e perda da sensibilidade de presença. Para serviços intolerantes a atraso e variação do atraso pode ser removido por meio de *buffers* de aplicação, que por sua vez, implicará na adição de um componente para controlar o atraso fim-a-fim.
 - **Denial of Service (DoS) ou segurança** – DoS causa indisponibilidade do serviço, mesmo que a infra-estrutura de rede suporte os requisitos de QoS exigidos pela aplicação. Segurança é também considerada um importante requisito em nível de aplicação, por ser capaz de fornecer um serviço de comunicação seguro (reuniões privadas, transações seguras, dentre outros).

A recomendação de Miras (2002) é que os requisitos de QoS das aplicações (desempenho), devem ser obtidos através de testes de usabilidade ou através de outros métodos específicos. Isto pode ser obtido levando em consideração a escolha dos valores de QoS apropriados para cada tipo específico de rede.

4.5 – Aplicações que necessitam de QoS

Inicialmente, é necessário considerar que não são todas as aplicações que realmente necessitam de garantias fortes e rígidas de qualidade de serviço para que seu desempenho seja considerado satisfatório. Dentre as diversas aplicações já identificadas neste trabalho, as aplicações multimídia são, normalmente, as que têm maior exigência de QoS, devido à sensibilidade de cada tipo de aplicação. O requisito mais básico e o mais presente nas especificações de QoS, certamente é vazão.

Uma aplicação multimídia *offline* que envolva, por exemplo, dados, gráficos e arquivos com animação, não necessita de sincronização, nem de cuidados especiais com os requisitos de QoS da rede. Uma aplicação de dados com animação, em termos gerais, necessita de uma determinada quantia de vazão, mas não exige o controle de atrasos, sincronização ou tempo de resposta, que de certa forma, é atendida pelo próprio projeto da rede. Uma aplicação multimídia de áudio conferência não é suficiente garantir apenas a vazão. Os atrasos de comunicação e as perdas de pacotes influenciam na interatividade dos usuários e na qualidade da aplicação. Pensando assim, é necessário que a rede forneça o mínimo necessário de QoS para que as aplicações possam operar de forma satisfatória.

4.5.1 – Aplicação de Voz Interativa

Uma aplicação de voz interativa é bastante difundida na literatura mais conhecida como *Voz sobre IP* (VoIP). Este tipo de aplicação é caracterizada por ser extremamente rigorosa com relação aos requisitos de QoS. A necessidade por vazão não é rigorosa em sistemas VoIP, pois acomoda um fluxo de voz em 64 Kbps ou menos; hoje em dia os algoritmos de codificação são sofisticados o suficiente para reduzir a banda para um número ainda menor. Nesta seção serão discutidos os efeitos de atraso, variação de atraso e perdas de pacotes em aplicações de voz sobre IP, requisitos cruciais para o funcionamento satisfatório desse tipo de aplicação.

A qualidade medida de uma aplicação de voz é muito subjetiva, e pode ser expressa primeiramente com respeito ao usuário individual (QoE). Para Miras (2002) a qualidade de VoIP é afetada principalmente por três fatores:

- **Clareza** – refere-se à fidelidade, clareza e inteligibilidade do sinal de voz;
- **Atraso** – refere-se ao atraso existente no caminho percorrido pelo pacote entre origem e destino;
- **Eco** – originado quando a voz do emissor “vaza” do canal de transmissão para o canal de recepção, não havendo a separação do sinal enviado e recebido. Quando a diferença do tempo entre a voz original e o eco for menor que 25-30ms, não causará nenhuma interrupção na conversação. Entretanto, como um atraso na Internet frequentemente excede o limiar de 30 ms, a técnica de cancelamento de eco é tipicamente usada.

Considerando a dificuldade em categorizar esses efeitos que atingem a qualidade individual de cada usuário, o estudo aqui realizado procura descrever quais efeitos qualitativos os requisitos de QoS tem sobre VoIP, e então sugere-se valores desejados (aproximados) para estas métricas com base nas definições propostas por ITU-T (1998).

4.5.1.1 – Efeito de Atraso

O atraso é introduzido em todos os estágios de um sistema VoIP, dispositivos de captura de som, módulos de codificação e empacotamento, transmissão na rede, *buffers* de recepção, decodificação e *playout*⁴ do sinal. No entanto, para o usuário não importa onde o atraso foi introduzido, a rede deve apresentar baixos índices de atraso para que possa ter interação real e a comunicação possa ser realizada. O atraso afeta principalmente a qualidade da conversação e a fidelidade da voz.

De acordo ITU-T (1998), os usuários não podem notar os efeitos de atraso se ele estiver entre 100-150ms. Atrasos entre 150 e 200ms são percebidos como uma rápida hesitação na resposta do parceiro da conversação. Por outro lado, quando esses atrasos excedem 300ms, ficam evidentes para os usuários e a conversação pode se tornar praticamente impossível de ser realizada.

⁴ Em sistemas *Broadcasting*, **playout** é um termo usado para transmissão de canais de radio ou TV em uma rede que os entregam para os usuários.

Sobreposição de voz se torna um fator significativo se o atraso em uma direção (*one way*) for maior que 250ms. Para alcançar alta-qualidade de voz, [ITU-T, 1998] recomenda que o atraso máximo desejado (*one-way*) seja 150ms. Para manter a interatividade natural da comunicação Vegesna (2001) sugere atraso fim-a-fim (*one-way*) de 100ms, enquanto Kumar (2001) define que para manter uma conversação de voz *full-duplex* o atraso (*one-way*) desejado deve ser menor que 300ms. A Tabela 5 ilustra as diretrizes de atraso para uma aplicação VoIP.

Tabela 5 – Diretrizes de atraso para VoIP

Atraso (<i>One-way</i>)	Efeito sobre a qualidade percebida
<100-150 ms	Atraso não detectado
150-200 ms	Qualidade aceitável, porém um rápido atraso é notável
Acima 250-300 ms	Atraso inaceitável, conversação normal impossível

4.5.1.2 – Efeito da variação do atraso

A variação do atraso é o resultado da variação do tempo de chegada entre os pacotes. Para minimizar esta variação um *buffer* pode ser usado para que os pacotes recebidos sejam armazenados (bufferizados) para serem lidos em uma taxa nominal. O tamanho dos *buffers* usualmente se adapta instantaneamente as condições da rede. Entretanto, a chegada dos pacotes com atrasos elevados são descartados e considerados perdidos (isso causa falhas na conversação) ou obstruem a reconstrução dos pacotes, gerando confusão na conversação (Miras, 2002). A Tabela 6 ilustra as diretrizes da variação de atraso e o efeito da qualidade percebida para as aplicações de VoIP.

Tabela 6 – Diretrizes de variação de atraso para VoIP

Variação de Atraso	Efeito sobre a qualidade percebida
<40 ms	Não detectável
40-75 ms	Boa qualidade, mas atrasos ou confusões notáveis
Acima 75 ms	Inaceitável; muita confusão na conversação

4.5.1.3 – Efeito de Perdas de Pacotes

Sabe-se que medir o impacto que a perda de dados tem sobre a qualidade percebida é uma tarefa difícil de ser realizada, pois depende de alguns fatores como *codecs* de áudio usado, existência de proteção ou correção de erros (FEC),

padrões de perdas de pacotes, dentre outros fatores. A perda de pacotes individuais é menos problemática do que as perdas em rajadas com grande quantidade de dados, pois as técnicas de reparação podem ser empregadas para recuperar o pacote perdido de forma isolada, o que não ocorre em grandes séries de perdas consecutivas.

O efeito de perda refletido na qualidade percebida depende da localização da perda em um *bitstream*. Perdas em segmentos em que não possuem dados de voz terão pequenos impactos na qualidade percebida. Se a perda ocorrer em um seguimento de voz, os danos devem ser considerados. O efeito da perda de pacotes depende do seu tamanho; quando pequeno (20ms de áudio), o impacto da perda pode ser facilmente amenizado com técnicas simples de descoberta de erros no receptor. Entretanto, é mais difícil de ser alcançado quando pacotes maiores são transmitidos (80-160ms). A Tabela 7 ilustra o efeito da perda de pacotes de voz através dos codificadores de áudio comumente mais utilizados de acordo com a classificação feita por Miras (2002).

Tabela 7 – Efeito de perdas de pacotes na qualidade de voz

Codificadores	Atraso (ms)	Perda de Pacote (%)	MOS⁵
G.711 w/o PLC	150	1	3.55
G.711 w PLC	150	1	4.31
G.711 w/o PLC	150	2	3.05
G.711 w PLC	150	2	4.26
G.729A + VAD	150	1	3.99
G.729A + VAD	150	2	3.82
G.7231A + VAD	150	1	3.82
G.7231A + VAD	150	2	3.60
G.711 w PLC	400	0	3.60

Na maioria dos casos a qualidade de voz pode ser alcançada se o atraso (*one-way*) for mantido abaixo de 150ms e a taxa de perdas abaixo de 2%. A recuperação de pacotes pode resolver o problema ora apresentado através de técnicas de recuperação e correção de erros, mas como efeito colateral, pois o uso destas técnicas constantemente aumentará o atraso fim-a-fim. É importante ressaltar que além dos fatores mencionados anteriormente, o maior prejuízo pode ser causado pela seleção específica de *codec* de áudio. Estes prejuízos ocorrem devido à distorção introduzida pelo *codec* e pela interação dos efeitos entre rede e

⁵ MOS (*Mean Opinion Score*) – Escala usada para medir a qualidade da voz em uma escala de 1 a 5.

codec. Miras (2002) resume na Tabela 8 os codificadores mais comuns com os 3 mais importantes fatores de impacto na codificação da voz: taxa, qualidade e complexidade.

Tabela 8 – Padrões de codificação de áudio

Codificadores	Descrição	Taxa (Kbps)	Qualidade (MOS)	Complexidade
ITU-T G.711	PCM (<i>Pulse Code Modulation</i>) para frequência de voz; 4 kHz de banda	64	>4	-
ITU-T G.722	SB-ADPCM – <i>Sub-Band Adaptive Differential Pulse Code Modulation</i> ; 16 kHz de amostragem de frequência	48-64	3.8	baixa
ITU-T G.723.1	Codec com taxa dupla (<i>dual rate speech codec</i>) para aplicações multimídia (MP-MLQ/ACELP)	6.4/5.3	3.9	Alta
ITU-T G.726	<i>Adaptive Differential Pulse Code Modulation</i> (ADPCM)	32	3.8	Baixa
ITU-T G.728	Banda de 4 kHz; baixo atraso CELP (LD-CELP) G.727H: taxa variável LD-CELP	16	3.6	Baixa
ITU-T G.729(A/B)	<i>Conjugate Structure Algebraic CELP</i> (CS-ACELP) G.729A: <i>Reduced Complexity Algorithm</i> ; G.729B: <i>Discontinuous Transmission</i> (DTX)	8	3.9	Media
ETSI GSM 06.10 (GSM FR)	Codec de voz <i>Full Rate</i> (FR) (RPE-LTP): <i>Regular Pulse Excitation – Long Term Prediction</i>	13	3.5	Baixa
ETSI GSM 06.20 (GSM HR)	Codec de voz <i>Half Rate</i> (HR) (VSELP: <i>Vector Sum Excited Linear Prediction</i>)	5.6	3.5	Alta
ETSI GSM 06.60 (GSM EFR)	Codec de voz <i>Enhanced Full Rate</i> (EFR) (ACELP: <i>Algebraic CELP</i>)	12.2	>4	Alta
ETSI GSM 06.70 (GSM AMR)	Codec de voz ETSI <i>Adaptive Multi-Rate</i> (AMR)	4.8-12.2	>4	Alta
Nokia AMR-WB	Proposta da Nokia para <i>Codec Adaptive Multi-Rate</i> (AMR) banda larga	12.6-23.85	>4	Muito alta

4.5.2 – Aplicações de Áudio

O termo áudio refere-se a algum tipo de sinal de som que não seja explicitamente voz, por exemplo, música. Os termos *voz* e *áudio* são distinguidos porque existem importantes diferenças entre eles. Em geral, um sinal de áudio requer alta quantidade de banda⁶ e altas taxas de transmissão são necessárias. Dependendo da aplicação (interativa ou não), o controle do atraso deve ser mais rigoroso.

Em contraste às expectativas das aplicações de voz, os usuários sempre esperam alta fidelidade no áudio, e geralmente não toleram a degradação de qualidade. Para uma música ser aceitável, ela deve ter uma qualidade de FM (*Frequency Modulation*), preferencialmente com qualidade de CD ou maior, isto é, o que os usuários estão habituados hoje em dia. A Tabela 9 ilustra a relação entre qualidade, taxa de transmissão para áudio e tipo de especificação utilizada (Silveira, 2009).

Tabela 9 – Relação entre a qualidade de áudio e Taxa de transmissão

Qualidade de Áudio	Especificação	Taxa de Tx
Áudio com qualidade de voz	1 canal Amostras de 8-bit / 8kHz	64 Kbps
Voz digitalizada	Padrão G.728, 3.4kHz	16 Kbps
Áudio monofônico	1 canal Amostras de 16 bit/ 44.1kHz	705,6 Kbps
Áudio com qualidade de CD	2 canais Amostras de 16-bit/ 44.1 kHz	1.4 Mbps
Áudio codificado com MPEG	Equivalente a qualidade de CD	384 Kbps

4.5.3 – Aplicação de vídeo

Vídeo digital é uma realidade cada vez mais presente na vida das pessoas. Uma transmissão de vídeo oferece a oportunidade de realizar a mais avançada forma de comunicação, colaboração e entretenimento, adicionando demandas e altas exigências de recursos de rede se comparado com outras mídias. A Tabela 10 ilustra a classificação proposta por Miras (2002) dos requisitos exigidos pelos formatos mais populares de vídeo.

⁶ Sinal de voz pode ser restringido a uma banda de 4 kHz, com praticamente nenhuma perda.

Tabela 10 – Requisitos típicos de banda para os formatos de vídeo mais comuns

Formato do Vídeo	Requisitos típicos de banda
HDTV descomprimido	1.5 Gbits/seg
HDTV, formato intercalado	360 Mbits/seg
Definição padrão – SDTV, SMPTE	270 Mbits/seg
MPEG-2 comprimido 4:2:2	25-60 Mbits/seg
<i>Broadcast</i> com qualidade HDTC (MPEG-2)	19.4 Mbits/seg
MPEG-2 com definição padrão (SDTV)	6 Mbits/seg
MPEG-1	1.5 Mbits/seg
MPEG-4	5 Kbits/seg – 4 Mbits/seg
H.323 (H.263)	28 Kbits/seg – 1 Mbits/seg
Vídeo NTSC (640x480) 24 bits/pixel	27 Mbps
Vídeo HDTV (1280x720) 24 bits/pixels	81Mbps

Os requisitos de QoS de um fluxo de vídeo são determinados primeiramente pelo contexto da aplicação. Caso a aplicação seja de tele-presença ou tele-dados e o nível de interatividade envolvido seja alto, estes fatores exigirão diferentes manipulações nos requisitos de QoS da rede. Dessa forma, duas grandes modalidades de aplicações de vídeo em rede foram identificadas: Vídeo Interativo e Vídeo *Streaming*.

4.5.3.1 – Vídeo Interativo

Vídeo interativo geralmente é usado em aplicações de tele presença como, videoconferência, ambientes virtuais colaborativos, ensino à distância, aplicações médicas (cirurgias remotas), aplicações científicas (exploração de dados emissivos, controle remoto de instrumentos científicos). Para estas aplicações, baixo atraso é um requisito essencial se a interatividade deve ser mantida. As aplicações de videoconferência com taxas entre 384-800 Kbps por fluxo, freqüentemente são usados em conferências típicas H.323, enquanto que as taxas entre 2-10 Mbps são utilizados para videoconferência baseada em MPEG-2, entre 19.2 Mbps – 1.5 Gbps para aplicações de tele presença com qualidade HDTV. A

Figura 11 ilustra uma adaptação da classificação proposta por Miras (2002) com relação aos requisitos típicos de vazão para serviços de vídeo interativo comum.

Do lado direito da Figura 11 estão os serviços de vídeo interativos sensíveis a atraso, com requisitos mais rígidos no que tange perdas de dados

(desktop, salas de vídeo conferência, interatividade com alta definição). Do lado esquerdo estão os serviços mais flexíveis a atrasos e menos tolerantes a perda de pacotes.

Para se ter interatividade neste tipo de aplicação, um controle rígido de atraso é requerido. Algumas regras gerais são aplicadas aos requisitos para vídeo interativo:

- Baixo atraso fim-a-fim;
- Usuários podem tolerar certa quantidade de distorção no vídeo em aplicações de tele-colaboração, pois a tolerância de perda de pacotes pode ser alta (se comparado com cenários de vídeo *streaming*) com qualidade aceitável e/ou o requisito de vazão pode ser mais flexível, e a adaptação da taxa pode ser usada para reduzir a vazão.
- Sincronização entre fluxos de áudio e vídeo (sincronização de lábio) é essencial para manter boa inteligibilidade;
- Pequenas e delimitadas variações de atraso são considerados requisitos importantes para operações remotas.

Figura 11 – Vazão e interatividade para serviços comuns de vídeo

Adaptado de (Miras, 2002)

A tecnologia empregada nas aplicações de vídeo interativo determina as taxas de perdas aceitáveis. Um vídeo H.323, por exemplo, é muito sensível a variação do atraso e perdas, enquanto que uma aplicação IP de áudio/vídeo pode resistir significativamente a altas taxas de perdas (acima de 10-15%) através das técnicas de redundância. Dessa forma, a sincronização de áudio e vídeo adiciona um requisito extra para o tempo de entrega de todos os fluxos envolvidos, o que deve ser calculado por uma aplicação de gerenciamento eficiente para melhor controlar estes requisitos e distribuir os recursos de forma personalizada para cada tipo de aplicação.

4.5.3.2 – Vídeo Streaming

Vídeo *Streaming* refere-se ao transporte em tempo real de vídeo ao vivo ou armazenado (sob-demanda). Recuperação de uma notícia, transmissão de vídeo ao vivo ou informação multimídia em *broadcasting* (*Webcasting*), TV fechada, vídeo sob-demanda, produção em estúdio baseado em rede e HDTV (exceto aplicações que fazem *download* de conteúdo antes de ser exibido) são exemplos deste tipo de aplicação.

Nas aplicações de vídeo *streaming*, interatividade não é a característica principal, exceto pelas funções de controle VCR (*Videocassete Recorder*) como pausar, parar, avançar, retroceder, etc.

Um *streaming* de vídeo com qualidade modesta (*Real Vídeo*, *Microsoft Advanced Streaming Format*, *QuickTime*) frequentemente envolvem baixa expectativa de vazão, variando de algumas dezenas de Kbps a 1 Mbps. Por outro lado, a expectativa de um fluxo de vídeo sob-demanda é possuir qualidade igual ou superior ao VHS (*Vídeo Home System*). De forma geral, a codificação MPEG-1 e MPEG-2 tem sido as mais utilizadas nos últimos tempos. Recentes evoluções do padrão MPEG-4 oferece qualidade similar com taxas mais baixas, além da oportunidade de criação e manipulação de novos tipos de apresentações (vídeo sintético, animação, integração de cenas naturais e sintéticas) oferecendo assim, muitas formas de manipular conteúdo visual. Para este tipo de aplicação, a vazão pode variar entre 1-3 Mbps na codificação MPEG-1 e até 10 Mbps para

codificação MPEG-2. O formato HDTV pode ser usado em vídeo de alta qualidade produzido em estúdio ou mesmo TV *broadcast* de alta qualidade e dependendo da compressão usada, a vazão pode variar entre 19.2 Mbps e 1.5 Gbps (Miras, 2002).

A principal exigência dos usuários que fazem uso das aplicações de *streaming* de vídeo não interativo é apresentar qualidades visuais elevadas. Dado que o atraso inicial pode ser usado para criar um *buffer* no receptor para acomodar elevados valores de variação de atraso, isto significa que o sistema terá perdas de dados dentro de uma faixa previamente delimitada (menor que 2-3%).

Hoje em dia, um *streaming* de vídeo é confinado em uma qualidade modesta com resolução restrita. Em cada cenário a distorção introduzida pela codificação comparativamente com baixa taxa e perdas de pacotes são usualmente tolerados pelos usuários. No entanto, os serviços avançados de vídeo sobre IP usam altas resoluções onde a distorção de codificação ou perda de informação é facilmente visualizada. O sistema deve garantir a banda e manter a taxa de perda relativamente baixa, pois o usuário possui experiência similar aos serviços não IP, o que dispõe de qualidade mais elevada como a TV digital, transmissão via satélite, DVD e VCR. Como a tolerância de perda de dados é muito baixa em cada serviço, o requisito perda deve ser mantido sempre o mínimo possível.

O requisito atraso é mais flexível para este tipo de aplicação, pois existe pouca interação envolvida. Pode ser usado para construir *buffers* de recepção, visando aumentar a resistência das aplicações para alguma variação no tempo de chegada entre os pacotes e também resolver o problema dos congestionamentos temporários. De forma geral, a transmissão com atraso deve manter-se controlada a fim de limitar a variação do atraso.

Algoritmos eficientes de transmissão/escalonamento podem ser aplicados para maximizar a utilização da rede e melhorar a qualidade das aplicações. Além disso, mecanismos avançados de proteção de perdas podem ser usados para reduzir os efeitos causados pelas perdas que resultam em erros de transmissão e perdas de dados.

Para os dois tipos de transmissão de vídeo, os requisitos de QoS são determinados através de uma combinação entre as necessidades de uma

aplicação e a capacidade da rede em oferecer este serviço. A qualidade do vídeo no receptor é uma função complexa e abrange diversos fatores como, taxa disponível no canal, taxa de quadros, resolução da imagem, dentre outros. Os requisitos de atraso e variação de atraso influenciam diretamente na interatividade da aplicação e no tempo de decodificação do vídeo. Com base nestas informações, é cada vez maior a necessidade de um modelo que permita realizar o efetivo gerenciamento desses requisitos visando determinar os melhores recursos para evitar perdas excessivas e comprometer significativamente a qualidade visual do sinal.

4.6 – Requisitos que alteram a qualidade

A tarefa de especificar os efeitos que os requisitos de QoS de rede exercem na qualidade do vídeo é uma tarefa difícil porém interessante. Os requisitos como vazão, atrasos, variação de atrasos e perda de pacotes diminuem a qualidade percebida ou a fidelidade do conteúdo recebido. Como consequência, esses requisitos afetam diretamente as tarefas das aplicações. Em aplicações de *streaming*, por exemplo, altos níveis de distorção e flutuação na qualidade são notáveis pela observação humana, e podem causar o não entendimento de um vídeo ou perda da inteligibilidade do conteúdo. Em aplicações colaborativas de tele-presença o resultado de perdas em sincronização entre *streaming* pode causar dificuldades na comunicação ou até mesmo causar paradas na comunicação. Operações de controle remoto podem ser afetadas severamente, quando o usuário estará respondendo a uma imagem de vídeo em tempo defasado devido aos altos atrasos ou variação do atraso.

As seções subseqüentes descrevem os efeitos que os parâmetros de QoS podem causar em vídeo *streaming* ou interativo. Embora estes efeitos tenham sido categorizados para cada uma desses parâmetros individualmente, pode ser notado que eles não afetam a qualidade de forma independente. Eles atuam em combinação ou cumulativamente, e em ultimo caso, somente este conjunto de efeitos é detectado pelos usuários.

4.6.1 – Taxa de Transmissão

Sabe-se que um fluxo de vídeo digital com alta qualidade consome quantidades elevadas de banda e conseqüentemente, a vazão que um *streaming* de vídeo recebe predomina na qualidade visual do vídeo. Nesta seção, será mostrada a relação entre qualidade e vazão e a evolução da qualidade do vídeo em função da taxa de codificação. Miras (2002) realizou uma implementação visando avaliar as métricas de qualidade do vídeo através de modelos computacionais que medem a qualidade de uma seqüência do vídeo, visando produzir resultados semelhantes aos obtidos pelos humanos. A escala da qualidade varia em uma faixa de 1 a 5. Foram codificadas 3 seqüências de vídeo: 150 quadros em formato CIF com tamanho de 352x288, com diversas taxas variando entre 256 Kbps a 3 Mbps. A Figura 12 ilustra um comparativo entre a qualidade do vídeo e a taxa de codificação de 3 seqüências de 150 frames (*akiyo*, *news* e *rugby*) com resolução de 352x288, utilizando dois diferentes codecs, H.263 (esquerda) e MPEG-1 (direita).

Figura 12 – Qualidade do vídeo x taxa de codificação
(Miras, 2002)

A qualidade do vídeo é uma função convexa que aumenta a taxa de codificação. Existe uma faixa inicial de banda onde a qualidade do vídeo aumenta substancialmente. Depois disso, a qualidade satura; alguma taxa extra oferecida ao codificador fornece apenas ganhos marginais de qualidade e é provavelmente uma imprudente utilização da banda. Conhecer o ponto onde a qualidade do vídeo começa a se degradar é extremamente importante para determinar a vazão e controlar a qualidade do vídeo.

4.6.2 – Perda de pacotes

O efeito da perda de pacotes em *streaming* de vídeo comprimido é particularmente perturbador, pois as distorções causadas à imagem são tipicamente mais desagradáveis aos usuários do que outros tipos de deficiências, como codificação de artefatos. Essas perdas ocorrem devido à compressão aplicada ao fluxo, causando redução no número de bits requeridos para a transmissão do fluxo, removendo algumas redundâncias inerentes ao conteúdo do vídeo. O efeito da perda de pacotes depende de alguns fatores, como: técnica de compressão usada, seleção de parâmetros codificados – taxa de compressão, taxa de perda, padrões de perdas e tamanho dos pacotes.

Visando reduzir as redundâncias temporais e alcançar melhores taxas de compressão, a maioria dos *codecs* usam a quantização e a compressão *inter-frames* compensando a exploração de movimento e estimativas. A perda de pacotes não somente afeta um *frame* em particular, mas uma perda em um *frame* de referência se propaga para todos os *frames* de dependências. Este fenômeno é chamado de propagação de erros (ITU-T, 1998). O impacto da perda de pacotes é alto se os pacotes são grandes. Pacotes pequenos causam menos distorção, mas por outro lado, mais pacotes serão transmitidos, o que aumenta o cabeçalho e *overhead* e conseqüentemente diminui a vazão.

Muitas técnicas estão sendo usadas para minimizar os efeitos das perdas de pacotes. Isto inclui a técnica FEC e retransmissões no emissor, técnicas de correção de erros no receptor/decodificador. Mas é inevitável, quanto mais sofisticada forem as técnicas, maiores serão o consumo de banda para manter a qualidade desejada. A Figura 13 faz uma relação entre a qualidade do vídeo e a taxa de codificação em condições de perdas (MPEG), como pode ser visualizada a qualidade aumenta em relação à taxa, mas depois cai com o efeito de perda.

Figura 13 – Qualidade do vídeo x taxa de codificação
(ITU-T, 1998)

O gráfico é baseado na indicação e recomendação do grupo de estudos 12 da ITU-T (ITU-T, 2001) e a literatura afirma que a qualidade para aplicações genéricas suportam tolerância abaixo de 1% para *streaming* de áudio e vídeo de alta qualidade e abaixo de 2-3% para serviços áudio visuais interativos, o gráfico mostra que a tolerância dos fluxos de áudio e vídeo pode ser aumentada pelo uso de FEC (Miras, 2002) (Beuran, 2003).

A Figura 14 ilustra os limiares aceitáveis de qualidade entre áudio e vídeo. A área marcada mostra onde a qualidade é aceitável e as linhas determinam os limites para a que irregularidade entre áudio e vídeo é detectável.

Figura 14 – Limiares aceitáveis de qualidade entre áudio e vídeo
Adaptado de (Miras, 2002]

Quando o atraso do áudio for menor do que o atraso do vídeo, o limiar de detecção está em torno de 120ms. Quando o atraso do áudio for maior que o atraso do vídeo, então o limiar para detecção ocorre em torno de 250ms (Miras, 2002). As aplicações podem usar diversas técnicas de adaptação para lidar com as condições de rede:

1. **Adaptação de vazão** – fluxos multimídia são capazes de se adaptarem de acordo com a banda disponível variando suas taxas de transmissão. Isto é possível devido à habilidade inerente dos *streamings* multimídia operarem com qualidade aceitável dentro de uma faixa de taxas oferecidas, explorando as redundâncias dentro do sinal. Não importa a circunstância da rede (melhor esforço ou redes com suporte a QoS), cada aplicação deve efetuar a adaptação de vazão e taxa.
2. **Adaptação de atraso e variação do atraso** – Aplicação pode se adaptar em uma variedade de atraso, enquanto estiver dentro de um intervalo razoável. Existem níveis de atraso que são aceitáveis pelas aplicações sem comprometer a qualidade. Quando estes níveis são excedidos, as aplicações podem desenvolver mecanismos (*buffers*) que permitem realizar a adaptação de vários atrasos.
3. **Adaptação para perda de pacotes** – Aplicação pode empregar diversas técnicas para melhorar a resistência dos fluxos para perda de pacotes. Cada técnica pode ser proativa ou direta, como uma codificação robusta e usa proteção de erro, ou reativa ou indireta fazendo a redução da taxa de transmissão para minimizar o congestionamento.

4.7 – Considerações Finais

Desde que as aplicações avançadas começaram a serem desenvolvidas em larga escala, os requisitos de QoS tem se tornado um mecanismo muito utilizado visando garantir a qualidade exigida por cada tipo de aplicação. Aplicar estas técnicas em rede IP implica em investimentos elevados, testes e planejamentos. O desenvolvimento de cada serviço requer mudanças drásticas na rede com custo considerável, tanto operacional como financeiro. Estes serviços requererem mudanças no suporte aos roteadores, pois os mesmos não dão suporte as complexidades elevadas exigidas pela abordagem QoS. De forma tradicional a solicitação de QoS para cada perfil de aplicação é denominada tipicamente de SLA (*Service Level Agreement*). O SLA deve definir claramente quais requisitos devem ser garantidos para que uma aplicação possa ser executada com qualidade. É importante ressaltar que geralmente esses contratos

apresentam informações estáticas para representar os requisitos de QoS. Devido a todas as limitações existentes, fica evidente a necessidade de novas soluções gerenciamento dinâmicos de rede para que os recursos e aplicações sejam usados de forma mais eficiente.

Capítulo 5

Categorias de QoS das Aplicações multimídia

De acordo com as normas e recomendações da ITU-T (2001) o ponto chave é prover informações com fatores que influenciam o QoS em aplicações multimídia. Partindo da perspectiva dos usuários finais, dentre as aplicações envolvidas estão: mídias de voz, vídeo, imagens e texto, e os parâmetros que administram a satisfação dos usuários para estas aplicações. Visando classificar as aplicações multimídia em grupos de acordo com suas características e perfil, esta seção faz uma breve descrição dos tipos de aplicações e as classifica em grupos baseados nos requisitos de QoS exigidos para que possa operar com qualidade.

5.1 – Requisitos de desempenho

O maior desafio para as redes emergentes é prover QoS adequado para diferentes serviços e aplicações existentes nos dias de hoje. Para resolver este problema, é necessário conhecer de forma detalhada os requisitos de desempenho para cada tipo particular de serviço e aplicação. De acordo com a ITU-T (2002) e ITU-T (2001), o ponto inicial para derivar os requisitos de desempenho deve ser sempre o usuário (QoE) (Harman, 1990).

Um usuário típico não conhece como um serviço em particular foi desenvolvido. Entretanto, ele está interessado em comparar o mesmo serviço oferecido por diferentes provedores de forma universal, onde os requisitos de QoS acabam sendo caracterizado de acordo com a experiência do usuário. Isto implica que o desempenho deveria ser expresso através dos seguintes pontos:

- Levar em conta todos os aspectos do serviço partindo do ponto de vista do usuário;
- Ter foco nos efeitos perceptíveis do usuário, ao invés das causas no âmbito de rede;
- Ser independente de uma arquitetura ou tecnologia específica de rede;
- Ter medidas objetivas ou subjetivas no ponto de acesso do serviço;

- Ser facilmente relacionados com os parâmetros de desempenho da rede;
- Garantir o serviço pelos provedores ou entidades prestadoras de serviços;

5.2 – Considerações de desempenho para diferentes aplicações

5.2.1 – Áudio

5.2.1.1 – Conversação por voz

Os requisitos exigidos pelas aplicações que suportam conversação por voz são fortemente influenciados por atraso. De forma geral, existem dois efeitos distintos de atraso; o primeiro é a criação de eco pelos fios ou mesmo pelo acoplamento acústico em um terminal e o segundo efeito ocorre quando o atraso aumenta em um ponto onde ele começa a impactar a dinâmica da conversação, o atraso no outro lado da conversação acaba sendo notado.

Entretanto, o ouvido humano é altamente intolerante à variação de atraso. Como questão prática, para todos os serviços de voz, a variação de atraso ocorre devido à variabilidade no tempo de chegada dos pacotes, este problema deve ser removido com as técnicas de buferização (*de-jitterizing buffers*) conforme já descrito anteriormente.

Os requisitos de perdas são influenciados pelo fato de que o ouvido humano é tolerante a certa quantia de distorção no sinal de voz. Nos sistemas de transmissão baseados em IP, o principal causador da degradação de voz ocorre devido ao uso de *codecs* de compressão de voz com baixo *bit-rate* e seu desempenho sobre as condições de perdas de pacotes. A Figura 15 ilustra os requisitos mínimos exigidos pela aplicação de Voz Convencional de acordo com a definição de ITU-T (2002).

Perfil VozConvencional**vazão maior ou igual a 64 kbit/s** (kilo bit por segundo)**atraso menor que 400 ms** (milissegundos)**jitter menor que 20 ms** (milissegundos)**perdas menor que 3%** (percentual)**Figura 15 – Perfil Voz Convencional****5.2.1.2 – Mensagem de voz**

Os requisitos para perdas de informação são essencialmente os mesmos usados pela conversação de voz (depende do *encoder* de voz), mas a principal diferença é que ele é mais tolerante ao atraso, desde que não esteja diretamente envolvida em uma conversação. A questão principal, entretanto é saber o quanto de atraso pode ser tolerável entre o usuário e o sistema através dos comandos de reprodução (*replay*) da mensagem de voz e o início do áudio. Não existem dados precisos sobre isso, mas baseado em estudos relatados sobre a aceitabilidade de atrasos estímulos-respostas para serviços de telecomunicação, um atraso da ordem de alguns segundos parece aceitável para estas aplicações (Miras, 2002). De fato, é possível fazer uma distinção entre a gravação e reprodução, em que a reação do usuário no que tange a reprodução é provável que seja a mais rigorosa exigência. A Figura 16 descreve os requisitos mínimos exigidos pela aplicação Mensagem de Voz.

Perfil MensagemVoz**vazão maior ou igual a 32 kbit/s** (kilo bit por segundo)**atraso menor que 1 ms** (milissegundos)**jitter menor que 1 ms** (milissegundos)**perdas menor que 3%** (percentual)**Figura 16 – Perfil Mensagem de voz****5.2.1.3 – Streaming de áudio**

Existe uma expectativa para que o *streaming* de áudio possa prover melhor qualidade que o telefone convencional, e em termos de perda de pacotes serão correspondentemente mais restritos. Entretanto, como na mensagem de voz, não existem elementos de conversação envolvidos e os requisitos de atraso para o

streaming de áudio pode ser mais flexível, apesar dos controlos, devem ser tratados adequadamente. A Figura 17 ilustra os requisitos mínimos exigidos pela aplicação *Streaming* de áudio de alta qualidade.

Figura 17 – Perfil *Streaming* de Áudio de Alta Qualidade

5.2.2 – Vídeo

5.2.2.1 – Videophone

Uma aplicação de *Videophone* pode ser caracterizada como um sistema *full-duplex*⁷ que pode transmitir vídeo, áudio e ainda ser usado em um ambiente de conversação natural. Em princípio, os mesmos requisitos de atraso usados para conversação de voz, serão aplicados no sistema *Videophone*. Nenhum eco e o mínimo efeito na dinâmica da conversação, juntamente com a adição do áudio e vídeo deve ser sincronizada, dentro de certos limites para proporcionar o sincronismo-labial (*lip-sync*). Como olhos humanos são tolerantes a pequenas perdas de informação, algumas perdas de pacotes podem ser aceitáveis, principalmente quanto o sistema dispõe de mecanismos de proteção de erro. É esperado que o *codec* de vídeo utilizado, como o MPEG-4, por exemplo, possa prover uma qualidade aceitável se a taxa de perdas não ultrapassar 1%. A Figura 18 descreve os requisitos mínimos exigidos pela aplicação *VideoPhone*.

Figura 18 – Perfil *VideoPhone*

⁷ *Full-duplex* – sistema de comunicação composto por dois interlocutores que podem comunicar entre si em ambas direcções.

5.2.2.2 Vídeo *One-way*

A principal característica de distinção de um vídeo *one-way* é que não há elementos de conversação envolvidos, significando que os requisitos de atraso não necessitam ser tão exigentes, e podem seguir a mesma configuração usada pelo streaming de vídeo. Baseado nas informações apresentadas anteriormente, a Tabela 11 (ITU-T, 2001) prove uma indicação geral e adequada aos objetivos de desempenho para aplicações de áudio e vídeo.

Tabela 11 – Requisitos de desempenho para aplicações de áudio e vídeo

Mídia	Aplicação	Simetria	Vazão	Atraso	Jitter	Perdas
Audio	Voz convencional	<i>Two-way</i>	4-64 kbit/s	<150 (preferencial) <400 (limite) *	<1 ms	<3% PLR
Audio	Mensagem de voz	<i>One-way</i>	4-32 kbit/s	<1s (playback)	<1 ms	<3% PLR
Audio	<i>Streaming</i> de áudio (<i>High Quality</i>)	<i>One-way</i>	16-128 kbit/s **	<10s	<1 ms	<1% PLR
Video	<i>Videophone</i>	<i>Two-way</i>	16-384 kbit/s	<150 ms (preferencial) <400 (limite) ***	-	<1% PLR (<80ms - Lip-sync)
Video	<i>One-way</i>	<i>One-way</i>	16-384 kbit/s	<10 s	-	<1% PLR
* Valor adequado assumido						
** Qualidade é dependente do tipo de codec e bit-rate						
*** Valores considerados a longo prazo e não podem ser alcançados com a tecnologia corrente						

5.2.3 – Dados

Para as aplicações de dados, os requisitos de atraso e variação de atraso acabam sendo irrelevantes, pois os fatores que impactam efetivamente o desempenho deste tipo de aplicação são vazão, taxa de erros ou perdas. Do ponto de vista do usuário, o principal requisito para uma aplicação de transferência de dados é garantir que o sistema não tenha perdas de informação, e ao mesmo tempo, a variação de atraso não ser notável pelo usuário, embora deva haver um limite no processo de sincronização entre o *stream* da mídia em uma sessão multimídia (ex. áudio em conjunto com uma apresentação em *white-board*). Entretanto, as diferentes aplicações tendem a distinguir as bases de atraso que podem ser toleráveis pelo usuário final, partindo do tempo em que o usuário requisitou um conteúdo, até sua apresentação.

5.2.3.1 – Web-browsing

De acordo com a recomendação da ITU-T (2001), esta categoria de aplicação refere-se à recuperação e visualização de componentes HTML de uma página Web, outros componentes como imagens, áudio/vídeo *clips* foram

separados em outra categoria. Do ponto de vista do usuário, o fator de desempenho principal é o quão rápido uma página carrega depois que ela foi requisitada. Atrasos de diversos segundos são aceitáveis, mas não deve ultrapassar os 10 segundos de acordo com a norma definida por ITU-T (2001). A Figura 19 ilustra os requisitos mínimos exigidos pela aplicação *Web-Browsing*.

Perfil WebBrowsing
vazão *menor que 10 kB* (kilo byte por segundo)
atraso *menor que 4s* (segundos)
jitter *não especificado*
perdas *menor ou igual a 0%* (percentual)

Figura 19 – Perfil *Web-Browsing*

5.2.3.2 – Volume de dados (*Bulk data*)

Esta categoria inclui a transferência de arquivos (em rajadas), e é claramente influenciada pelo tamanho do arquivo. Enquanto não houver uma indicação de que a transferência do arquivo foi processada, é razoável prever um tempo mais longo de atraso do que a tolerância requerida por uma simples página Web. A Figura 20 ilustra os requisitos mínimos exigidos pela aplicação *Bulk Data Transfer*.

Perfil BulkDataTransfer
vazão *menor que 10 MB* (Mega byte por segundo)
atraso *menor que 60s* (segundos)
jitter *não especificado*
perdas *menor ou igual a 0%* (percentual)

Figura 20 – Perfil *Bulk DataTransfer*

5.2.3.3 – Serviços de transação de alta prioridade (*e-commerce*)

O principal requisito de desempenho para este tipo de serviço é prover uma sensação de rapidez para o usuário e que a transação esteja processando sem problemas, um atraso de não mais que alguns segundos é desejável. A Figura 21 ilustra os requisitos mínimos exigidos pela aplicação de alta prioridade (*e-commerce*).

Perfil TransactionHighPriority
vazão *menor que 10 kB* (kilo byte por segundo)
atraso *menor que 4s* (segundos)
jitter *não especificado*
perdas *menor ou igual a 0%* (percentual)

Figura 21 – Perfil Aplicação de alta prioridade

5.2.3.4 – Comandos/controles

As funções acionadas através de controles ou comandos, de forma clara, implicam em limites muito restritos dentro de um atraso tolerável, muito menor que um segundo. Vale ressaltar que o principal diferencial de uma conversação por voz e serviços de vídeo com similaridades no que tange o baixo atraso é tolerância zero para perda de informação. A Figura 22 ilustra os requisitos mínimos exigidos pela aplicação *CommandControl*.

Perfil CommandControl
vazão *menor ou igual a 1 kB* (kilo byte por segundo)
atraso *menor que 250 ms* (Milissegundos)
jitter *não especificado*
perdas *menor ou igual a 0%* (percentual)

Figura 22 – Perfil CommandControl

5.2.3.5 – Imagens estáticas

Esta categoria inclui uma variedade de formatos de codificação, alguns deles pode ser tolerante as perdas uma vez que serão vistas por um olho humano. Entretanto, dado um simples erro em um *bit*, pode causar grandes distúrbios em outros formatos de imagens estáticas, e de forma geral, este tipo de categoria não tolera perdas de informação. Os requisitos de atraso para a transferência de imagens estáticas não são exigentes e pode ser comparável com a transferência de volume de dados (*bulk data*), dado que a imagem tende a ser construída no momento de seu recebimento, fornecendo uma indicação de que a transferência dos dados está em progresso. A Figura 23 ilustra os requisitos mínimos exigidos por esta categoria de aplicação.

Perfil *ImagensEstaticas***vazão** *menor que 100 kB* (kilo byte por segundo)**atraso** *menor que 60 s* (segundos)**jitter** *não especificado***perdas** *menor ou igual a 0%* (percentual)**Figura 23 – Perfil *Imagens estáticas*****5.2.3.6 – Jogos interativos**

Os requisitos para jogos interativos são obviamente dependentes de cada jogo em particular, onde as aplicações requerem por si só baixos atrasos, da ordem de uma fração de segundos, coerentes com as exigentes das aplicações interativas. A Figura 24 ilustra os requisitos mínimos exigidos pelas aplicações de Jogos Interativos.

Perfil *JogosInterativos***vazão** *menor que 100 kB* (kilo byte por segundo)**atraso** *menor que 60 s* (segundos)**jitter** *não especificado***perdas** *menor ou igual a 0%* (percentual)**Figura 24 – Perfil *Jogos Interativos*****5.2.3.7 – Telnet**

A aplicação *Telnet* é descrita aqui como uma aplicação que demanda por requisitos com baixo atraso, na ordem de fração de segundos, o que permite prover acesso instantâneo aos dados desejados, conforme ilustrado pela Figura 25.

Perfil *Telnet***vazão** *menor ou igual a 1 kB* (kilo byte por segundo)**atraso** *menor que 200 ms* (milissegundos)**jitter** *não especificado***perdas** *menor ou igual a 0%* (percentual)**Figura 25 – Perfil *Telnet***

5.2.3.8 – E-mail

Aplicação de email é geralmente considerada um serviço de armazenamento e reencaminhamento de dados, o que em princípio, pode tolerar atrasos de alguns minutos ou até mesmo horas. Entretanto, é importante diferenciar comunicação entre usuários e entre servidores de email local e servidores para servidores de transferências de dados. Quando um usuário se comunica com o servidor de email local, existe uma expectativa de que o email será transferido dentro de alguns segundos. A Figura 26 ilustra os requisitos mínimos exigidos pela aplicação de email.

Perfil EmailServerAccess

vazão *menor que 10 kB* (kilo byte por segundo)

atraso *menor que 4 ms* (milissegundos)

jitter *não especificadas*

perdas *menor ou igual a 0%* (percentual)

Figura 26 – Perfil EmailServerAccess

5.2.3.9 – Fax

Fax pode ser incluso nesta categoria desde que não tenha a pretensão de ser considerada uma comunicação altamente interativa e em tempo real. Para ser chamada de fax em tempo real, existe uma expectativa na maioria dos cenários comerciais que o fax deve ser recebido dentro de 30 segundos. Como o atraso para armazenamento e reencaminhamento do fax geralmente ocorre em um tempo mais alto, este tipo de aplicação requer zero ou nenhuma perda de informações. A Figura 27 ilustra os requisitos mínimos exigidos pela aplicação de Fax.

Perfil FaxStoreForward

vazão *menor ou igual a 10 kB* (kilo byte por segundo)

atraso *pode ser diversos minutos*

jitter *não especificado*

perdas *menor 0%* (percentual)

Figura 27 – Perfil FaxStoreForward

5.2.3.10 – Transações de baixa prioridade

Este tipo de aplicação é categorizado pela ITU-T (ITU-T, 2001) como exemplo, por um serviço de mensagens curtas (SMS – *Simple Message System*). O tempo de dez segundos (10s) é considerado um valor aceitável para o atraso de entrega. A Figura 28 ilustra os requisitos mínimos exigidos pelas aplicações de baixa prioridade.

Perfil TransactionLowPriority
vazão *menor que 10 kB* (kilo byte por segundo)
atraso *menor que 30s* (segundos)
jitter *não especificado*
perdas *menor 0%* (percentual)

Figura 28 – Perfil Transação de baixa Prioridade

5.2.3.11 – E-mail (Server-to-server)

Esta categoria de aplicação foi inclusa neste trabalho apenas para integralidade das aplicações abordadas. Conforme já mencionado, o principal interesse no email é no seu tempo de acesso. A Figura 29 ilustra os requisitos mínimos exigidos pela aplicação de email (*Server-to-Server*).

Perfil EmailServerToServerTransfer
vazão *menor que 10 kB* (kilo byte por segundo)
atraso *aceitável em alguns minutos*
jitter *não especificado*
perdas *menor ou igual a 0%* (percentual)

Figura 29 – Perfil EmailServerToServerTransfer

5.2.3.12 – Usenet

Uma aplicação *Usenet* pode ser classificada como um sistema de discussão distribuída pela rede (*world-wide*), que consiste em um conjunto de novos grupos classificados hierarquicamente por assuntos. Artigos e mensagens são postados por estes grupos em computadores que utilizam softwares apropriados. Estes artigos são então enviados através de *broadcast* para outros computadores interconectados através de diferentes redes. Este é um serviço de

baixa prioridade, o que representa requisitos de atraso mais flexíveis. Entretanto, é desejado que as mensagens sejam recebidas pelos usuários na ordem em que elas foram postadas, evitando assim a exibição de uma resposta antes mesmo da mensagem original chegar. A Figura 30 ilustra os requisitos mínimos exigidos pela aplicação *Usenet*.

Perfil Usenet
vazão maior que 1 MB (Mega byte por segundo)
atraso pode ser diversos minutos
jitter não especificado
perdas menor 0% (percentual)

Figura 30 – Perfil Usenet

Baseado em todas as informações descritas para cada tipo de aplicação, a Tabela 12 (ITU-T, 2001) fornece uma indicação dos objetivos de desempenho para aplicações de dados.

Tabela 12 – Objetivos de desempenho das aplicações de dados

Mídia	Aplicação	Simetria	Quantidade de dados	Atraso	Jitter	Perdas
Dados	Web-browsing - HTML	One-way	~10 KB	<2 s (preferencial) / <4 (aceitável)	-	zero
Dados	Bulk data transfer/retrieval	One-way	10 KB - 10MB	<15 s (preferencial) / <60 s (aceitável)	-	zero
Dados	Transações de alta prioridade	Two-way	<10 KB	<2 s (preferencial) / <4 (aceitável)	-	zero
Dados	Comando/Controle	Two-way	~1 KB	< 250 ms	-	zero
Dados	Imagens estáticas	One-way	< 100 KB	<15 s (preferencial) / <60 (aceitável)	-	zero
Dados	Jogos interativos	Two-way	< 1 KB	< 200 ms	-	zero
Dados	<i>Telnet</i>	Two-way	< 1 KB	< 200 ms	-	zero
Dados	<i>E-mail (server access)</i>	One-way	< 10 KB	<2 s (preferencial) / <4 (aceitável)	-	zero
Dados	<i>E-mail (server to server transfer)</i>	One-way	< 10 KB	Aceitavel em alguns minutos	-	zero
Dados	Fax - tempo real	One-way	~10 KB	<30s/página	-	<10-6 BER
Dados	Fax - <i>store and forward</i>	One-way	~10 KB	Pode ser diversos minutos	-	<10-6 BER
Dados	Transações de baixa prioridade	One-way	< 10 KB	< 30 s	-	zero
Dados	Usenet	One-way	1 MB ou mais	Pode ser diversos minutos	-	zero

5.3 – Classificações dos requisitos de desempenho do usuário em categorias de Qualidade de Serviço

Baseados nos requisitos de desempenho apresentados anteriormente, seguindo a recomendação da ITU-T (2008), várias aplicações podem ser mapeadas entre os eixos de perdas de pacotes e atraso. O tamanho e formato das caixas representando as aplicações provêm uma indicação geral do limite de atraso e perdas toleráveis para cada classe de aplicação, conforme ilustra a Figura 31.

Figura 31 – Mapeamento dos requisitos de QoS centrados no usuário
Adaptado de (ITU-T-2008)

A ITU-T (2008) separa as aplicações em 8 grupos distintos que abrangem as aplicações identificadas conforme ilustrado pela Figura 31. Dentro destes grupos, existe uma separação entre as aplicações que podem tolerar alguma perda de informação e aquelas que não toleram nenhuma perda, e quatro áreas gerais tolerantes ao atraso.

Este mapeamento pode ser formalizado através da Figura 32, por fornecer um modelo recomendado para categorizar os requisitos de QoS dos usuários finais, onde essas quatro áreas com relação ao atraso são dadas por nomes escolhidos para ilustrar o tipo de interação envolvido. Ainda é possível subdividir estas categorias em subcategorias, provendo um serviço específico através de níveis de qualidade, e tem sido feito para conversação por voz pelo ETSI TS 101329-2 (ETSI-TS, 2002).

Figura 32 – Modelo de categorias de QoS centrado no usuário
Adaptado de (ETSI-TS, 2002)

Partindo deste princípio, as principais características do modelo seguindo a recomendação por ITU-T (2001) e ITU-T (2002) são:

1. É baseado na percepção dos usuários finais e não é dependente de nenhuma tecnologia para sua validação (independente da tecnologia utilizada (IP, ATM, *wireline*, *wireless*, dentre outros)).
2. Fornece uma indicação dos limites (superiores e inferiores) para que uma aplicação possa ser considerada aceitável ou não pelos usuários.
3. Determina se um canal é capacitado para transportar um fluxo de dado de uma determinada aplicação.
4. Mostra como as dificuldades adjacentes relacionadas a perdas de informações e atrasos podem ser agrupados de forma apropriada, sem implicar se uma classe é melhor que a outra (categorização *Gold*, *Silver*, *etc*).
5. As aplicações apresentadas são exemplares e não foram tratadas de formas exaustivas. Outras aplicações podem ser alocadas dentro deste esquema, através de sua similaridade com os exemplos apresentados.

Capítulo 6

Arquitetura do Sistema de Gerenciamento - PBQoS

Este capítulo descreve a arquitetura do sistema de gerenciamento PBQoS composto por um modelo de gestão por políticas que representam as regras que serão aplicadas na rede de distribuição. O sistema tem como base as informações contextuais coletadas dinamicamente da rede e os requisitos mínimos exigidos por cada perfil de aplicação. Com base nestas informações o sistema busca pelos melhores recursos que serão utilizados na composição da melhor rota para atender aos requisitos mínimos exigidos por uma aplicação. Foi escolhido um cenário de rede sobrepostas com intuito de migrar parte da complexidade existente da rede IP para a camada de aplicação, além do desenvolvimento de serviços que não são suportados ou difíceis de serem implementados na rede física. Dessa forma, não há a necessidade de modificar a estrutura existente nas redes atuais, nos protocolos existentes, além de fornecer um sistema escalável. O sistema de gerenciamento baseado em políticas permitirá definir os requisitos específicos de QoS para cada perfil de aplicação, garantindo assim os melhores recursos para a efetiva distribuição de fluxos multimídia pela rede.

6.1 – Arquitetura

A arquitetura proposta para o sistema de gerenciamento baseia-se em uma estrutura hierárquica e distribuída, cujo principal objetivo é determinar a melhor rota entre um nó de origem e destino tendo como base os requisitos de QoS (vazão, atraso, variação do atraso e perdas). Esta arquitetura visa melhorar a distribuição dos recursos de comunicação existentes na rede gerenciada através de mecanismos que permitam determinar a melhor rota disponível. Desta forma, é possível tratar cada serviço individualmente em busca da melhor configuração da rede para atender as necessidades de uma aplicação.

Como a maioria das arquiteturas de rede é organizada em camadas, nos quais as camadas inferiores são responsáveis por prover serviços para as

camadas superiores, mesmo sendo completamente independentes (Verma, 2000), pretende-se neste momento utilizar o modelo definido pela arquitetura TCP/IP como base no intuito de ilustrar onde a arquitetura PBQoS se encontra inserida. Conforme ilustra a Figura 33, a rede nativa é aqui representada pelas da arquitetura TCP/IP (Rede, Internet (IP), Transporte (TCP/UDP) e aplicação), enquanto que as camadas de serviços *overlay* (*Serviços Básicos Overlay* e *Serviços de rede Overlay*) descrevem a arquitetura PBQoS e seus elementos.

Figura 33 – Adaptação da Arquitetura TCP/IP para o sistema PBQoS

- **Interface com a rede** – representa o menor nível de abstração da arquitetura TCP/IP. Define as características técnicas dos dispositivos e equipamentos do sistema. Esta camada provê acesso aos equipamentos físicos responsáveis por conectarem as máquinas, como as LAN's ou enlaces de acesso. Para isto, utiliza padrões de conexão à rede física, seja ela local (*Ethernet-CSMA/CD*, *Token Ring*, *FDDI* e *ATM*) ou de longa distância (*HDLC*, *X. 25* *ATM*) (Verma, 2000).
- **Internet** – A camada Internet (IP – Internet Protocol) provê entrega de pacotes sem conexão através do modelo melhor esforço. Esta camada estabelece os protocolos para transporte não confiáveis de mensagens IP; para controle de comunicação e informe de erros (*ICMP* – *Internet*

Control Message Protocol) e para roteamento de mensagens (EGP – *Exterior Gateway Protocol* e RIP – *Routing Information Protocol*).

- **Transporte** – A camada Transporte inclui os protocolos TCP (*Transfer Control Protocol*) e UDP (*User Datagram Protocol*). Provê uma conexão confiável fim-a-fim e entrega de mensagens não confiáveis, respectivamente (Verma, 2000). Sua principal tarefa é prover o transporte econômico e confiável de dados, seja por TCP ou UDP, independente da rede física ou das redes atualmente em uso e realiza o controle de fluxo, ordenação de pacotes e correção de erros.
- **Serviços Básicos Overlay (BOS – *Basic Overlay Service*)** – Esta camada é composta pelos serviços básicos de comum acesso para todos os serviços definidos na camada superior (SON – *Service Overlay Network*). Pode ser considerada uma extensão da camada de rede e transporte, no qual é responsável por prover suporte básico aos elementos de cada serviço definido pelas camadas superiores.
- **Rede de Serviços Overlay (SON)** – Forma efetiva de prover serviços de rede avançada que não podem ser aplicados diretamente na camada física *best-effort*. Representam alguns dos serviços que poderiam ser suportados pela arquitetura PBQoS. Destre estes serviços destacam-se: vídeo ao vivo (*Live*), vídeo sob-demanda (VoD), VoIP, QoS, segurança, etc.
- **Aplicação** – a camada de aplicação apresenta o maior nível de abstração definido pela arquitetura TCP/IP (Verma, 2000). Dessa forma, é possível definir um conjunto de serviços manipulados pelos usuários, por exemplo, acesso aos serviços providos pela camada SON, aplicações provedoras e consumidores de conteúdo, dentre outros serviços que utilizam a arquitetura cliente-servidor.

Após a contextualização da proposta através da arquitetura TCP/IP, a Figura 34 ilustra de forma detalhada alguns dos serviços suportados pela arquitetura PBQoS (camada SON) e os elementos básicos providos pela camada BOS.

Figura 34 – Detalhamento das camadas BOS e SON

- **Serviço de Monitoração de QoS** - Provê monitoração dos elementos de rede, para manter o repositório de métricas sempre atualizado com o estado atual da rede física e lógica através dos requisitos de QoS.
- **Serviço de Distribuição** - Serviço de entrega de conteúdo desenvolvido nos nós de serviços, por exemplo, servidores utilizados para a distribuição de conteúdo multimídia (Almeida, 2006). Um sistema de distribuição eficiente visa reduzir o tempo de resposta das requisições dos usuários através de provedores de conteúdo intermediários (*Caching*) próximos desses usuários. Consiste em uma rede de servidores para onde os usuários serão direcionados de forma transparente através de diversos esquemas de balanceamento de carga (Verma, 2002). No âmbito das tecnologias e serviços de distribuição de vídeo, o sistema tem foco em aplicações Web e normalmente a distribuição de conteúdo pela rede se dá de duas formas: serviço de distribuição sob-demanda e serviço de distribuição ao vivo (Lacerda, 2007).
- **Aplicação de gestão** – Trata-se de uma aplicação que permite ao administrador da rede manipular políticas de configuração, manutenção e controle e perfis das aplicações. Esta aplicação pode controlar os elementos de rede presente na camada BOS, bem como manipular as configurações dos serviços disponíveis na camada SON.
- **Serviços multimídia** – São as aplicações de rede que permitem o armazenamento e manipulação de dados e informação através de uma

variedade de formas como áudio, vídeo, texto, gráficos e animação. Dentre estas aplicações, destacam-se:

- VoD (*Vídeo on Demand*) – são as aplicações que possuem conteúdos pré-gravados e armazenados em servidores.
 - Live - são as aplicações que dispõe de mídia contínua transmitida ao vivo.
 - VConf (*Vídeo Conference*) – trata-se de uma aplicação multimídia com mídia contínua interativa (conferência de áudio e/ou vídeo).
 - VoIP (*Voice over IP*) – também conhecida como telefonia IP. É uma aplicação que permite a conversação humana através da Internet ou qualquer outra rede de computadores baseada no Protocolo IP.
 - Segurança – são as aplicações relacionadas com a proteção de um conjunto de dados, no sentido de preservar o valor que possuem para um indivíduo ou uma organização.
- **PBQoS** (*Policy Based Quality of Service*) – Trata-se da arquitetura de gerenciamento baseada em políticas, no qual faz uso de requisitos de QoS, para administrar os serviços e recursos da rede sobreposta. Esta arquitetura dispõe de diversos componentes que operam em conjunto para definir a melhor rota capaz de atender os requisitos mínimos exigidos por uma determinada aplicação. Por se tratar da principal contribuição deste trabalho, a arquitetura PBQoS será detalhada na seção 6.1.2.

6.1.2 – Arquitetura PBQoS

A arquitetura PBQoS (*Policy Based Quality of Service Overlay Network*), define os componentes necessários para representar um sistema de gerenciamento baseado em políticas aplicado em redes sobrepostas. Tem o intuito de realizar um controle dos requisitos de QoS e elementos da rede para melhor distribuição de conteúdo pela rede.

O sistema é responsável pela definição das atividades de gerenciamento e administração dos elementos do sistema como um todo, coordenando a definição de rotas e alocação de recursos entre provedores e consumidores de conteúdo.

Para realizar esta operação, a arquitetura PBQoS permite coletar informações operacionais das diversas entidades da rede gerenciada, levando em consideração, proximidade do usuário, requisitos de QoS, operacionalidade e disponibilidade dos recursos.

A adoção do conceito de proximidade dos usuários permite uma melhor utilização dos recursos da rede, pois, mesmo dentro de um ambiente impreciso, é possível determinar os elementos que rede que serão utilizados para distribuir o tráfego gerado. Esta arquitetura segue o modelo de gerenciamento proposto pela IETF (2008) e adota um modelo de gerenciamento dos requisitos de QoS aplicados em um sistema de distribuição de conteúdo multimídia. A arquitetura procura minimizar os impactos causados no desempenho das aplicações, ocasionados pelas mudanças na carga dos requisitos de QoS da rede.

Nos dias de hoje, ter um sistema de gerenciamento eficiente é crucial nas redes de distribuição, pois automatiza tarefas normalmente executadas manualmente pelos administradores de rede. Pode garantir prioridades de tráfego, entrega de serviços em determinados níveis de recebimento, segurança (recusando tráfego de origem desconhecida, não permitindo que o mesmo entre na rede de acesso), controle de aplicações, dentre outras. A Figura 35 ilustra a arquitetura geral do sistema de gerenciamento PBQoS.

Figura 35 – Elementos arquiteturais – PBQoS

6.1.2.1 – Repositório TED

Um sistema de Engenharia de Tráfego pode ter a topologia de uma rede representada em uma base de dados, no qual os dados são obtidos pela troca de informação de roteamento entre os nós. Quando um nó detecta que a conexão com um vizinho falhou, ele informa aos outros nós através do seu domínio de roteamento. No momento em que um nó volta a estar operacional, ocorre a comunicação com todos os nós pertencentes ao seu domínio de roteamento. Geralmente este processo é realizado através do mecanismo de *flooding* gerado pelo protocolo OSPF (*Open Shortest Path First*), por exemplo. Maiores detalhes do protocolo OSPF serão detalhados posteriormente.

Uma vez que a topologia da rede e o estado atual dos enlaces são conhecidos, o sistema pode calcular uma rota sob-demanda, atingindo assim os objetivos da Engenharia de Tráfego (gerenciamento dos elementos da rede,

fechamento da malha gerenciada, monitoração das interfaces, dentre outros). Com base nessas informações, o repositório TED (*Traffic Engineering Database*) representa a base de dados que contém informações referentes à topologia de rede, estado de cada enlace (QoS medido), demanda de tráfego e rotas possíveis. As informações contidas neste repositório devem estar sempre atualizadas, pois servirá de base para realização do cálculo de uma rota capaz de atender as necessidades exigidas pelas aplicações. O repositório TED é composto pela MIB (*Management Information Base*) geral do sistema que contém todas as informações de QoS das MIBs localizadas nos PEP's (*Policy Enforcement Point*).

Visando ilustrar uma representação da MIB utilizada pelo sistema PBQoS, foi assumido três nós (A, B e C), nós quais armazenam informações dos enlaces que os conectam. A Figura 36 ilustra um exemplo abstrato da representação de uma MIB no repositório TED.

Figura 36 – Representação da MIB do sistema

6.1.2.2 – Repositório Políticas

O repositório de políticas é composto pelas políticas que definem as regras impostas ao sistema de gerenciamento PBQoS. Essas regras determinam o comportamento do sistema de gerenciamento e é basicamente composto por três tipos distintos de políticas: políticas de controle, políticas de configuração e políticas de manutenção conforme ilustra a Figura 37.

Figura 37 – Representação abstrata do Repositório de Políticas

- Políticas de Controle** – Definem as regras que determinam a administração do sistema. É responsável pelo controle de admissão de novos serviços, envio de alertas de desempenho ao administrador da rede e mecanismos de atualização de MIB. Visando manter o sistema operacional, sem que seus recursos sejam completamente esgotados, as políticas de controle definem regras que impedem que um serviço seja admitido quando a taxa de ocupação do enlace (T) não exceder um valor pré-definido. Na implementação da política, é possível definir posições intermediárias de alerta para o administrador. No contexto deste trabalho, foram utilizados dois mecanismos de alerta de desempenho: um "alerta amarelo" quando T atingir uma porcentagem α do seu valor máximo, e um "alerta vermelho" quando T atingir uma porcentagem β de seu valor máximo.

Exemplo:

Política de Controle (Alertas de desempenho)

=====

Descrição: Define mecanismos que emitem alertas ("amarelo" e "vermelho") quando os limites impostos pelo sistema são atingidos.

=====

Sendo T a Taxa de ocupação do enlace

α o fator limite da rota para emissão do "alerta amarelo" ao administrador da rede

β o fator limite da rota para emissão de um "alerta vermelho" ao administrador da rede

Condição 1: SE ($T \geq \alpha$) && ($T \leq \beta$)

Ação 1.1: Sistema envia um "alerta amarelo" ao administrador da rede

Condição 2: SE ($T \geq \beta$)

Ação 2.1: Sistema envia um "alerta vermelho" ao administrador da rede

Política de Controle (Admissão de serviços)

=====

Descrição: Define as regras que determinam a administração do sistema realizando o controle de admissão de novos serviços.

=====

Sendo a **vazãoRota** a vazão total de uma rota fim-a-fim e a **vazãoSLA** o valor definido para vazão no SLA da aplicação. Para $\lambda = 85\%$:

Condição 1: SE ($\text{vazãoRota} * \lambda \geq \text{vazãoSLA}$)

Ação 1.1: Sistema define uma rota e admite o serviço

- **Políticas de Configuração** – são definidas como um conjunto de regras que determinam a configuração inicial do sistema com base no SLA de uma aplicação. Para isto, relaciona as informações das aplicações, da rede e políticas definidas no sistema. Como a topologia de rede e os requisitos exigidos por um SLA são conhecidos, o sistema de PBQoS através do cruzamento dessas informações, determina quais recursos serão alocados na construção da rota que melhor atenda um serviço ou aplicação. As políticas direcionam o sistema de gerenciamento para melhor definir rotas e escolher recursos para efetiva distribuição de conteúdo em uma rede sobreposta.

Exemplos:

Política de Configuração - Maximizar vazão

=====

Descrição: Vazão resultante fim-a-fim para cada rota. Encontrada através do cálculo do Mínimo Comum das vazões dos enlaces concatenados nesta rota.

=====

//busca pelas rotas possíveis entre origem e destino

Condição 1: SE existirem rotas entre os nós de origem e destino

Ação 1.1: armazenar rotas possíveis fim-a-fim entre origem e destino

// rotas que atendem ao SLA da aplicação

Condição 2: SE (vazão do enlace definido na MIB $\geq$ vazão especificada no SLA)

Ação 2.1: armazenar a rota válida (rota que atende o SLA)

// ordenação crescente pela vazão resultante fim-a-fim de cada rota

Condição 3: SE existirem rotas fim-a-fim que atendem os requisitos especificados no SLA

Ação 3.1: retorna a rota que apresentar maior vazão

Política de Configuração - Minimizar atraso

=====

Descrição: Atraso resultante fim-a-fim para cada rota. Encontrado através da soma dos atrasos de cada enlace

=====

//busca pelas rotas possíveis entre origem e destino

Condição 1: SE existirem rotas entre os nós de origem e destino

Ação 1.1:armazenar rotas possíveis entre origem e destino

// rotas que atendem ao SLA da aplicação

Condição 2: SE o atraso total da rota $\leq$ atraso especificado no SLA

Ação 2.1: armazenar a rota fim-a-fim válida (rota que atende o SLA)

//ordenação pelo atraso fim-a-fim de cada rota

Condição 3: Se existirem rotas fim-a-fim que atendam o requisito atraso especificado no SLA

Ação 3.1: retorna a rota que apresentar menor atraso

Política de Configuração - Minimizar jitter

=====

Descrição: Variação do atraso fim-a-fim resultante para cada rota. Encontrado através da soma da variação do atraso de cada enlace

=====

//busca pelas rotas possíveis entre origem e destino

Condição 1: SE existirem rotas entre os nós de origem e destino

Ação 1.1:armazenar rotas possíveis entre origem e destino

Condição 2: SE a variação do atraso total da rota $\leq$ a variação do atraso especificado no SLA

Ação 2.1: armazenar a rota fim-a-fim válida (rota que atende o SLA)

//ordenação pela variação do atraso fim-a-fim de cada rota

Condição 3: Se existirem rotas fim-a-fim que atendam o requisito variação do atraso especificado no SLA

Ação 3.1: retorna a rota que apresentar menor variação do atraso fim-a-fim

Política de Configuração - Minimizar perdas

=====

Descrição: Perda total fim-a-fim para cada rota. Encontrado através da soma das perdas em cada enlace que compõe a rota

=====

//busca pelas rotas possíveis entre origem e destino

Condição 1: SE existirem rotas entre os nós de origem e destino

Ação 1:armazenar rotas possíveis entre origem e destino

Condição 2: SE a soma das perdas de cada enlace $\leq$ a perda especificada no SLA

Ação 2: armazenar a rota fim-a-fim válida (rota que atende o SLA)

//ordenação pelo requisito perda total de cada rota

Condição 3: Se existirem rotas fim-a-fim que atendam o requisito perda especificado no SLA

Ação 3: retornar a rota que apresentar menor perda fim-a-fim

Política de Configuração - Visão Global de QoS

=====

Descrição: Cálculo dos requisitos QoS (vazão, atraso, variação do atraso e perdas) de forma isolada e através do critério de desempenho por hierarquia de parâmetros os requisitos são integrados.

=====

//busca pelas rotas possíveis entre origem e destino

Condição 1: SE existirem rotas entre os nós de origem e destino

Ação 1.1: armazenar rotas possíveis entre origem e destino

Condição 2: PARA o requisito vazão

Ação 2.1: divisão da vazão especificada no SLA pela vazão total da rota fim-a-fim ($\text{vazão_SLA} / \text{vazãoTotalRota_MIB}$)

Ação 2.2: armazenamento do resultado da ação 2 para cada rota

Condição 3: PARA o requisito atraso

Ação 3.1: divisão do atraso total fim-a-fim pelo atraso especificado no SLA ($\text{atrasoTotalRota_MIB} / \text{atraso_SLA}$)

Ação 3.2: armazenamento do resultado da ação 4 para cada rota

Condição 4: PARA o requisito variação do atraso (jitter)

Ação 4.1: divisão da variação do atraso total fim-a-fim pela variação do atraso especificado no SLA ($\text{jitterTotalRota_MIB} / \text{jitter_SLA}$)

Ação 4.2: armazenamento do resultado da ação 6 para cada rota

Condição 5: PARA o requisito perda

Ação 5.1: divisão da perda total fim-a-fim pela perda especificada no SLA ($\text{perdaTotalRota_MIB} / \text{perda_SLA}$)

Ação 5.2: armazenamento do resultado da ação 6 para cada rota

Condição 6: PARA políticas selecionadas

Ação 6.1: Calcular o produto dos resultados obtidos através das ações 2.2, 3.2, 4.2 e 5.2

//a rota que apresentar o menor valor será considerada a melhor rota

Ação 7: retornar a rota que apresentar o menor valor encontrado através da ação 6.1

-
- **Políticas de Manutenção** – são as regras impostas pelo sistema para mantê-lo operacional sem sobrecarregar os elementos da rede. Dispõe de mecanismos que buscam analisar o perfil de cada aplicação ou serviço admitido e o que dê fato está sendo atendido pelo sistema. Quando um SLA é registrado (admissão de um serviço), o sistema PBQoS ativa as políticas de manutenção para mantê-lo operacional e estabelece alguns critérios para que novas tomadas de decisões possam ser realizadas. Um dos critérios utilizados pelas políticas de manutenção é a ativação das políticas de controle quando alguns limites impostos pelo administrador são atingidos. Além disso, esta política pode solicitar alterações de uma rota ao PDP.

Exemplo:**Política de Manutenção – Manutenção de uma rota**

=====

Descrição: Define regras para manter o sistema operacional e balanceado. Dispõe de mecanismos de manutenção capaz de analisar o perfil de cada aplicação ou serviço admitido e o que dê fato está sendo atendido pelo sistema.

=====

Sendo T a Taxa de ocupação do enlace
 α o fator limite da rota para emissão de um alerta ao administrador ("alerta amarelo")
 β o fator limite da rota para emissão de um alerta ao administrador ("alerta vermelho")

Condição 1: SE ($T \geq \alpha$) && ($T \leq \beta$)

Ação 1.1: Sistema envia um "alerta amarelo" ao administrador da rede (Política de controle)

Condição 2: SE ($T \geq \beta$)

Ação 2.1: Sistema envia um "alerta vermelho" ao administrador da rede

Ação 2.2: Retira a rota admitida pelo serviço e armazena em uma base de dados

Ação 2.3: Recalculo da rota e análise do estado atual da rede

Ação 2.4: Alocação da nova rota para o serviço

6.1.2.3 – Repositório SLA Aplicações

SLA (*Service Level Agreement*) pode ser definido como um acordo formal entre duas partes envolvidas na prestação de um serviço qualquer (fornecedor e cliente). Neste acordo, deve ser definido: todas as atribuições de ambas as partes e resultado esperado do serviço prestado. Restrições envolvidas na prestação de serviço (por exemplo, restrições orçamentárias); data de início e fim do contrato, dentre outras informações. Em outras palavras, trata-se de um acordo de nível de serviço, no qual representa um contrato entre um fornecedor de serviços de TI (Tecnologia da Informação) e um cliente especificando em geral os quais serviços o fornecedor vai prestar. Níveis de serviços são definidos no início de uma relação

de prestação de serviço (*outsourcing*) e usados para mensurar e monitorar o desempenho de um fornecedor

O repositório SLA Aplicações representa a base de dados que armazena os contratos SLAs (*Service Level Agreement*) que o sistema suporta. Conforme já descrito anteriormente, um SLA é um contrato entre um fornecedor de serviços de TI (Tecnologia da Informação) e um cliente, o qual especifica em termos mensuráveis quais níveis de serviços e requisitos de QoS serão prestados pelo provedor de conteúdo (Overby, 2010)].

No contexto deste trabalho, um SLA representa os níveis de serviço e perfis das aplicações que serão definidos no início de uma relação entre consumidor e provedor de conteúdo, e são usados para mensurar e monitorar o desempenho do sistema. Na arquitetura PBQoS, um SLA é composto pelas informações definidas pelo perfil de uma aplicação. Este perfil representa os requisitos mínimos de QoS para garantir a qualidade exigida por uma aplicação. Os perfis de cada aplicação suportado pela arquitetura PBQoS pode ser mapeado e definido através dos requisitos vazão, atraso, variação do atraso e perdas conforme ilustra a Figura 38.

Figura 38 – Perfis de aplicações

6.1.2.4 – PDP

O elemento PDP (*Policy Decision Point*) representa o núcleo de inteligência do sistema, pois é o responsável pela escolha dos nós e enlaces que irão compor

a rota através das políticas de configuração, manutenção e controle. Para realizar esta operação, este elemento tem como base as informações contextuais representadas pelos repositórios MIB, SLA e TED. De acordo com (Wang, 2001), o cálculo de uma rota é o ponto central da Engenharia de Tráfego, pois calcula uma rota com base na demanda de tráfego da rede.

Com base nestas informações, por ser o ponto central da arquitetura proposta neste trabalho, o PDP é capaz de realizar a tomada de decisão no que tange a criação, adaptação e finalização de um serviço. Possui um papel fundamental na arquitetura PBQoS, pois além de administrar os recursos do sistema, realiza também o SLM (*Service Level Management*) de forma automática (cruzamento das informações contextuais da rede com os requisitos impostos pelo SLA de uma aplicação). Este elemento atua como um coordenador de operações, cuja principal função é comunicar-se com os repositórios de políticas, TED e SLA das aplicações, e decidir qual configuração será enviada aos PEPs para efetivá-las na rede. Além disso, pode conectar-se a outros servidores e banco de dados para implementar funcionalidades adicionais como autenticação de usuário e sistema de cobrança, por exemplo.

Um PDP pode ter um ou mais PEPs sob seu controle, enquanto que o PEP pode ter diferentes áreas de aplicação como: segurança, QoS, controle de admissão, dentre outras funcionalidades. Para determinar os nós de rede que irão compor uma rota, o PDP analisa e avalia todas as rotas válidas, aptas a atender o SLA de uma aplicação. Uma rota é a representação de um ou vários enlaces entre dois nós, representando assim uma conexão entre eles. Após avaliar quais rotas atendem aos requisitos mínimos exigidos pelo perfil de uma aplicação, o PDP constrói uma rota para acomodar os fluxos gerados pela aplicação solicitante.

6.1.2.5 – PEP

O elemento PEP (*Policy Enforcement Point*) é um agente responsável por determinar o ponto onde a política será efetivamente aplicada. É o elemento que realiza a mudança de configuração dos serviços nos nós da rede. Seu principal objetivo é controlar os recursos e funcionalidades de um nó, efetuar mudanças de

configuração baseando-se em informações contextuais e usar estas informações (em algumas situações) para gerar políticas de forma dinâmica. Atua como cliente de um PDP e realiza os procedimentos necessários para que as decisões enviadas pelo PDP sejam cumpridas.

Cada PEP ativo na rede manterá uma conexão com seus vizinhos formando pares de nós. Eles irão trocar informações de controle entre si, mantendo atualizado o estado atual de cada enlace. As informações atualizadas de cada enlace são enviadas para todos os nós vizinhos e/ou para uma estação de gerenciamento da rede, que no contexto deste trabalho é representado pelo repositório TED.

Um componente PEP, ao receber uma ordem do PDP, realiza a transformação dessas informações em políticas de configurações, manutenção ou controle, e dessa forma, realiza as alterações de estado das interfaces dos dispositivos e serviços. Quando uma política de configuração é enviada a um PEP, o mesmo deve realizar a tradução desta política em configuração e aplicá-la aos nós da rede. Para isto, cada PEP dispõe de uma MIB local composta pelas informações específicas do nó monitorado e informações extraídas da tabela de roteamento de cada nó. Desta forma, qualquer mudança na MIB de um PEP, implicará em mudanças nos nós da rede sobreposta. As atualizações das MIB's locais de cada nó são realizadas com base nas informações geradas pelo algoritmo OSPF, fazendo com que todos os nós conheçam todos os enlaces da rede.

O OSPF é um protocolo de roteamento para redes IP, que provê de forma eficiente, informações topológicas das redes autônomas. É amplamente desenvolvido e tem os mecanismos necessários para distribuir o estado de cada enlace para os nós interligados e dessa forma, constroi a topologia em um repositório de dados (repositório TED) (Wang, 2001). Utiliza um algoritmo de roteamento do tipo *link-state*, que envia avisos sobre o estado da conexão (*link-state advertisements*, LSA) a todos os outros roteadores em uma mesma área hierárquica, operando dentro de um domínio simples de roteamento (AS – *Autonomous System*). Reúne informações do *link-state* dos roteadores e constrói um mapa da topologia da rede que será apresentada à camada de Internet para tomar decisões de roteamento dos pacotes IP. As informações sobre interfaces ligadas, métrica usada e outras variáveis são incluídas nas LSAs.

Ao mesmo tempo em que o roteador OSPF acumula informações sobre o estado do enlace, ele usa o algoritmo SPF (*Shortest Path First*) para calcular a menor rota para cada nó (RFC 2328) (Perlman, 1999). Dessa forma, as informações atualizadas nos LSAs são replicadas nas MIB's locais dos PEP's que por sua vez atualiza o repositório TED.

Um PEP também pode ser visto como uma entidade colaboradora, pois é o elemento que de fato, efetiva uma determinada configuração aos elementos da rede sobreposta. Quando em estado ativo, seu funcionamento se dá através do recebimento de uma notificação ou uma mensagem dos elementos da rede, que requerem uma decisão de política. Dessa forma, o mesmo constrói uma requisição e a encaminha ao PDP, que por sua vez retorna uma decisão de política que será aplicada de acordo com o perfil da aplicação. Quando passivo, um agente PEP terá como principal funcionalidade, aplicar políticas de acordo com a decisão tomada por um PDP. Vale ressaltar que um PDP pode negar a aplicação de uma determinada política quando esta não representar uma mudança confiável de configuração aos elementos do sistema.

6.1.2.6 – Monitor

O monitor é um componente responsável por coletar as informações de gerenciamento das MIBs presente nos PEPs e enviá-las ao repositório TED (*Traffic Engineering Database*). Esta entidade receptora é responsável por receber os dados de monitoração, analisá-los e gerar estatísticas sobre as métricas coletadas. Com base nessas métricas, o sistema de gerenciamento conhece o estado operacional dos elementos da rede, e assim determina qual política será aplicada para melhor atender os requisitos mínimos exigidos por uma determinada aplicação. Dessa forma, o sistema se torna auto-gerenciável e de forma eficiente realiza uma melhor distribuição dos recursos na rede.

Qualquer mudança na topologia de rede e/ou estado atual dos enlaces deve ser monitorada. Informações dinâmicas como banda disponível ou utilização de um enlace são coletados através dos sistemas de gerenciamento de rede como SNMP (*Simple Network Management Protocol*), Traps e Polling (Wang, 2001). Outra forma de obter estas informações é estender as funcionalidades dos

protocolos de roteamento como o OSPF que através de *broadcast* periódicos, é capaz de manter atualizado o estado de cada enlace monitorado.

6.1.2.7 – Aplicação do Administrador

Localizada na camada BOS, a aplicação de Administração é representada por uma interface gráfica que permite aos administradores criar, atualizar e remover políticas e especificar os SLA's das aplicações nos repositório de políticas e SLA das aplicações respectivamente. Como não é definido o formato padronizado pela IETF para esta ferramenta, sua complexidade varia de acordo com sua capacidade de validar, resolver conflitos e traduzir as políticas para vários níveis de abstração antes de armazená-las no repositório.

6.1.2.8 – Aplicação de Gerenciamento

Trata-se de uma ferramenta localizada na camada SON, permitindo aos usuários visualizar as informações no repositório TED, criar gráficos de desempenho da rede, visualizar estatísticas e perfis dos serviços ativos e inativos no sistema PBQoS. Através desta aplicação, o usuário pode intervir na configuração da rede de forma explícita e alterar uma decisão tomada pelo PDP.

6.2 – Cenário de uso – Arquitetura PBQoS

Após descrever cada elemento que compõe a arquitetura de forma individual e detalhada, é possível definir uma topologia de rede no qual seja possível ilustrar um cenário de uso onde a arquiteturas PBQoS possa ser utilizada. A Figura 39 – ilustra um exemplo da arquitetura PBQoS atuando sobre uma rede nativa gerenciada.

Figura 39 – Cenário de uso – Arquitetura PBQoS

Durante o processo de configuração o sistema leva em consideração diversos fatores para determinar a melhor rota capaz de atender as necessidades de um serviço ou aplicação. Todos os enlaces disponíveis na rede dispõem de parâmetros que representam os requisitos de QoS mapeados no sistema, que por

sua vez, refletem o estado atual da rede. Este processo é realizado através da alimentação periódica do repositório TED.

Baseando-se nos parâmetros determinados por (ITU-T, 2002), uma aplicação VoIP, por exemplo, necessita que a rede atenda o SLA da aplicação com a seguinte configuração: vazão mínima de 64 Kbps, atraso aceitável entre 150 e 400 ms, variação do atraso aceitável entre 20 e 50 ms e uma taxa de perdas menor que 10%. A Figura 40 ilustra a representação de uma rede sobreposta composta por 11 nós interligados entre si, os elementos arquiteturais PBQoS e uma solicitação de recursos ao PDP. Em adição, a Figura 41 ilustra os requisitos de QoS de cada enlace desta rede; por questões de organização, os enlaces aqui definidos são representados por uma letra seguido do identificador dos enlaces em questão.

Figura 40 – Solicitação de rota para um aplicação VoIP

A(1-2) Vazão 2 Mbps Atraso 20 ms Jitter <20 ms Perdas <5%	B(1-4) Vazão 512 kbps Atraso 30 ms Jitter <30 ms Perdas <8%	C(2-4) Vazão 8 Mbps Atraso 5 ms Jitter <20 ms Perdas <2%	D(2-3) Vazão 800 kbps Atraso 200 ms Jitter <20 ms Perdas <4%	E(3-4) Vazão 4 Mbps Atraso 30 ms Jitter <20 ms Perdas <1%	F(3-5) Vazão 600 Atraso 200 ms Jitter <40 ms Perdas <8%
	H(5-6) Vazão 128 kbps Atraso 130 ms Jitter <50 ms Perdas <20%	I(6-7) Vazão 256 kbps Atraso 450 ms Jitter <40 ms Perdas <8%	J(4-8) Vazão 10 Gbps Atraso 5 ms Jitter <5 ms Perdas <1%	K(6-8) Vazão 64 kbps Atraso 420 ms Jitter <70 ms Perdas <10%	L(8-9) Vazão 512 kbps Atraso 50 ms Jitter <20 ms Perdas <3%
N(7-11) Vazão 1 Gbps Atraso 10 ms Jitter <10 ms Perdas <5%	O(9-10) Vazão 256 kbps Atraso 450 ms Jitter <40 ms Perdas <8%	P(10-11) Vazão 512 kbps Atraso 70 ms Jitter <30 ms Perdas <3%	Q(8-11) Vazão 2 Mbps Atraso 20 ms Jitter <10 ms Perdas <3%		

Figura 41 – Representação do estado de tráfego dos enlaces da rede

Quando ocorre uma solicitação de recursos ao sistema PBQoS, o primeiro componente que entra em operação é o PDP. Através das informações obtidas no processo de solicitação (SLA aplicação + Política), o PDP verifica os enlaces da rede que estão aptos a atender o SLA requisitado e através da política de gerenciamento selecionada, o sistema determina a rota que melhor atende a solicitação, e então a envia aos PEP's responsáveis pela efetivação da configuração na rede. Detalhe de como o sistema configura, controla e realiza a manutenção das rotas serão descritos nas sessões seguintes.

6.2.1 – Configuração do Sistema

Para que a configuração inicial do sistema seja realizada, o PDP precisa conhecer o tipo de serviço solicitado (perfis de aplicações) e a política que irá determinar o funcionamento do sistema. Conforme já definido, as políticas de configuração previamente definidas são: *maximizar de vazão*, *minimizar atraso*, *variação do atraso e perdas* e *visão global de QoS*.

Conforme ilustra a Figura 42, ao receber uma solicitação (1), o PDP consulta o repositório SLA para obter o perfil da aplicação solicitante (2). Com as informações que representam o perfil da aplicação e o tipo de política selecionada, o PDP verifica o estado atual da rede através do repositório TED, e identifica os enlaces que dispõem dos recursos necessários para que o atendimento da solicitação seja realizado (3). Dessa forma, o PDP concentra todas as informações necessárias para que a tomada de decisão possa ser realizada. Conhecendo o perfil da aplicação, a política requerida e o estado atual da rede, o PDP faz uma análise destas informações e determina a melhor rota para atender um serviço através das *políticas de configuração* (4). Após este processo, o PDP envia uma sinalização aos PEPs (5), que por sua vez configura os elementos da rede sobreposta.

Figura 42 – Configuração inicial do Sistema

Conforme já mencionado anteriormente, a arquitetura PBQoS suporta os requisitos de QoS como: vazão, atraso, variação do atraso e perdas. O requisito *custo* definido pela MIB na Figura 42, é meramente ilustrativo como um serviço que pode ser adicionado futuramente.

6.2.2 – Manutenção e Controle de uma rota

Conforme ilustra a Figura 43, após definir e aplicar a configuração inicial nos enlaces da rede sobreposta **(1)**, o PDP registra o SLA (configuração inicial) no repositório de SLA contratado **(2)**; Além disso, o PDP ativa o processo de monitoração do serviço contratado, alimentando o repositório TED (MIB serviço) com as informações atuais do serviço **(3)**. Buscando cumprir o contrato requerido pelo cliente, o sistema através do SLM automático faz um cruzamento de informações entre o status do serviço e os requisitos especificados no SLA contratado **(4)**. Uma vez que os limites pré-estabelecidos pelas políticas de configuração foram atingidos, as *Políticas de Manutenção* são acionadas em busca de uma nova configuração para o sistema **(5)**. Após ativar este mecanismo, o sistema recalcula uma nova rota que possa atender a solicitação de um serviço sem sobrecarregar a rede e uma nova configuração é determinada. No momento em que uma nova rota for definida, o PDP envia esta nova configuração aos PEPs que por sua vez, as aplicam nos nós da rede **(6)**.

Figura 43 – Manutenção de uma rota

Uma política de manutenção deve ser aplicada toda vez que os limites definidos pelas políticas de configuração forem atingidos. Para acionar este mecanismo, o sistema de gerenciamento deve conhecer a MIB do serviço, que por sua vez, representa os requisitos de QoS definidos no SLA contratado e a rota definida para atender esta solicitação. Através destas informações, o PDP pode aplicar uma política de manutenção que melhor se adapte ao serviço contratado. Uma representação deste processo pode ser visualizada através da Figura 44.

Figura 44 – Manutenção de uma rota

Visando detalhar o mecanismo que permite realizar a manutenção e controle de um serviço, foi escolhido o requisito *atraso* da rota. No exemplo ilustrado pela Figura 45 é possível observar que no momento em que o consumo de um enlace atingir α , as Políticas de Controle são ativadas e um “alerta amarelo” é enviado ao administrador da rede para possíveis intervenções. No contexto deste trabalho, α representa 70% da capacidade total do enlace.

Figura 45 – Manutenção e Controle – “alerta amarelo” ($\alpha = 70\%$)

De forma análoga ao alerta amarelo, o sistema PBQoS conta com políticas de controle que enviam alertas vermelhos ao administrador da rede toda vez em que os recursos consumidos de um enlace exceder o limite máximo dos recursos disponíveis na rota definido pelo administrador da rede. Para ilustrar este cenário, apenas o requisito atraso será ilustrado, visto que este critério pode ser aplicado aos demais requisitos de QoS definidos neste trabalho. No momento em que um enlace atingir β , o PDP é acionado para recalculer uma nova rota que seja capaz de atender ao SLA especificado. Este processo cria uma tabela virtual, representada pela MIB do serviço o qual retira a rota anteriormente definida e realiza o recálculo da nova rota visando normalizar a rede e atender os requisitos especificados pelo SLA contratado. Uma vez recalculada a rota e visto que o sistema será normalizado, esta nova configuração pode ser aplicada ao sistema e uma nova rota é armazenada na MIB do serviço. Um exemplo de como as Políticas de Manutenção e Controle faz a gestão do processo de manutenção de rota pode ser visualizado na Figura 46.

Figura 46 – Manutenção e Controle – “alerta vermelho” ($\beta = 85\%$)

Para o cliente que solicita recursos ao sistema de gerenciamento PBQoS, a definição da rota se dá de forma transparente conforme ilustra a Figura 47.

Figura 47 – Abstração da definição de uma rota pelo PDP

Quando um serviço é solicitado ao sistema de gerenciamento, o PDP entra em operação para que a melhor rota disponível na rede seja definida e alocada ao usuário. Ao receber o tipo de serviço requerido e a política desejada, o PDP dispõe de mecanismos que abstrai todo o processo de definição de rotas e direciona a configuração para os PEPs ativos na rede. Neste cenário, a rede nativa representa os elementos físicos de rede constituída por roteadores, servidores, computadores, dentre outros.

Alguns elementos da rede nativa podem ser mapeados em um nível de abstração superior, formando assim uma rede sobreposta que será controlada pelos agentes PEPs. O PDP é representado por um super nó capaz de definir,

controlar e manter uma configuração e enviá-la aos agentes PEPs, que por sua vez, as aplicam nos nós da rede sobreposta.

6.2.3 – Escalabilidade da arquitetura PBQoS

Em comunicação de dados e engenharia de sistemas, escalabilidade é uma propriedade de um sistema, de uma rede de dados, ou até mesmo de um processo como um todo, que indica que essa entidade pode crescer, ou até mesmo reduzir de tamanho de uma maneira elegante do ponto de vista de engenharia (Bondi, 2000). A característica escalabilidade está diretamente ligada ao binômio desempenho versus recurso consumido.

Escalabilidade, como propriedade de um sistema, é difícil de definir de uma maneira formal. A primeira dificuldade a ser enfrentada é determinar qual ou quais dimensões do sistema são importantes para serem usadas na avaliação de escalabilidade (Hill, 1990). Por exemplo, um sistema pode ser dito escalável, quando sua capacidade de vazão aumenta proporcionalmente ao incremento de capacidade de hardware que lhe foi aplicado. Nesse caso, está se utilizando o quesito capacidade de vazão.

Outro quesito importante que pode ser analisado é a eficiência. Dir-se-ia então que um sistema é escalável quando ele mantém a eficiência mesmo quando utilizado em estruturas maiores. Dessa maneira, a escalabilidade pode ser medida de diversas maneiras, em mais de uma dimensão (Hill, 1990):

- Escalabilidade de carga: habilidade de um sistema distribuído ser facilmente expandido ou contrair suas reservas de recursos para se adaptar a diferentes tipos de cargas;
- Escalabilidade geográfica: habilidade de manter o desempenho, funcionalidades ou usabilidade, independentemente da atuação do sistema em áreas geográficas mais concentradas ou mais distribuídas;
- Escalabilidade administrativa: habilidade de um sistema distribuído ser compartilhado por um número maior ou menor de organizações usuárias;
- Escalabilidade funcional: habilidade de um sistema suportar um número maior de funções através de um aumento mínimo de esforço ou recursos.

Partindo do princípio inicial de sistemas distribuídos, existem dois métodos para acrescentar mais recursos (Michael, 2007): aumento de capacidade de um simples nó, chamado de escalonamento vertical (*scale up*); e aumento do número de nós, denominado escalamento horizontal (*scale out*).

Analisando o sistema PBQoS com o objetivo de chegar a algumas conclusões sobre suas características ou habilidades de escalabilidade, chega-se aos seguintes aspectos:

- Sua arquitetura padrão, derivada da arquitetura de PBNM possui três níveis de sub-sistemas: os PEP's, o PDP, e o Repositório de Políticas;
- A comunicação entre PEP's e o PDP é feita através de um protocolo simples, de pergunta/resposta, que é o COPS;
- A comunicação entre o PDP e o repositório é feita, preferencialmente, via LDAP, que também é um protocolo simples de pergunta/resposta;
- Não há uma definição nos padrões do IETF para PBNM, de uma estrutura de distribuição da operação dos PDP's. A visão atual de implementação é de uma estrutura centralizada, ou, de domínio de operação de um PDP;
- As políticas são utilizadas e os PDP's atuam em dois grandes momentos: no estabelecimento de um conexão ou fluxo e na manutenção de seus parâmetros de QoS durante o tempo de vida dessa conexão ou fluxo.

Dessa maneira, o impacto da operação do sistema PBQoS sobre a infraestrutura não é muito diferente de tantos outros sistemas distribuídos de controle sobre redes de computadores, inclusive de sistemas de gerência de rede em geral. Analisando a arquitetura proposta neste trabalho (Figura 35), têm-se os seguintes pontos a serem considerados:

1. Carga de processamento de controle sobre os equipamentos hospedeiros dos PEPs. Normalmente são equipamentos ativos de comunicação de dados;
2. Carga de rede para a comunicação entre o PDP e os PEPs. Considerando inclusive o aumento concentrado de tráfego à medida que se chega mais perto do PDP na estrutura de rede. Esse é um fenômeno que ocorre também nas arquiteturas de gerência de rede;

3. Carga de processamento sobre o PDP;
4. Carga de processamento sobre o Repositório de Políticas;
5. Aumento de carga de rede devido a uma maior atividade de gerência de rede, no que tange a atualização de informações de engenharia de tráfego;
6. Eventual aumento de carga na rede devido a mudanças mais acentuadas de configuração, provocando atividade em protocolos de roteamento.

Considerando o escopo deste trabalho, que foi a implementação de uma prova de conceito da arquitetura e de um conjunto básico de políticas, tem-se como condições de contorno uma arquitetura centralizada com um PDP, um conjunto de PEP's, um repositório de políticas e um repositório de SLA's.

Dessa maneira, os pontos importantes para análise de escalabilidade são o próprio PDP, a comunicação com os PEP's e a comunicação com os Repositórios. As simulações feitas provaram a funcionalidade da arquitetura, mas sem modelar os aspectos de comunicação da rede de suporte e nem das características do protocolo COPS.

Portanto, é possível fazer uma análise qualitativa da escalabilidade, verificando a quantidade de mensagens e interações entre os componentes, mas sem a análise quantitativa. Dentre os pontos analisados, destacam-se:

- **Carga de processamento de controle sobre os equipamentos hospedeiros dos PEP's** – A carga de processamento sobre os equipamentos de rede será muito pequena, pois significará, na maior parte dos casos, mudanças de configuração de operação.
- **Carga de rede para a comunicação entre o PDP e os PEP's** – Para análise desse ponto, é importante verificar com um pouco de detalhe o que acontece, com relação às interações de rede, principalmente na atuação do PBQoS. Partindo de um cenário composto de uma rede de 4 nós, tem-se uma estrutura composta por um PDP e um PEP por nó. Dessa maneira, tem-se a seguinte situação de interações de rede:
 - 1 mensagem de solicitação do cliente
 - 2 mensagens (do PDP para Repositório de SLA's) e (do PDP para o TED)
 - N mensagens do PDP para os PEP's (uma para cada PEP que compõem a rota). Onde o N será maior do que o número de nós da rede, sendo apenas uma para cada nó);

Dessa forma, é possível dizer que o aumento de carga sobre a rede, devido à comunicação entre PDP e PEPs depende linearmente do número de nós afetados na rota. De uma maneira geral, em redes de porte, uma rota não ocupa todos os nós. Dessa maneira, pode-se dizer que o aumento de carga sobre a rede pode ser significativo, mas não afetando a escalabilidade da solução.

- **Carga de processamento sobre o PDP e o Repositório de Políticas** – é possível dizer que um aumento na capacidade de processamento dos servidores hospedeiros dos sub-sistemas PDP e Repositório de Políticas pode resolver o problema de escalabilidade. Um aumento desproporcional pode ser necessário, no caso de uso de políticas mais complexas, utilizando modelos de predição, algoritmos complexos ou dados históricos de rede armazenados em outros sistemas. Isso deve ser tratado de em uma linha de trabalhos de estudos de políticas, importante para trabalhos futuros.
- **Aumento de carga de rede devido a uma maior atividade de gerência de rede** – O PDP utiliza informações de tráfego na rede, armazenadas no TED, para tomar decisões. Essas decisões irão alterar as configurações da rede, estabilizando-a em uma nova situação de carga e de suporte aos SLA's que estão gerenciando. Dessa maneira, existirá um aumento de tráfego de informações de gerência de rede, provocado pela atuação da estrutura do sistema PBQoS. Entretanto, essa atividade acontecerá somente em duas situações:
 - Quando certos gatilhos de limites de necessidade de emissão de notificações ocorrem;
 - No caso de, intencionalmente, a frequência de atualização de informações de tráfego for aumentada, para diminuir o tempo de resposta do sistema como um todo, inclusive do sistema PBQoS.

O primeiro item depende do grau de intensidade da mudança que o sistema PBQoS provoca na configuração do equipamento. Esse grau de alteração depende tanto da situação da rede no momento da atuação, quanto do porte da necessidade apontada pelo usuário do fluxo. Ambos os casos dependem de alguns fatores combinados, levando à necessidade e oportunidade de estudos futuros. De qualquer maneira, pode-se dizer

que apenas uma percentagem das incursões do PBQoS irá provocar essas interações na rede, indicando que não será um item a impactar a escalabilidade.

- **Operações de roteamento** – Este item recai no mesmo caso da atividade de gerência de rede, pois é dependente das condições já existentes de carga antes da operação do sistema PBQoS e da profundidade da alteração que ele provocou. Com a rede mais folgada, trabalhando razoavelmente longe de seus limites, ocorrerão poucas alterações de roteamento provocadas pelo sistema PBQoS. Isso já não ocorreria no caso de uma rede próxima de seus limites de carga. Considerando a ocorrência de alterações fortes na rede, teríamos a situação a seguir: Analisando a troca de mensagens entre os nós que compõe a rede, a situação deixa de ser proporcional e parte para o exponencial. Considere o esquema simples composto por 2 nós ilustrado pela Figura 48:

Figura 48 – Cenário composto por 2 nós

A Figura 49 ilustra um cenário composto por 4 nós.

Figura 49 – Cenário composto por 4 nós

Por fim, a
composto por 6 nós.

Figura 50 ilustra um cenário

Figura 50 – Cenário composto por 6 nós

Com base nestes cenários, foi possível analisar que a troca de mensagens entre nós, aumenta exponencialmente em função do número de conexões entre um nó e seus vizinhos. Quanto maior o número de vizinhos (conexões), maior será a troca de mensagens entre eles. A partir das considerações realizadas, é possível chegar a duas conclusões:

1. A carga sobre a rede devido ao funcionamento do PBQoS parece ser bastante linear e ocorrendo normalmente em partes da rede, raramente em seu todo;
2. Essa carga é fortemente dependente das políticas em uso e da situação geral da rede.

Capítulo 7

Algoritmo de cálculo de rota – PBQoS

Para realizar todo o processo de busca pela melhor rota, o sistema PBQoS define um algoritmo híbrido responsável por minimizar o consumo total de recursos por rota. Através de diversos mecanismos, este algoritmo tenta evitar a sobrecarga nos enlaces que comporão uma rota, maximizando a capacidade residual da rede. Trata-se de um processo de otimização local no sentido de não considerar demandas futuras.

Este algoritmo utiliza diversos mecanismos que permitem determinar e manter uma rota capaz de atender o SLA de uma aplicação. Detalhes do funcionamento deste algoritmo serão descritos nas seções 7.1 e 7.2.

7.1 – Processo de definição da melhor rota

Nas áreas da Matemática e Ciência da Computação, otimização ou programação matemática refere-se ao processo de escolha do melhor elemento dentro de um conjunto de alternativas disponíveis. De forma simples, isto corresponde a resolução de problemas onde se pretende minimizar ou maximizar uma função real sistematicamente escolhendo os valores das variáveis reais ou inteiras dentro de um conjunto previamente definido (Elster, 1993)(Panos, 2000).

Uma definição mais genérica da teoria de otimização poderia ser definida pelo processo de encontrar os “melhores” valores de uma função objetivo dado um domínio definido, incluindo uma variedade de diferentes tipos de funções objetivo e domínios. De acordo com Ausiello (2003), um problema de otimização pode ser representado da seguinte forma:

Dada uma função $f: A \rightarrow \mathbf{R}$ de algum conjunto $\mathbf{A}$ de números reais, procura-se um elemento x_0 em A de tal forma que $f(x_0) \leq f(x)$ para todo x em $\mathbf{A}$ (“*minimização*”) ou de tal forma que $f(x_0) \geq f(x)$ para todo x em $\mathbf{A}$ (“*maximização*”). Esta representação é conhecida como *Problema de Otimização* ou *Problema de Programação Matemática* (um termo que não está diretamente relacionado com a programação de computador, mas ainda em uso, como ocorre em programação

linear). Diversos problemas podem ser modelados usando este conceito, na área da Física e Computação, por exemplo, pode referir-se a técnica de *minimização de energia*, onde o valor da função f representa a energia do sistema que está sendo modelado (Hromkovic, 2002) (Mordecai, 2003).

Normalmente, A é um subconjunto do *espaço Euclidiano* $\mathbb{R}^n$ freqüentemente especificado por um conjunto de restrições, igualdades e desigualdades onde os elementos de A tem que satisfazer. O domínio A de f é chamado de espaço de busca ou conjunto de escolhas, enquanto os elementos de A são chamados de soluções candidatas ou soluções possíveis. A função f é chamada *função objetivo*, onde uma solução possível que minimiza (ou maximiza), a função objetivo é chamada de *solução ótima* (Elster, 1993) (Panos, 2000). Neste trabalho, o subconjunto A é formado por todas as rotas definidas na MIB.

Pensando assim, a implementação do sistema PBQoS recai em um problema de otimização. Desta forma, deseja-se saber qual a melhor rota entre dois pontos (origem e destino), dado um conjunto de rotas, cada qual definida pelos valores de vazão, perda, variação do atraso e atraso. Sabe-se que o conceito de *melhor* é subjetivo, e dessa forma, é necessário estabelecer um critério objetivo para se definir a “melhor rota”. Colocando sob o ponto de vista matemático, com base na definição de Hromkovic (2002) e Mordecai (2003) pode-se representar o problema de duas formas:

(a) Para os requisitos atraso, variação do atraso e perda:

1. Dado $f: A \rightarrow \mathbb{R}$, $f = f(\text{perda, variação do atraso, atraso})$
2. A é formado por todas as rotas definidas na rede(mib).
3. Sujeito a um conjunto de restrições (SLA e políticas)
4. Encontrar x_0 em A tal que $f(x_0) \leq f(x)$ para todo x em A .

(b) Para o requisito vazão:

1. Dado $f: A \rightarrow \mathbb{R}$, $f = f(\text{vazão})$
2. Sujeito a um conjunto de restrições (SLA e políticas)
3. Encontrar x_0 em A tal que $f(x_0) \geq f(x)$ para todo x em A .

É importante ressaltar que para os requisitos atraso, variação do atraso e perda, o sistema busca pelos enlaces que apresentam os menores valores para

cada requisito, enquanto que o requisito vazão, o sistema busca pelos enlaces que apresentam as maiores velocidades.

Existem vários métodos para resolução de problemas de otimização (Rich, 1990) (Yang, 2008) (Mordecai, 2003). Para resolver o problema de otimização proposto neste trabalho, foi elaborado um algoritmo baseado no modelo proposto por Cardoso (1993) e Cardoso (1994). Neste algoritmo, é realizada uma busca exaustiva da função f , calculando-se o seu valor em todos os pontos do domínio A' , definido pelas rotas válidas existentes no conjunto A . Conforme já definido por Elster (1993), a função f , chamada de *função objetivo*, é calculada em todos os pontos do domínio e neste trabalho, a rota que apresentar o menor valor de f é escolhida como a melhor rota disponível no momento da solicitação.

Avaliando as variáveis de estado do problema (vazão, perda, atraso e variação do atraso), pode-se chegar a algumas conclusões importantes para a definição da função f :

Os menores valores dos requisitos perda, atraso e variação do atraso indicam a escolha que parece ser a melhor no momento (escolha ótima local); enquanto que para o requisito vazão, o maior valor representado, indica a escolha que parece ser a melhor no momento (escolha ótima local). Dessa forma, a função f deve contabilizar o exposto em (a) e (b) da página anterior.

Para efeito de comparação de rotas, é necessário que os valores das variáveis de estado sejam desprovidos de qualquer unidade física, pois as mesmas podem ser representadas por diferentes valores e unidades. Essas variáveis devem representar o produto ou quocientes de quantidades cujas unidades se cancelam e dependendo do seu valor. Estes números têm um significado físico que caracteriza determinadas propriedades para alguns sistemas. Pensando assim, é adequado adimensioná-las sob algum critério.

Conforme descrito no Capítulo 6, no sistema PBQoS, as variáveis de estado foram adimensionadas pelos valores correspondentes definidos no SLA. Foram consideradas as políticas selecionadas (ativas) para definição da função f . Deste modo, caso apenas uma política seja selecionada, a melhor rota dentre as possíveis, que doravante neste trabalho será chamada de rota ótima, será calculada por:

$$f = \min \left\{ \left(\frac{Q_i^{ROTA}}{Q^{SLA}} \right)^z \right\} \begin{cases} \text{para } Q = \text{variável de estado selecionada} \\ \text{para } i = 1 \dots N_{\text{rotas possíveis}} \\ z = -1 \text{ para vazão e 1 caso contrário} \end{cases}$$

Caso mais de uma política seja selecionada, o valor da função f será calculado por:

$$f = \min \left\{ \sum_{j=1}^k \left(\frac{Q_{ij}^{ROTA}}{Q_j^{SLA}} \right)^z \right\} \begin{cases} \text{para } Q = \text{variável de estado selecionada} \\ \text{para } i = 1 \dots N_{\text{rotas possíveis}} \\ \text{para } k = \text{número de políticas selecionadas} \\ z = -1 \text{ para vazão e 1 caso contrário} \end{cases}$$

É importante ressaltar que quando mais de uma política estiver ativa, o sistema realiza a contabilização definida em a) e b) em busca do menor valor para a função f (obtida através das rotas válidas). Dessa forma, pode-se dizer que, dentro das restrições e contexto, o sistema é capaz de encontrar a solução ótima global. O cálculo para encontrar o valor para a função f será descrito detalhadamente na sessão 7.2.

7.2 – Definição do algoritmo PBQoS

Com base nos métodos de otimização analisados (Rich, 1990) (Yang, 2008) (Mordecai, 2003) (Cardoso, 1993) e (Cardoso, 1994) foi elaborado um algoritmo que pudesse atender as necessidades do sistema de gerenciamento PBQoS. O algoritmo busca encontrar a melhor rota possível, levando em consideração as restrições impostas pelas políticas de gerenciamento.

Para ilustrar os mecanismos e critérios detalhados adotados no processo de definição de uma rota, foi assumido um grafo direcionado contendo cinco (5) nós (A, B, C, D e E) conforme ilustra a Figura 51.

Figura 51 – Representação abstrata de uma rede

Para contextualizar a utilização deste algoritmo, considere uma solicitação de rota entre os nós A e E (A→E) para uma aplicação VoIP. Conforme definido por ITU-T (2002), uma aplicação VoIP requer uma configuração mínima da rede para que a mesma possa operar de forma eficiente. A Figura 52 ilustra uma configuração de SLA para uma aplicação VoIP.

Figura 52 – SLA para uma Aplicação VoIP

De todos os enlaces disponíveis na MIB, o sistema deve realizar o processo de filtragem em busca de separar os enlaces válidos que possam ser utilizados na composição da rota para atender o SLA requisitado. Com base nas informações definidas no SLA da aplicação VoIP e nas informações na MIB geral do sistema atualizadas pelo algoritmo OSPF, o sistema verifica as possíveis rotas entre os nós A e E e calcula o QoS fim-a-fim para cada rota encontrada. Um exemplo da MIB geral do sistema pode ser visualizado na Tabela 13.

Tabela 13 – Representação do estado atual da rede – MIB

MIB Geral Sistema				
Enlaces	Vazão	Perdas	Jitter	Atraso
AB	100 Mbps	2%	1 ms	100 ms
BC	16 Gbps	0%	0 ms	0 ms
BE	8 Mbps	2%	20 ms	5 ms
CE	600 Kbps	4%	20 ms	200 ms
DE	800 Kbps	1%	20 ms	30 ms

Com base nas informações definidas na MIB, juntamente com as informações de configuração definidas pelo SLA, o sistema determina quais enlaces estão aptos a serem utilizados na composição de uma rota. As regras impostas pelo sistema na definição de uma rota são determinadas pelas políticas de configuração.

Visando ilustrar este mecanismo, considere que a política *Visão Global de QoS* tenha sido ativada no sistema. Quando ativa, esta política faz com que as

políticas *maximizar vazão*, *minimizar atraso* e *variação do atraso* e *minimizar perdas* sejam executadas. Mas para isso, define uma hierarquia de parâmetros, tornando possível encontrar uma rota válida dentre as possíveis.

Para iniciar o processo, o sistema faz uma leitura das informações atualizadas na MIB geral do sistema e define as rotas possíveis entre origem e destino. Assumindo que diversas rotas tenham sido encontradas, o sistema calcula o QoS fim-a-fim de cada rota encontrada.

Seguindo os critérios adotados pela hierarquia de parâmetros, o sistema inicia todo o processo executando a política *maximizar vazão*. Esta política ao entrar em operação, realiza uma análise do requisito *vazão* presente em cada e verifica quais rotas possuem vazão suficiente para atender a configuração definida pelo SLA.

De acordo com Lowekamp (2003), a vazão de um enlace (ou de uma rota) é definida pela capacidade máxima disponível deste enlace ou rota. No momento em que uma rota é composta por diversos enlaces, o enlace com a menor taxa de transmissão determina a capacidade do enlace. Para maximizar a vazão de uma rota, o sistema utiliza o critério definido por Lowekamp (2003), no qual a vazão resultante fim-a-fim para cada rota é definida pela velocidade do menor enlace desta rota (Mínimo Comum dos enlaces concatenados em cada rota). Para ilustrar este processo considere o grafo definido pela Figura 53.

Figura 53 – Representação dos enlaces para requisito vazão

Com base na solicitação de uma rota entre A e E para atender ao SLA de uma aplicação VoIP, é possível observar que existem duas rotas possíveis capazes de atender a requisição. A rota $A \rightarrow B \rightarrow E$ dispõe de uma vazão fim-a-fim de 8 Mbps, enquanto que a rota $A \rightarrow B \rightarrow C \rightarrow E$ dispõe de uma vazão fim-a-fim de 600 Kbps. Ambas as rotas possuem vazão suficiente para atender ao SLA da aplicação, que por sua vez requer uma rota com vazão mínima de 64

Kbps. Como a política busca a maximização da vazão, a rota selecionada será composta pelos nós A-B-E, no qual dispõe de uma vazão fim-a-fim de 8 Mbps.

Uma vez encontrada rotas que atedem ao SLA com base na vazão total de uma rota, o sistema inicia o segundo processo em busca do atraso médio total de uma rota ($\sum$ atrasos). De acordo com a definição da RFC 2679, o atraso de um enlace é definido por um número real, calculado entre pacotes selecionados (dois ou mais). O cálculo do atraso é iniciado no momento em que o emissor envia o primeiro bit de um pacote em um determinado tempo T , e o receptor recebe o último bit deste mesmo pacote em $T+Tdt$. Caso o receptor não receba o pacote enviado pelo emissor, o número que define o atraso em segundos é considerado indefinido.

Conforme define a RFC 2679, o atraso unidirecional (d^i) é definido como sendo o tempo que o pacote i demora a sair de um ponto A e chegar a outro ponto B . Para obtenção desta métrica, tanto A como B precisam ter os relógios sincronizados com uma boa precisão. Caso haja fragmentação do pacote analisado, o atraso é dado em função do último fragmento recebido (caso todos tenham sido recebidos).

Com base na definição de Mieghem (2006) em que o atraso fim-a-fim de uma rota, pode ser calculado através da soma dos atrasos individuais em cada enlace que compõe a rota, o sistema analisa cada enlace da rede em busca de compor uma rota que apresente o menor atraso fim-a-fim entre os nós A e E ($A \rightarrow E$). Para ilustrar este conceito, considere o grafo definido pela Figura 54.

Figura 54– Representação dos enlaces para o requisito atraso

De acordo com a definição de ITU-T (2002), uma aplicação VoIP requer uma configuração mínima na rede suporte um atraso menor que 400 ms. Como é possível observar, todos os enlaces (exceto o enlace entre os nós D e E, por não participar da rota solicitada).

Após realizar o cálculo para encontrar o atraso médio fim-a-fim de cada rota válida, o sistema apresenta duas possibilidades: a rota $A \rightarrow B \rightarrow E$ possui atraso total de 28 ms, enquanto que a rota $A \rightarrow B \rightarrow C \rightarrow E$ dispõe de um atraso total de 24 ms. Como a política busca pelo atraso médio total fim-a-fim ($\sum$ dos atrasos de cada enlace que compõe a rota) de cada rota, a rota $A \rightarrow B \rightarrow C \rightarrow E$ será selecionada, por apresentar um atraso total fim-a-fim menor que a rota $A \rightarrow B \rightarrow E$.

Também conhecido como *jitter*, a variação do atraso em pacotes IP (IPDV – *Internet Packet Delay Variation*) é calculado em pacotes pertencentes a um mesmo fluxo. A variação do atraso é definida em um par de pacotes selecionados através de dois pontos de medição MP1 e MP2 (*Measurement Point*). Quando um fluxo de pacotes P é enviado entre origem e destino, ao passar pelo primeiro ponto de medição (MP1) o primeiro pacote recebe o índice zero, enquanto que os pacotes subsequentes ao passarem pelo mesmo ponto de medição receberão um índice maior que o anterior. Geralmente este índice é fornecido de forma incremental para cada pacote do fluxo (RFC 3393).

Visando ilustrar este conceito, considere T_1 o momento em que o emissor enviar o primeiro bit do pacote, e T_2 o tempo em que o emissor envia o primeiro bit do segundo pacote. A variação do atraso nestes pacotes é definida pela diferença de tempo entre esses dois pacotes selecionados pela função f entre dois pontos de medição. Considerando que o emissor tenha enviado dois pacotes, o primeiro no tempo T_1 , e o segundo no tempo T_2 (a medição inicia sempre no primeiro bit de cada pacote), e os pacotes foram recebidos pelo destino no tempo $dT_1 + T_1$, onde dT_1 representa o tempo em que foi recebido o último bit do primeiro pacote, e no tempo $dT_2 + T_2$, (último bit do segundo pacote). Dessa forma, a variação do atraso entre origem e destino no ponto T_1 e T_2 é definido por $ddT = dT_2 - dT_1$.

Vale ressaltar que a variação de atraso entre os tempos T_1 e T_2 é considerada *indefinida* quando o emissor envia um pacote em T_1 e o outro pacote em T_2 e, por alguma falha na rede, o receptor não recebe nenhum dos pacotes enviados. A Figura 55 ilustra esta definição. Supõe que os pacotes $P(i)$ e $P(k)$ foram selecionados.

Figura 55 – Definição de variação de atraso
(RFC 3393)

De acordo com a definição da RFC 3393, a variação do atraso em pacotes IP segue os princípios definidos na teoria das variáveis aleatórias independentes, pois a ocorrência de um evento em e em um dado intervalo de tempo t , é independente dos outros eventos que possam ocorrer.

Uma variável aleatória pode ser considerada como o resultado numérico de operar um mecanismo não determinístico ou de fazer uma experiência não determinística para gerar resultados aleatórios. Intuitivamente, uma variável aleatória pode ser vista como a medição de um parâmetro que pode gerar um valor diferente a cada medida.

Matematicamente, Aczel (2009) define variável aleatória como sendo uma função mensurável de um espaço de probabilidade para um espaço mensurável. Este espaço mensurável é o espaço de possíveis valores da variável, e é normalmente tomado como a *sigma-álgebra de Borel* baseada na topologia usual dos números reais. Dessa forma, ela pode armazenar um valor incerto, cujo valor depende do acaso. Possui uma regra que atribui probabilidades com diferentes valores em variáveis aleatórias (lei da probabilidade). Uma variável aleatória pode assumir valores discretos e contínuos.

Uma variável aleatória é discreta se a sua distribuição de probabilidade é discreta. Uma distribuição de probabilidade discreta é tal que é completamente caracterizada pela sua função massa de probabilidade. Assim, X é uma variável aleatória discreta caso:

$$\sum_u P(X = u) = 1$$

com u percorrendo todo o conjunto de possíveis valores da variável aleatória X . Algumas das mais conhecidas distribuições de probabilidade discretas são: Distribuição de Bernoulli, Distribuição geométrica, Distribuição de Poisson, dentre outras. Se uma variável aleatória é discreta então o conjunto de

todos os possíveis valores que ela pode assumir é finito ou enumeravelmente infinito, porque a soma de muitos números positivos reais enumeráveis diverge sempre para o infinito (Aczel, 2009).

Uma variável aleatória contínua pode assumir qualquer valor em um intervalo de números. Os valores das variáveis contínuas podem ser medidos em qualquer grau de precisão. A distribuição da probabilidade de uma variável aleatória X deve satisfazer as seguintes condições:

$$P(x) \geq 0 \text{ para todos os valores } x, \text{ e}$$

$$\sum P(x) = 1$$

Todo x

Para cada valor de x , $P(x) = P(X=x)$ a probabilidade do evento cuja variável aleatória seja igual a x . Por definição, x representa todos os valores que a variável aleatória X pode assumir desde que X assumira apenas um valor por vez. A ocorrência desses valores são eventos mutuamente exclusivos. Entretanto, a soma de todas as probabilidades $P(x)$ deve ser 1.00 (Aczel, 2009).

Dependendo do contexto utilizado, tem-se interesse em somar diversas variáveis aleatórias, importando apenas a composição linear de diversas variáveis aleatórias. Uma composição linear de variáveis aleatórias $X_1, X_2, \dots, X_k$ pode ser apresentada da seguinte forma:

$$a_1X_1 + a_2X_2 + \dots + a_kX_k, \text{ onde } a_1, a_2, \dots, a_k \text{ são constantes.}$$

Considerando que $X_1, X_2, \dots, X_k$ sejam valores aleatórios de k diferentes itens que foram comprados em uma loja, e $a_1, a_2, \dots, a_k$ seus respectivos preços. Então $a_1X_1 + a_2X_2 + \dots + a_kX_k$ será a quantia aleatória total que será pago pelos itens. A soma das variáveis é uma composição linear, onde todos os a 's são 1.

Dessa forma, é necessário saber como calcular o valor esperado e a variância da soma ou a composição linear de diversas variáveis aleatórias. O valor esperado da soma de diversas variáveis aleatórias é a soma dos valores individuais esperados, conforme a expressão:

$$E(X_1 + X_2 + \dots + X_k) = E(X_1) + E(X_2) + \dots + E(X_k)$$

De forma análoga, o valor esperado de uma composição linear é dado por:

$$E(a_1X_1 + a_2X_2 + \dots + a_kX_k) = a_1E(X_1) + a_2E(X_2) + \dots + a_kE(X_k)$$

No caso de variância, considere apenas o caso em que $X_1, X_2, \dots, X_k$ são mutuamente independentes, pois caso não sejam, o cálculo envolve covariância.

Uma variável ser mutuamente independente significa que um evento $X_i = x$ e outro evento $X_j = y$ são independentes. Assim, é possível mencionar o resultado da seguinte forma:

Se $X_1, X_2, \dots, X_k$ são mutuamente independentes, então a variância da soma é a soma das variâncias individuais, pois a covariância é zero, ou seja: $V(X_1, X_2, \dots, X_k) = V(X_1) + V(X_2) + \dots + V(X_k)$, de forma análoga, a variância de uma composição linear é dada por: $V(a_1X_1 + a_2X_2 + \dots + a_kX_k) = a_1^2V(X_1) + a_2^2V(X_2) + \dots + a_k^2V(X_k)$ (Aczel, 2009).

Assumindo que a variação do atraso ocorre entre pacotes selecionados em que um evento e é independente de outros eventos que possam ocorrer, o cálculo da variação do atraso fim-a-fim de uma rota será realizado através da soma das variações individuais dos pacotes selecionados (variação do atraso total = $\sum$ variação dos atrasos individuais). Dessa forma, o sistema avaliará os enlaces que atendem ao requisitado no SLA requisitado. A Figura 56 ilustra uma rede abstrata contendo a variação de atraso média de cada enlace.

Figura 56 – Representação dos enlaces para o requisito variação do atraso

De acordo com a recomendação da ITU-T (2002), uma aplicação VoIP requer uma configuração mínima que de suporte a uma variação de atraso inferior a 20 ms. Como observado, dentro deste contexto, todos os enlaces são válidos, pois nenhum excede este valor. Ao realizar o cálculo para encontrar a variação de atraso fim-a-fim de uma rota, o sistema PBQoS encontrou a rota $A \rightarrow B \rightarrow E$ e a rota $A \rightarrow B \rightarrow C \rightarrow E$, com variação média de atraso de 1 ms e 5 ms respectivamente. Visto que o SLA da aplicação VoIP define uma variação de atraso inferior a 20 ms, as duas as rotas atendem esta configuração. Após este processo, o sistema aplica a política *minimizar variação do atraso* entre as rotas encontradas e a que apresentar a menor variação do atraso será selecionada. Dentro deste contexto,

por apresentar menor variação de atraso fim-a-fim, a rota $A \rightarrow B \rightarrow E$ será selecionada.

Para dar continuidade ao processo de cálculo de QoS, é preciso encontrar a perda fim-a-fim de uma rota. De acordo com Mieghem (2006), o número de falhas em uma rede de telecomunicações e/ou número de bits corrompidos depois de uma transmissão devido a erros no canal (dentre outros fenômenos) são considerados Processos Estocásticos.

Um Processo Estocástico é definido por um conjunto de variáveis aleatórias indexadas por um parâmetro $t \in \mathbf{R}$ (entendido como tempo).

$$X = \{ X(t_0), X(t_1), X(t_2), \dots, X(t_n) \}$$

A variável aleatória $X(t)$ é definida em um espaço denominado *espaço de estados*. Intuitivamente, se uma variável aleatória é um número real que varia aleatoriamente, um Processo Estocástico é uma função temporal que varia aleatoriamente. Qualquer tipo de evolução temporal (determinística ou essencialmente probabilística) que seja analisável em termos de probabilidade pode ser chamado de Processo Estocástico (Mieghem, 2006) (RFC 2680).

O *espaço de estado* de um Processo Estocástico é um conjunto de valores possíveis que uma variável aleatória $X(t)$ pode assumir. Cada valor possível é chamado de *estado* de um processo. Se o índice do conjunto T for enumerável ou finito $X(t)$ é um Processo Estocástico Discreto. Se o índice do conjunto T for contínuo (conjunto não enumerável ou infinito), $X(t)$ é um Processo Estocástico Contínuo. De acordo com [Mieghem-2006], o número de chegada dos pacotes em um roteador entre um intervalo de tempo $[a, b]$ é um Processo Estocástico Contínuo, pois $T \in [a, b]$.

O tempo contínuo de um Processo Estocástico $X(t)$ tem incrementos independentes se o valor assumido por $X(t_j)$ não depende do valor assumido por $X(t_i)$, se $i \neq j$, ou seja, se para todo $t_0 < t_1 < \dots < t_n$, as variáveis aleatórias $X(t_1) - X(t_0)$, $X(t_2) - X(t_1)$, ..., $X(t_n) - X(t_{n-1})$ são independentes (Mieghem, 2006).

Assumindo que uma rota é composta por diversos enlaces de forma linear, para realizar o cálculo das perdas de uma rota fim-a-fim, é preciso primeiramente encontrar a perda individual de cada enlace conforme ilustra a Figura 57.

Figura 57 – Cálculo da perda em um enlace

onde:

T = Taxa de transmissão no emissor

p_i = Taxa de perda no enlace

P_t = Taxa de perda total na rota

Quando uma rota for composta por mais de um enlace, é preciso encontrar a taxa de perda individual de cada enlace. Para isso, o sistema verifica a transmissão T em um em um enlace e com uma taxa de perda p . Através destas informações é possível calcular a perda total fim-a-fim da rota analisada conforme ilustra a Figura 58.

Figura 58 – Cálculo de perda para uma rota com diversos enlaces

Para dar continuidade ao processo de definição de uma rota para uma aplicação VoIP, a Figura 59 ilustra uma rede representada pelo grafo composto pelos enlaces e suas respectivas taxas de erro.

Figura 59 – Representação dos enlaces para o requisito perdas

De acordo com a definição de ITU-T (2002), uma aplicação VoIP define um SLA que suporta uma taxa de perda total inferior a 3%. Considerando uma transmissão T de 64 kbps, o sistema deve verificar as taxas de perdas das rotas

possíveis. A Tabela 14, ilustra o cálculo de perda fim-a-fim das rotas possíveis, (a) $A \rightarrow B \rightarrow E$ e (b) $A \rightarrow B \rightarrow C \rightarrow E$.

Tabela 14 – Cálculo da perda fim-a-fim para uma aplicação VoIP

ABE		AB		BE				(a)
T (kbps)	A	Perda AB	B	Perda BE	E	Vazão	Perda Total	
64	64	1%	63,36	1%	62,73	98,01%	1,99%	

ABCE		AB		BC		CE				(b)
T (kbps)	A	Perda AB	B	Perda BC	C	Perda CE	E	Vazão Total	Perda Total	
64	64	1%	63,36	2%	62,09	2%	60,85	95,08%	4,92%	

Como a política *minimizar perda* tem como meta encontrar a rota que apresenta à menor taxa de perda, após analisar os resultados, a rota $A \rightarrow B \rightarrow E$ será selecionada por apresentar uma taxa de perda menor (1,99%) que a rota $A \rightarrow B \rightarrow C \rightarrow E$ (4,9%), que além de ser maior, excedeu o limite imposto pelo SLA da aplicação. Dessa forma, a Figura 60 ilustra o contexto resultante das possíveis rotas encontradas pelo algoritmo.

Figura 60 – Grafos resultantes após a iteração do sistema

Após calcular o QoS total para cada rota encontrada na rede, o sistema fará uma análise em busca das rotas que atendem a especificação definida no SLA. Quando as rotas válidas (que atendem o SLA) são conhecidas, o sistema aplica a política de configuração ativada em busca da melhor rota disponível para atender a requisição da aplicação.

Como pode ser observado durante a execução do algoritmo, a cada iteração, as rotas que não se adaptam as restrições impostas pelo sistema, vão sendo automaticamente descartadas do processo de definição de rota.

Caso o contexto resultante apresente apenas uma rota válida, esta será considerada a melhor rota para atender o SLA requerido. Por outro lado, mais de uma rota válida pode ser encontrada, fazendo com que o sistema PBQoS execute a política ativada para que o processo de definição de rota seja efetivado. Quando apenas uma política estiver ativa, o sistema realiza a ordenação das rotas válidas com base nos critérios definidos pela política, que neste trabalho, pode ser maior vazão, menor atraso, menor perda e menor variação do atraso. Por outro lado, a composição de diversas políticas é permitida pelo sistema, onde o usuário pode ativar uma ou mais políticas simultaneamente. Dessa forma, o algoritmo realiza o cálculo do fator para que, dentre as diversas rotas válidas encontradas, o sistema possa determinar a melhor delas.

O cálculo do fator é realizado com base nas políticas ativas e segue as regras impostas pelas políticas de controle conforme ilustra a Tabela 15. Assim, dadas duas rotas válidas x e y , se $f(x)z \leq f(y)z$, então a rota x é melhor do que a rota y . O valor de z é -1 para o requisito de vazão e 1 caso contrário.

Tabela 15 – Critérios definidos pelas Políticas de Controle para o processo de definição do fator de otimização de uma rota

Requisitos	Operação (fator)
Vazão	$(Vazão_{SLA} / Vazão_{MIB})$
Atraso	$(Atraso_{MIB} / Atraso_{SLA})$
Jitter	$(Jitter_{MIB} / Jitter_{SLA})$
Perdas	$(Perdas_{MIB} / Perdas_{SLA})$

O cálculo deste fator inicia-se através do adimensionamento dos valores associados para cada requisito. Visto que o requisito vazão é o único que deve ser maximizado, o fator parcial para este requisito é obtido através da operação: *vazão SLA / vazão MIB (rota)*. Por outro lado, os requisitos atraso, variação do atraso e perda devem ser minimizados. E dessa forma, o fator parcial para cada requisito será obtido através das operações: *atraso MIB (rota) / atraso SLA* para o requisito atraso; *jitter MIB (rota) / jitter SLA* para o requisito jitter; *perda MIB (rota) / perda SLA* para o requisito perda. No momento em que todos os fatores parciais forem encontrados, o sistema encontra o produto dos fatores parciais ($fator = (parcial_vazão * parcial_atraso * parcial_jitter * parcial_perdas)$) que corresponde

ao fator encontrado para a rota. A rota que apresentar o menor fator será considerada a melhor rota. Para ilustrar este processo, considere a Tabela 16.

Tabela 16 – Cálculo do fator para rotas encontradas

Cálculo do Fator para rotas válidas encontradas					
Rotas	Vazão (kbps)	Atraso (ms)	Jitter (ms)	Perdas (%)	Fator (adimensional)
ABE	8000	28	1	1,99	0,0000186
ABCD	600	24	5	4,92	0,0026240

Como a rota A→B→E apresenta o menor fator, a mesma será escolhida para atender ao SLA requisitado. O diagrama de todo processo utilizado por este algoritmo pode ser visualizado através da Figura 61.

Figura 61 – Fluxograma do Algoritmo PBQoS

Inicialmente o sistema faz a leitura dos dados de entrada (origem, destino, SLA requerido e política(s) ativa(s)) obtidos através da interface gráfica do usuário e das informações atualizadas da rede através da MIB geral do sistema. Com base nestas informações, o sistema faz uma análise das informações dos enlaces obtidos através do algoritmo OSPF e carrega todas as rotas possíveis entre o nó de origem e destino. Havendo rotas válidas, o sistema as armazena em uma estrutura de dados e uma política de controle é ativada.

As políticas de controle determinam o comportamento funcional do sistema através dos critérios definidos pelo administrador da rede. Para que um serviço seja admitido, é necessário que a taxa de ocupação do enlace não exceda uma porcentagem λ de sua capacidade máxima disponível na rota. No contexto deste trabalho, λ foi definido em 85%, o que representa uma margem de segurança para que o sistema possa se manter operacional. Através deste mecanismo, o sistema realiza um processo de filtragem para eliminar as rotas cujos enlaces não

são válidos. Uma rota é considerada inválida quando algum enlace que a compõe possuir vazão insuficiente para atender o SLA de uma aplicação.

O critério definido pelas políticas de controle determina que, quando a vazão solicitada exceder λ * (vazão disponível em cada enlace) o sistema não admite a requisição, conforme ilustra a Tabela 17.

Tabela 17 – Controle do processo de admissão de um serviço

Política de controle – Admissão de uma requisição	
Para $\lambda = 85\%$	
Condição	SE (vazão da rota * λ) $\geq$ vazão do SLA
Ação	Sistema ADMITE a requisição!

Para ilustrar o controle de admissão de uma rota, considere um SLA que requer uma vazão mínima de 90 Mbps e a rota dispõe de uma capacidade total de 100 Mbps. O sistema realiza a operação matemática para determinar se o serviço pode ser admitido da seguinte forma: $(100 * 85\% (\lambda) \rightarrow 85 < 90)$. Como o valor solicitado excede λ , o sistema PBQoS não admite o serviço. Por outro lado, uma solicitação de 50 Mbps seria atendida pelo sistema $(100 * 85\% (\lambda) \rightarrow 85 > 50)$, pois o valor requerido não excede o valor de λ . Com base nestas informações, o sistema PBQoS utiliza o mecanismo interno de filtragem e armazena todas as rotas que atendem o SLA solicitado entre nós de origem e destino definidos na interface gráfica do usuário.

A próxima etapa realizada pelo sistema, é calcular o QoS fim-a-fim para cada rota válida encontrada. Com base no QoS calculado, o sistema realiza o cruzamento das informações de cada rota encontrada com as informações definidas no SLA. Apenas as rotas que atendem o SLA da aplicação serão selecionadas e armazenadas.

Para finalizar o processo, o sistema PBQoS conta com quatro políticas (*maximizar vazão, minimizar atraso, minimizar variação do atraso e minimizar perdas*) que podem ser ativadas individualmente ou simultaneamente. Quando apenas uma política for ativada, o sistema realiza a ordenação da melhor rota pela política selecionada. No caso da política *maximizar vazão*, a rota que apresentar maior vazão fim-a-fim será escolhida. Para as demais políticas a rota que apresentar o menor atraso, variação do atraso ou perdas, será escolhida.

Para que o sistema possa funcionar corretamente é preciso que pelo menos uma política seja ativada. Caso isso não ocorra, por padrão o sistema ativa a política *maximizar vazão*. Quando mais de uma política estiver ativa, o sistema inicia o processo para encontrar o fator para de rota válida. A rota que apresentar o menor fator será considerada a melhor rota.

Capítulo 8

Implementação Computacional do Sistema PBQoS

Como prova de conceito, a implementação computacional do sistema PBQoS faz uso do algoritmo de seleção de uma rota definido neste trabalho, utilizado para definir, manter e controlar os recursos de uma rota. Os resultados serão apresentados através de dois cenários no qual diversas iterações simultâneas são realizadas no sistema PBQoS. Dessa forma, é possível analisar o número de admissões e rejeições obtido pelo sistema para cada política ativa e a escalabilidade do algoritmo com relação ao número de nós e enlaces ativos em uma rede. As funcionalidades dos elementos arquiteturais poderão ser observadas através da execução da aplicação PBQoS.

8.1 – Cenário 1

O cenário 1 apresenta uma topologia composta por 5 nós parcialmente interconectados através de 5 enlaces. Inicialmente será ilustrado os mecanismos de Configuração, Manutenção e Controle de uma rota utilizado pelo sistema PBQoS. Na sequência será realizada uma simulação com diversas requisições sendo realizadas ao sistema de gerenciamento PBQoS.

8.1.1 – Processo de Configuração

O processo de configuração de uma rota é realizado através de mecanismos que permitem ao sistema encontrar a melhor rota entre dois pontos de uma rede. Para isto, leva em consideração os requisitos mínimos de QoS exigidos por uma aplicação e as regras impostas pelo sistema que são definidas pelas Políticas de Configuração.

Inicialmente o sistema deve conhecer a rede que será gerenciada, e para isto, todas as informações topológicas serão armazenadas na MIB no repositório TED. Considerando que o estado atual da rede é monitorado e atualizado

constantemente na MIB, para ilustrar um exemplo de uma MIB, considere a Tabela 18.

Tabela 18 – Exemplo da MIB geral da rede

Enlace	Origem	Destino	Vazão	Perdas	Jitter	Atraso
1	1	2	10000.0	2.0	1.0	4.0
2	1	3	15000.0	3.0	1.5	1.0
3	2	3	12000.0	2.0	2.0	2.0
4	3	5	18000.0	2.0	3.0	4.0
5	4	5	15000.0	1.0	2.5	8.0
6	2	4	20000.0	4.0	1.0	2.0

As informações contidas na MIB servirão de base para que o sistema de gerenciamento PBQoS possa verificar o estado de cada enlace e assim determinar a melhor rota entre dois nós. Para iniciar o processo de configuração de uma rota, o sistema PBQoS faz uma leitura periódica das MIBs locais de cada PEP, que por sua vez são atualizadas através do algoritmo OSPF. Após conhecer os nós e seus vizinhos, o sistema é capaz de pré-calculas todas as rotas possíveis existentes na rede. Um exemplo dos resultados gerados por este processo pode ser visualizado na Tabela 20.

Tabela 19 – Rotas encontradas pelo sistema PBQoS

1→2	1→3	1→3→5→4→2	3→5
1→2→3	1→3→2	2→3	3→5→4
1→2→4	1→3→2→4	2→4	3→5→4→2
1→2→3→5	1→3→5	2→3→5	4→5
1→2→3→5→4	1→3→5→4	2→3→5→4	5→4→2

Para facilitar a leitura das informações de cada enlace presente na rede (Tabela 18) e as conexões entre os nós (Tabela 19), podem ser visualizados em forma de grafo através da

Figura 62.

Figura 62 – Representação da rede em forma de Grafo

Após encontrar todas as rotas possíveis na rede, o sistema PBQoS ativa um mecanismo que permite calcular o QoS fim-a-fim para cada rota disponível na rede conforme ilustra a Tabela 20.

Tabela 20 – Cálculo do QoS para cada rota encontrada na rede

Enlace	Rota	Vazão	Perda	Jitter	Atraso
1	1→2	10	2,00%	1	4
2	1→2→3	10	3,96%	3	3
3	1→2→4	10	5,92%	2	3
4	1→2→3→5	10	5,88%	6	7
5	1→2→3→5→4	10	6,82%	8,5	15
6	1→3	15	3,00%	1,5	1
7	1→3→2	12	4,94%	3,5	3
8	1→3→2→4	12	8,74%	4,5	5
9	1→3→5	15	4,94%	4,5	5
10	1→3→5→4	15	5,89%	7	13
11	1→3→5→4→2	15	9,65%	8	15
12	2→3	12	2,00%	2	2
13	2→4	20	4,00%	1	2
14	2→3→5	12	3,96%	5	6
15	2→3→5→4	15	4,92%	7,5	14
16	3→5	18	2,00%	3	4
17	3→5→4	15	2,98%	5,5	12
18	3→5→4→2	15	3,96%	6,5	14
19	4→5	15	1,00%	2,5	8
20	5→4→2	15	4,96%	3,5	10

Como prova de conceito, foi desenvolvida uma aplicação que permite realizar a inserção das informações necessárias para que uma requisição seja realizada. Dentre estas informações, estão: nó de origem e destino, SLA e política(s) ativa(s). Com base na MIB definida pela Tabela 18, a execução da aplicação PBQoS pode ser visualizada na Figura 63.

Figura 63 – Aplicação PBQoS

Através da aplicação PBQoS também é possível visualizar a vazão nominal de cada enlace no momento monitorado. Inicialmente todos os enlaces dispõem de 100% de seus recursos disponíveis e a cada iteração, esses recursos vão sendo consumidos e o gráfico por sua vez é atualizado.

Conforme já definido anteriormente, para que o processo de configuração seja realizado, é preciso que o usuário entre com as informações necessárias para que o sistema possa determinar a melhor rota que irá atender a requisição. Para ilustrar este mecanismo, considere a configuração requisitada ilustrada pela Tabela 21.

Tabela 21 – Configuração requisitada ao sistema PBQoS

Nós		SLA				Configuração
Origem	Destino	Vazão (Kbps)	Perda (%)	Jitter (ms)	Atraso (ms)	Política
1	2	7000	10	10	20	Maximizar vazão

Quando uma requisição é realizada, o sistema PBQoS faz uma busca por todas as rotas possíveis entre os nós de origem e destino e realiza uma análise do QoS fim-a-fim calculado em cada rota.

Quando ativa, uma política determina as regras de operabilidade do sistema. Com base na requisição ilustrada pela Tabela 21, apenas a Política *Maximizar a Vazão* foi ativada, fazendo com que o sistema busque pela rota que apresentar a maior vazão fim-a-fim.

Conforme ilustra a Tabela 22, três rotas válidas entre os nós 1 e 2 foram encontradas (1→2, 1→2→3 e 1→3→5→4→2). Como a política ativa busca pela rota que apresentar a maior vazão, a rota 1→3→5→4→2 será selecionada e a requisição será admitida nesta rota.

Tabela 22 – Política *Maximizar Vazão* – Rotas encontradas entre os nós 1 e 2

Rota	Rotas encontradas	Vazão
1→2	1→2	10
	1→3→2	12
	1→3→5→4→2	15

Como pode ser visualizada na Figura 64, a aplicação PBQoS selecionou a rota 11 (1→3→5→4→2), pois, além de apresentar maior vazão, esta rota dispõe de recursos suficientes para atender o SLA requisitado.

Figura 64 – Política *Maximizar Vazão* – Requisição de rota ao sistema PBQoS

O sistema PBQoS dispõe de um mecanismo que registra em um sistema de *log* as principais informações e resultados de uma requisição. Com base na configuração definida pela Tabela 21, é possível visualizar o *log* gerado pelo sistema PBQoS através da Figura 65.

análise executada dia 20/ 9/2010 hora: 16: 7:23 hs									
origem:		destino:							
1		2							
sla:	vazao	perda	jitter	atraso					
	7000.0	10.0	10.0	20.0					
politica:	vazao	perda	jitter	atraso					
	1	0	0	0					
rota	vazao	perda	jitter	atraso	enlaces da rota				
11	15000.00	9.7	8.00	15.00	1 3 5 4 2				
matriz inuse									
enlace	org	dest	vazão						
1	1	2	0.0						
2	1	3	7000.0						
3	2	3	0.0						
4	3	5	7000.0						
5	4	5	7000.0						
6	2	4	7000.0						

Figura 65 – *log* gerado pelo sistema – Política *Maximizar Vazão*

Analisando os resultados gerados, é possível visualizar que a requisição foi atendida pela rota 11 (1→3→5→4→2), no qual a vazão solicitada (7000 kbps) foi reservada nos enlaces 1→3, 3→5, 5→4, 4→2. No momento em que estes recursos forem liberados pela rota, os mesmo se tornam disponíveis na MIB geral da rede.

De forma similar, este mesmo processo pode ser realizado para outras políticas ativas. Como já mencionado anteriormente, a Política *Minimizar Perda* tem como critério básico encontrar a rota que apresentar a menor taxa de perda fim-a-fim de uma rota. Dessa forma, ao ativar esta política, o sistema PBQoS busca pelas rotas válidas capaz de atender o SLA requisitado. Com base na requisição definida pela Tabela 21, as rotas válidas encontradas podem ser visualizadas na Tabela 23.

Tabela 23 – Política *Minimizar Perdas* – Rotas encontradas entre os nós 1 e 2

Rota	Rotas encontradas	Perdas
1→2	1→2	2
	1→3→2	4,94
	1→3→5→4→2	9,65

Conforme ilustra a Tabela 23, três rotas válidas entre os nós 1 e 2 foram encontradas (1→2, 1→2→3 e 1→3→5→4→2). Como a política ativa busca pela

rota que apresentar a menor taxa de perdas, a rota 1→2 será selecionada e a requisição será admitida nesta rota. Como pode ser visualizada na Figura 66, a aplicação PBQoS selecionou a rota 1 (1→2), pois, além de apresentar menor taxa de perda, esta rota dispõe de recursos suficientes para atender a requisição.

Figura 66 – Política *Minimizar Perdas* – Requisição de rota ao sistema PBQoS

O log gerado pelo sistema PBQoS pode ser visualizado na Figura 67.

```

análise executada dia 20/ 9/2010  hora: 16:12:30 hs
origem:          destino:
  1              2
sla:      vazao      perda      jitter      atraso
          7000.0      10.0      10.0      20.0
politica: vazao      perda      jitter      atraso
          0          1          0          0
rota      vazao      perda      jitter      atraso      enlaces da rota
  1      10000.00      2.0      1.00      4.00      1      2

matriz inuse
enlace org dest      vazão
  1      1      2      7000.0
  2      1      3          0.0
  3      2      3          0.0
  4      3      5          0.0
  5      4      5          0.0
  6      2      4          0.0
  
```

Figura 67 – Política *Minimizar Perda* – log gerado pelo sistema

Como a rota 1 (1→2) é composta por apenas dois nós e um enlace, ocorre a reserva de recursos neste enlace.

A Política *Minimizar Variação do atraso* tem como regra básica encontrar a rota que apresentar a menor variação do atraso fim-a-fim de uma rota. Com base na requisição definida pela Tabela 21, as rotas válidas encontradas podem ser visualizadas na Tabela 24.

Tabela 24 – Política *Minimizar Variação do atraso* – Rotas encontradas entre os nós 1 e 2

Rota	Rotas encontradas	Jitter
1→2	1→2	1
	1→3→2	3,5
	1→3→5→4→2	8

Das três rotas válidas encontradas entre os nós 1 e 2 (1→2, 1→2→3 e 1→3→5→4→2), a rota 1→2 por apresentar a menor variação do atraso, será selecionada pelo sistema PBQoS e a requisição será admitida nesta rota.

Como pode ser visualizado na Figura 68, a aplicação PBQoS selecionou a rota 1 (1→2), pois, além de apresentar a menor variação do atraso, esta rota dispõe de recursos suficientes para atender a requisição.

Figura 68 – Política *Minimizar variação do atraso* – Requisição de uma rota ao sistema PBQoS

Os detalhes desta admissão podem ser visualizados através do *log* gerado pelo sistema através da Figura 69.

análise executada dia 20/ 9/2010 hora: 16:24:52 hs											
origem:			destino:								
1			2								
sla:		vazao		perda		jitter		atraso			
		7000.0		10.0		10.0		20.0			
politica:		vazao		perda		jitter		atraso			
		0		0		1		0			
rota		vazao		perda		jitter		atraso		enlaces da rota	
1		10000.00		2.0		1.00		4.00		1 2	
matriz inuse											
enlace		org		dest		vazão					
1		1		2		7000.0					
2		1		3		0.0					
3		2		3		0.0					
4		3		5		0.0					
5		4		5		0.0					
6		2		4		0.0					

Figura 69 – log gerado pelo sistema – Política *Minimizar Perda*

Como a rota 1 (1→2) é composta por apenas dois nós e um enlace, ocorre a reserva de recursos neste enlace (1→2 = 7000 kbps).

A Política *Minimizar Atraso* busca pela rota que apresentar o menor atraso fim-a-fim. Com base na requisição definida pela Tabela 21, as rotas válidas encontradas podem ser visualizadas na Tabela 25.

Tabela 25 – Política *Minimizar Atraso* – Rotas encontradas entre os nós 1 e 2

Rota	Rotas encontradas	Atraso
1→2	1→2	4
	1→3→2	3
	1→3→5→4→2	15

Das três rotas válidas encontradas entre os nós 1 e 2 (1→2, 1→2→3 e 1→3→5→4→2), a rota 1→3→2 por apresentar a menor atraso, será selecionada pelo sistema PBQoS e a requisição será admitida nesta rota.

Como pode ser visualizado na Figura 70, a aplicação PBQoS selecionou a rota 1 (1→2), pois, além de apresentar a menor variação do atraso, esta rota dispõe de recursos suficientes para atender a requisição.

Figura 70 – Requisição de rota ao sistema PBQoS – Política *Minimizar Atraso*

O log gerado pelo sistema PBQoS pode ser observado da Figura 68.

```

análise executada dia 20/ 9/2010  hora: 16:22:57 hs
origem:          destino:
  1              2
sla:      vazao      perda      jitter      atraso
          7000.0     10.0       10.0       20.0
politica: vazao      perda      jitter      atraso
          0          0         0         1
rota  vazao      perda      jitter      atraso      enlaces da rota
  7    12000.00    4.9       3.50       3.00         1    3    2

matriz inuse
enlace org dest      vazão
  1     1    2         0.0
  2     1    3       7000.0
  3     2    3       7000.0
  4     3    5         0.0
  5     4    5         0.0
  6     2    4         0.0

```

Figura 71 – log gerado pelo sistema – Política *Minimizar Atraso*

Como o sistema encontrou a rota 7 (1→3→2) e a mesma é composta por três nós e dois enlaces, ocorre a reserva de recursos nesses enlaces (1→3 e 2→3 = 7000 kbps).

A Política *Visão Global de QoS* define regras para que o sistema PBQoS possa encontrar a melhor rota entre dois nós, com base em todas as políticas (*Maximizar Vazão e Minimizar Perdas, Atraso e Variação do Atraso*) ativas. Porém, como já definido anteriormente, quando mais de uma política for ativada

simultaneamente, o sistema muda seu modo de operação e ativa o mecanismo para encontrar o fator de cada rota válida. Através de uma hierarquia de parâmetros, a rota que apresentar o menor fator será considerada a melhor rota.

Com base na mesma requisição definida na Tabela 21, ao ativar todas as Políticas e realizar uma nova requisição ao sistema PBQoS, as rotas válidas encontradas e os requisitos de QoS calculados para cada rota podem ser visualizados na Tabela 26.

Tabela 26 – Política *Visão Global de QoS* – Rotas encontradas entre os nós 1 e 2

Rota	Rotas encontradas	Vazão	Perdas	Jitter	Atraso	Fator
1→2	1→2	10	2%	1	4	0,028
	1→3→2	12	4,94%	3,5	3	0,151
	1→3→5→4→2	15	9,65%	8	15	2,703

Por apresentar o menor fator, a rota 1→2 será considerada a melhor rota e ao ser selecionada pelo sistema PBQoS, a requisição será admitida nesta rota. Como pode ser visualizado na Figura 72, a aplicação PBQoS selecionou a rota 1 (1→2), pois, além de apresentar a menor variação do atraso, esta rota dispõe de recursos suficientes para atender a requisição.

Figura 72 – Política *Visão Global de QoS* – Requisição de rota ao sistema PBQoS

O log gerado pelo sistema PBQoS pode ser visualizado através da Figura 73.

```

análise executada dia 20/ 9/2010  hora: 16:28:30 hs
origem:                destino:
    1                    2
sla:          vazao      perda      jitter   atraso
          7000.0        10.0        10.0      20.0
politica:    vazao      perda      jitter   atraso
          1            1          1          0
rota  vazao      perda      jitter      atraso  enlaces da rota
    1  10000.00    2.0        1.00        4.00        1      2

matriz inuse
enlace org dest      vazão
  1      1    2      7000.0
  2      1    3         0.0
  3      2    3         0.0
  4      3    5         0.0
  5      4    5         0.0
  6      2    4         0.0

```

Figura 73– log gerado pelo sistema – Política Visão Global de QoS

Como a rota 1 (1→2) é composta por apenas dois nós e um enlace, ocorre a reserva de recursos neste enlace (1→2 = 7000 kbps).

8.2 – Processo de Manutenção e Controle de uma rota

O processo de manutenção é responsável pela seleção de novas rotas quando a capacidade nominal de um enlace que compõe a rota admitida atingir os limites de segurança impostos pelas políticas de controle. Como prova de conceito, as políticas de controle foram definidas através de três funcionalidades:

- **Alerta amarelo:** O sistema monitora cada enlace definido na rede visando controlar o envio de alertas. Quando a utilização de um enlace exceder 70% de sua capacidade, o sistema envia um alerta “amarelo” ao administrador da rede.
- **Alerta vermelho:** Quando a utilização de um enlace exceder 85% de sua capacidade, o sistema envia um alerta “vermelho” ao administrador da rede. Quando um alerta “vermelho” é emitido, o sistema ativa as políticas de configuração em busca de uma nova rota (menos congestionada) para atender o SLA requisitado.
- **Controle da admissão:** Uma requisição só é admitida se e somente se, os recursos solicitados pelo SLA não exceder 85% da capacidade total disponível na rota.

Para demonstrar este mecanismo em operação, considere a configuração definida na Tabela 27.

Tabela 27 – SLA requerido – Processo de manutenção de rota

Nós		SLA				Configuração
Origem	Destino	Vazão (Kbps)	Perda (%)	Jitter (ms)	Atraso (ms)	Política
1	2	8500	20	10	40	Visão Global de QoS

Ao executar a configuração definida na Tabela 27, é possível observar o sistema PBQoS buscando por rotas alternativas quando algum enlace da rota atingir o limite mínimo de segurança (85% da capacidade disponível na rota). A Figura 74 ilustra o início deste processo.

Figura 74 – Política Visão Global de QoS – Processo de Manutenção de rota

Como pode ser observado, sistema PBQoS admitiu a requisição com base do SLA requisitado, fornecendo assim os recursos solicitados pelo SLA. Conforme ilustra a Figura 74, a rota 1 dispõe de recursos suficientes (vazão: 10000.00 kbps, jitter: 1.0 ms, perda: 2.00% e atraso: 4.00 ms) para atender a requisição feita pelo SLA. Ao admitir a requisição, a vazão nominal do enlace 1→2 atingiu 85% dos recursos totais disponíveis, fazendo com que um alerta “vermelho” seja enviado ao administrador da rede. Os detalhes desta admissão podem ser observados no log gerado pelo sistema através da Figura 75.

```

análise executada dia 20/ 9/2010  hora: 17:53:50 hs

origem:          destino:
  1              2
sla:      vazao      perda      jitter      atraso
          8500.0      20.0      10.0      40.0
politica: vazao      perda      jitter      atraso
          1          1          1          1
rota  vazao      perda      jitter      atraso      enlaces da rota
  1    10000.00      2.0      1.00      4.00          1      2

matriz inuse
enlace org dest      vazão
  1      1      2      8500.0
  2      1      3        0.0
  3      2      3        0.0
  4      3      5        0.0
  5      4      5        0.0
  6      2      4        0.0

```

Figura 75 – log do sistema PBQoS – Políticas de Manutenção

Ao receber um alerta “vermelho”, o sistema PBQoS muda seu modo de operação e ativa as políticas de manutenção em busca de uma rota alternativa capaz de atender o SLA requisitado. Como ilustra a Figura 76, ao iniciar este processo, o sistema encontrou a rota 7 (1→3→2) e por dispor de recursos suficientes para atender a requisição, esta rota é selecionada.

Figura 76 – Política Visão Global de QoS – Processo de Manutenção de uma rota

Os detalhes desta admissão podem ser observados no log gerado pelo sistema através da Figura 77.

```

análise executada dia 20/ 9/2010  hora: 18:11:52 hs
origem:          destino:
  1              2
sla:      vazao      perda      jitter      atraso
          8500.0      20.0      10.0      40.0
politica: vazao      perda      jitter      atraso
          1          1          1          1
rota  vazao      perda      jitter      atraso      enlaces da rota
  7    14000.00    4.9      5.50      2.00          1    3    2

matriz inuse
enlace org dest      vazão
  1     1   2      8500.0
  2     1   3      8500.0
  3     2   3      8500.0
  4     3   5         0.0
  5     4   5         0.0
  6     2   4         0.0

```

Figura 77 – log gerado pelo sistema PBQoS – Processo de manutenção

Como o enlace entre os nós 1 e 2 (1→2) atingiu 85% do total de recursos disponíveis na rota, o sistema reativará o processo de configuração em busca de uma nova rota capaz de atender o SLA requisitado. Vale ressaltar que no momento em que os recursos utilizados por uma rota vão sendo liberados, os mesmos são devolvidos ao sistema (MIB geral da rede), que por sua vez, poderá ser utilizados por novas requisições.

Ao iniciar o processo de configuração em busca de um rota alternativa, o sistema PBQoS encontrou a rota 11 (1→3→5→4→2). Por dispor de recursos suficientes (vazão: 13600.00 kbps, *jitter*: 8.0 ms, perda: 9.65% e atraso: 15.00 ms) a requisição será admitida nesta rota, conforme ilustra a Figura 78.

Figura 78 – Política Visão Global de QoS – Processo de Manutenção de uma rota

Ao admitir a requisição, a vazão nominal no enlace 1→3 atingiu 70% do total disponível, fazendo com que um alerta “amarelo” seja emitido ao administrador da rede. O *log* gerado pelo sistema PBQoS pode ser visualizado através da Figura 79.

```

análise executada dia 20/ 9/2010  hora: 18:15:41 hs
origem:          destino:
  1              2
sla:      vazao   perda   jitter  atraso
          8500.0   20.0    10.0    40.0
politica: vazao   perda   jitter  atraso
          1       1       1       1
rota  vazao   perda   jitter  atraso  enlaces da rota
11    14900.00  9.7     8.00    15.00  1    3    5    4    2

matriz inuse
Enlace org dest      vazão
  1     1     2      8500.0
  2     1     3     17000.0
  3     2     3      8500.0
  4     3     5      8500.0
  5     4     5      8500.0
  6     2     4      8500.0
  
```

Figura 79 – log do sistema PBQoS – Política de manutenção

Como diversos enlaces foram envolvidos na composição da rota 11 (1→3→5→4→2), os recursos são reservados em cada enlace.

8.1.3 – Análise dos resultados gerados pelo sistema PBQoS

Os resultados gerados pelo sistema PBQoS serão apresentados de duas formas: requisições simultâneas com diferentes SLA's e requisições simultâneas com base em um único SLA. Todos os resultados serão guiados pelas políticas de configuração ativas. Para gerar carga na rede, diversas aplicações devem fazer requisições simultaneamente. A Tabela 28 ilustra as aplicações selecionadas e seus respectivos perfis que servirão de base para compor os SLA's utilizados pelas requisições neste cenário.

Tabela 28 – Conjunto de perfis de Aplicações

Vídeo				
Tipo de Aplicação	Vazão	Atraso	Jitter	Perdas
HDTV descomprimido	1.5 Gbps	150-400 ms	<30 ms	1-3%
SDTV, SMPTE	270 Mbps	150-400 ms	<30 ms	1-3%
MPEG-2 comprimido 4:2:2	25-60 Mbps	150-400 ms	<30 ms	1-3%
HDTV (1280x720)	81 Mbps	150-400 ms	<30 ms	1-3%
Áudio				
Tipo de Aplicação	Vazão	Atraso	Jitter	Perdas
Voz convencional	64 Kbps	150-400 ms	<20ms	1-3%
CD	1.4 Mbps	150-400 ms	<20ms	1-3%
MPEG	384 Kbps	150-400 ms	<20ms	1-3%
VoIP (PCM)	64 Kbps	100-150 ms	<40ms	1-2%
VoIP (Nokia)	12-24 Kbps	100-200 ms	40-75 ms	1-2%
VoIP (MP-MLQ/ACELP)	5-7 Kbps	100-200 ms	40-75 ms	1-2%
Dados				
Tipo de Aplicação	Vazão	Atraso	Jitter	Perdas
Web Browsing	10 Kbps	<2 seg	<200 ms	0%
Jogos interativos	100 Kbps	200 ms	<50 ms	0%
Transferência de arquivos	10 Mbps	60 seg	<300 ms	0%
Telnet	1 Kbps	<200 ms	<100 ms	0%

Assumindo que a topologia da rede seja composta pelas informações definidas na Tabela 29 (MIB geral da rede), como amostragem, foram realizadas dez requisições com diferentes configurações, e os resultados serão obtidos com base nas políticas ativas. Dessa forma, será possível analisar os enlaces que serão utilizados na composição de uma rota, disponibilidade de recursos da rota, número de aplicações admitidas e rejeitadas pelo sistema PBQoS.

Tabela 29 – MIB geral da rede

Enlace	origem	destino	vazão	perda	jitter	perda
1	1	2	10000.0	2.0	1.0	4.0
2	1	3	18000.0	3.0	1.5	1.0
3	2	3	14000.0	2.0	4.0	1.0
4	3	5	18000.0	2.0	3.0	4.0

5	4	5	15000.0	1.0	2.5	8.0
6	2	4	20000.0	4.0	1.0	2.0

A Política *Maximizar Vazão* define regras e critérios para que o sistema PBQoS possa selecionar a rota que apresentar a maior vazão fim-a-fim dentre as rotas válidas encontradas. Para ilustrar os resultados gerados pelo processo de configuração com base na política *Maximizar Vazão* ativa considere os dados da Tabela 30.

Tabela 30 – Requisição de recursos – Política *Maximizar Vazão*

Requisição	Nós		SLA				Configuração
	Origem	Destino	Vazão (Kbps)	Perda (%)	Jitter (ms)	Atraso (ms)	Rota selecionada
1	1	4	5000	15	10	100	1→3→5→4
2	2	5	7000	10	15	300	2→3→5
3	5	1	3500	8	8	50	5→3→1
4	4	1	6000	10	10	30	4→2→1
5	3	4	600	30	30	100	3→5→4
6	1	2	3000	10	10	50	1→3→2
7	5	3	400	20	20	80	5→3
8	2	5	1500	10	10	20	2→4→5
9	4	3	1000	10	10	150	4→5→3
10	1	4	64	30	30	100	1→2→4

Na requisição 1, por exemplo, o sistema PBQoS admitiu o SLA requerido (vazão = 5000 kbps, perda = 15%, *jitter* = 10 ms e atraso = 100 ms) na rota 1→3→5→4. De forma similar, as demais requisições foram admitidas em rotas que dispunham de recursos para atender o SLA requisitado.

A Política *Minimizar Atraso* define regras e critérios para que o sistema PBQoS possa determinar a rota que apresentar o menor atraso fim-a-fim dentre as rotas válidas encontradas. De forma análoga ao processo descrito pela política *Maximizar Vazão*, foram realizadas 10 requisições simultâneas visando demonstrar o número de aplicações admitidas e rejeitadas pelo sistema PBQoS. Para ilustrar os resultados gerados considere os dados da Tabela 31.

Tabela 31 – Requisição de recursos – Política *Minimizar Atraso*

Requisição	Nós		SLA				Configuração
	Origem	Destino	Vazão (Kbps)	Perda (%)	Jitter (ms)	Atraso (ms)	Rota selecionada
1	1	4	5000	15	10	100	1→3→2→4
2	2	5	7000	10	15	300	2→3→5
3	5	1	3500	8	8	50	5→3→1
4	4	1	6000	10	10	30	4→2→1
5	3	4	600	30	30	100	3→5→4
6	1	2	3000	10	10	50	1→2
7	5	3	400	20	20	80	5→3
8	2	5	1500	10	10	20	2→3→5
9	4	3	1000	10	10	150	4→5→3
10	1	4	64	30	30	100	1→2→4

Com base nas configurações definidas pelos SLA's, é possível observar que o sistema PBQoS admitiu as requisições em rotas que dispunham de recursos para atendê-los.

A Política *Minimizar Variação do Atraso* define regras e critérios para que o sistema PBQoS possa determinar a rota que apresentar a menor variação do atraso fim-a-fim dentre as rotas válidas encontradas. Para ilustrar os resultados gerados pelo processo de configuração com base na política minimizar variação do atraso considere os dados da

Tabela 32.

Tabela 32 – Requisição de recursos – Política *Minimizar Variação do Atraso*

Requisição	Nós		SLA				Configuração
	Origem	Destino	Vazão (Kbps)	Perda (%)	Jitter (ms)	Atraso (ms)	Rota selecionada
1	1	4	5000	15	10	100	1→2→4
2	2	5	7000	10	15	300	5→4→2
3	5	1	3500	8	8	50	5→3→1
4	4	1	6000	10	10	30	4→2→3→1
5	3	4	600	30	30	100	3→5→4
6	1	2	3000	10	10	50	1→2
7	5	3	400	20	20	80	5→3
8	2	5	1500	10	10	20	2→4→5
9	4	3	1000	10	10	150	4→5→3
10	1	4	64	30	30	100	1→2→4

Na requisição 1, por exemplo, o sistema PBQoS admitiu a requisição através da rota 1→2→4, enquanto que, a requisição 2 foi admitida na rota 5→4→2. As demais requisições foram admitidas através de rotas que dispunham de recursos para atender o SLA requisitado.

A Política *Minimizar Perdas* define regras e critérios para que o sistema PBQoS possa determinar a rota que apresentar a menor perda fim-a-fim dentre as rotas válidas encontradas. Para ilustrar os resultados gerados pelo processo de configuração com base na política *Minimizar Perdas* considere os dados da Tabela 33.

Tabela 33 – Requisição de recursos – Política *Minimizar Perdas*

Requisição	Nós		SLA				Configuração
	Origem	Destino	Vazão (Kbps)	Perda (%)	Jitter (ms)	Atraso (ms)	Rota selecionada
1	1	4	5000	15	10	100	1→3→5→4
2	2	5	7000	10	15	300	2→3→5
3	5	1	3500	8	8	50	5→3→1
4	4	1	6000	10	10	30	4→2→1
5	3	4	600	30	30	100	3→5→4
6	1	2	3000	10	10	50	1→2
7	5	3	400	20	20	80	5→3
8	2	5	1500	10	10	20	2→4→5
9	4	3	1000	10	10	150	4→5→3
10	1	4	64	30	30	100	1→3→5→4

Por dispor de recursos disponíveis na rede, o sistema PBQoS foi capaz de atender cada requisição através das rotas calculadas que dispunham de recursos para atender o SLA requisitado. Com base nos resultados apresentados pela Tabela 33, por exemplo, a requisição 9 entre os nós 4 e 3 foi admitida na rota 4→5→3, enquanto que a requisição 10 entre os nós 1 e 4 foi admitida na rota 1→3→5→4. As demais requisições foram admitidas nas rotas calculadas que dispunham de recursos suficientes para atender cada SLA.

Por fim, a política *Visão Global de QoS* define regras e critérios para que o sistema PBQoS possa determinar a melhor rota dentre as rotas válidas encontradas com base em todas as políticas ativas simultaneamente. Com base neste cenário, o sistema PBQoS define uma hierarquia de parâmetros, para que seja possível determinar a melhor rota dentre as válidas encontradas. O contexto resultante desta operação é encontrar a rota que apresentar a maior vazão, o menor atraso, menor variação do atraso e a menor taxa de perdas. Para ilustrar os resultados gerados pelo processo de configuração com base na política *Visão Global de QoS* considere os dados da Tabela 34.

Tabela 34 – Requisição de recursos – Política *Visão Global de QoS*

Enlace	Nós		SLA				Configuração
	Origem	Destino	Vazão (Kbps)	Perda (%)	Jitter (ms)	Atraso (ms)	Rota selecionada
1	1	4	5000	15	10	100	1→2→4
2	2	5	7000	10	15	300	2→3→5
3	5	1	3500	8	8	50	5→3→1
4	4	1	6000	10	10	30	4→5→3→1
5	3	4	600	30	30	100	3→5→4
6	1	2	3000	10	10	50	1→2
7	5	3	400	20	20	80	5→3
8	2	5	1500	10	10	20	2→4→5
9	4	3	1000	10	10	150	-
10	1	4	64	30	30	100	1→2→4

Com base em cada SLA definido, o sistema foi capaz de admitir as requisições nas rotas que dispunham de recursos para atender o SLA requisitado. Conforme ilustrou a Tabela 34, por não dispor de recursos disponíveis, a requisição 9 não pode ser atendida pelo sistema PBQoS.

A Tabela 35 ilustra um comparativo dos resultados gerados para cada política ativa. Dessa forma, é possível observar que para cada regra imposta pelas políticas de configuração, manutenção e controle, diferentes rotas poderão ser selecionadas para atender uma requisição. Mostrando assim a eficiência do

sistema PBQoS no que tange o cálculo e seleção das melhores rotas para atender os SLA's requisitados.

Tabela 35 – Comparativo das rotas encontradas pelo sistema PBQoS

Nós			SLA				Rotas				
Enlace	Origem	Destino	Vazão (kbps)	Perdas (%)	Jitter (ms)	Atraso (ms)	Max. Vazão	Min. Perda	Min. Jitter	Min. Atraso	Visão Global QoS
1	1	4	5000	15	10	100	1→3→5→4	1→3→5→4	1→2→4	1→3→2→4	1→2→4
2	2	5	7000	10	15	300	2→3→5	2→3→5	5→4→2	2→3→5	2→3→5
3	5	1	3500	8	8	50	5→3→1	5→3→1	5→3→1	5→3→1	5→3→1
4	4	1	6000	10	10	30	4→2→1	4→2→1	4→2→3→1	4→2→1	4→5→3→1
5	3	4	600	5	30	100	3→5→4	3→5→4	3→5→4	3→5→4	3→5→4
6	1	2	3000	8	10	50	1→3→2	1→2	1→2	1→2	1→2
7	5	3	400	5	20	80	5→3	5→3	5→3	5→3	5→3
8	2	5	1500	10	10	20	2→4→5	2→4→5	2→4→5	2→3→5	2→4→5
9	4	3	1000	10	10	150	4→5→3	4→5→3	4→5→3	4→5→3	-
10	1	4	64	8	30	100	1→2→4	1→3→5→4	1→2→4	1→2→4	1→2→4

Com base nos resultados gerados, foi possível observar que para as políticas que controlam os requisitos de QoS individualmente, o número de admissões foi equivalente, ou seja, das dez requisições realizadas, por haver recursos na rede, todas foram admitidas pelo sistema PBQoS. Por outro lado, quando houver a ativação de duas ou mais políticas, os critérios serão mais rígidos, fazendo com que o sistema rejeite um número maior de requisições. Dentro deste contexto, um resumo dos resultados gerados pelo sistema PBQoS pode ser visualizado na Tabela 36.

Tabela 36 – Resumo dos resultados gerados pelas políticas ativas

	Requisição	Admissão	Rejeição	%admitido
Maximizar vazão	10	10	0	100
Minimizar perda	10	10	0	100
Minimizar jitter	10	10	0	100
Minimizar atraso	10	10	0	100
Visão Global de QoS	10	9	1	90

A segunda simulação feita com o sistema PBQoS, teve o intuito de realizar diversas requisições simultaneamente, com base no SLA definido na Tabela 37. Dessa forma, foi possível visualizar o processo de configuração, manutenção e controle em operação, além do número de requisições que o sistema conseguiu admitir.

Tabela 37 – SLA utilizado para gerar carga na rede

Nós		SLA			
Origem	Destino	Vazão (Kbps)	Perda (%)	Jitter (ms)	Atraso (ms)
1	4	3000	10	8	100

Para cada política ativa, diferentes rotas foram selecionadas pelo sistema PBQoS e os resultados gerados por cada requisição podem ser visualizados através da Tabela 38.

Tabela 38 – Rotas selecionadas para cada requisição

Requisições	Rotas selecionadas				
	Max. Vazão	Min. Perda	Min. Jitter	Min. Atraso	Visão Global
1	1→3→5→4	1→3→5→4	1→2→4	1→3→2→4	1→2→4
2	1→3→2→4	1→3→5→4	1→2→4	1→3→2→4	1→2→4
3	1→3→5→4	1→3→5→4	1→2→4	1→3→2→4	1→3→2→4
4	1→2→4	1→3→5→4	1→3→2→4	1→3→2→4	1→2→4
5	1→3→2→4	1→2→4	1→3→2→4	1→3→2→4	1→3→2→4
6	1→2→4	1→2→4	1→3→2→4	1→2→4	1→3→5→4
7	1→3→2→4	1→2→4	1→3→5→4	1→2→4	1→3→2→4
8	1→2→4	1→3→2→4	1→3→5→4	1→3→5→4	1→3→5→4

Os resultados foram obtidos através da execução da aplicação PBQoS sobre a arquitetura proposta neste trabalho. Com base no SLA requerido e nas regras de configuração definidas pela(s) política(s) ativa(s), diversas requisições foram realizadas simultaneamente. Com base nos resultados apresentados pela Tabela 38, é possível observar que o sistema busca por rotas que possuam recursos disponíveis para atender o SLA requisitado.

Visto que os recursos disponíveis na rede são limitados e em cada requisição admitida estes recursos vão sendo consumidos, neste cenário, o sistema PBQoS foi capaz de atender 8 requisições simultâneas até que os recursos da rede fossem esgotados.

8.2 – Cenário 2

O cenário 2 apresenta uma topologia composta por 6 nós totalmente interligados através de 17 enlaces, formando um grafo completo. Um grafo é dito completo quando todo vértice for adjacente a todos os outros vértices, o que o torna mais complexo de gerenciar. Como prova de conceito, serão ilustrados os processos de configuração, manutenção e controle de uma rota.

8.2.1 – Processo de Configuração

De forma análoga ao apresentado pelo cenário 1, o processo de configuração de uma rota inicia-se através de mecanismos que permitem ao sistema PBQoS encontrar a melhor rota entre dois pontos. Para isto, leva em consideração os requisitos mínimos de QoS exigidos por uma aplicação e as regras impostas pelo sistema (Políticas de Configuração). Com base nestas informações, o sistema PBQoS faz uma análise das informações definidas na MIB e define quais elementos de rede serão utilizados na composição da rota. Para ilustrar a topologia de rede deste cenário, considere a MIB de rede definida pela Tabela 39.

Tabela 39 – MIB geral da rede – Cenário 2

Enlace	Origem	Destino	Vazão	Perdas	Jitter	Atraso
1	1	2	100000.0	2.0	1.0	100.0
2	1	3	64.0	5.0	3.0	300.0
3	1	4	128000.0	3.0	2.0	150.0
4	1	5	34300.0	4.0	1.0	40.0
5	2	3	16000000.0	2.0	1.0	100.0
6	2	4	100000.0	1.0	1.0	50.0
7	2	5	50000.0	4.0	1.0	30.0
8	3	4	10000.0	2.0	1.0	20.0
9	3	5	60000.0	4.0	1.0	20.0
10	4	5	60000.0	4.0	1.0	10.0
11	1	6	60000.0	4.0	1.0	10.0
12	2	6	60000.0	4.0	1.0	10.0
13	3	6	60000.0	4.0	1.0	10.0
14	4	6	60000.0	4.0	1.0	10.0
15	5	6	60000.0	4.0	1.0	10.0

A Figura 80 ilustra a representação destas informações em forma de grafo (grafo completo).

Figura 80 – Topologia da rede – Cenário 2

Para gerenciar uma rede, o sistema deve conhecer o estado atual de cada enlace que a compõe. Para obter estas informações, o sistema faz uma leitura das informações presentes na MIB geral da rede, e calcula o QoS de cada rota. A Figura 81 ilustra a execução da aplicação PBQoS, onde é possível observar além dos campos de entrada da aplicação, a capacidade nominal de cada enlace e sua disponibilidade atual. Inicialmente todos os enlaces dispõem de 100% de seus recursos disponíveis e a cada iteração, esses recursos vão sendo consumidos e o gráfico por sua vez, é atualizado.

Figura 81 – Execução da Aplicação PBQoS

Para gerar carga na rede é preciso que o sistema realize a admissão de diversas aplicações simultaneamente, e para isto, foi considerado o mesmo conjunto de aplicações definido pela Tabela 28.

De forma análoga ao apresentado no Cenário 1, os resultados serão descritos de duas formas: a primeira visa ilustrar o número de aplicações admitidas e rejeitadas pelo sistema PBQoS. A segunda visa ilustrar o processo de manutenção e controle de uma rota, onde é possível visualizar o sistema PBQoS buscando por rotas alternativas quando os recursos de uma rota atingirem os limites de segurança impostos pelas políticas de controle.

Para dar inicio ao processo de configuração, considere a requisição definida pela Tabela 40, no qual apenas a Política *Maximizar Vazão* se encontra ativa.

Tabela 40 – Requisição de recursos – Política Maximizar Vazão (1)

Nós		SLA				Configuração
Origem	Destino	Vazão (Kbps)	Perda (%)	Jitter (ms)	Atraso (ms)	Política
1	5	50000	20	10	1000	Maximizar vazão

Após realizar a requisição através da aplicação PBQoS, os resultados gerados podem ser visualizados na Figura 82.

Figura 82 – Política *Maximizar Vazão* – Requisição de rota (1)

Conforme já definido no Cenário 1, após cada iteração, o sistema PBQoS armazena as informações da requisição em um sistema de *log*. Analisando o *log* gerado, é possível visualizar que a requisição foi atendida e houve a admissão na rota 50, composta pelos nós 1→4→5 conforme ilustra a Figura 83.

```

análise executada dia 11/ 9/2010  hora: 19:46:33 hs
origem:          destino:
  1              5
sla:      vazao      perda      jitter      atraso
          50000.0    20.0      10.0      1000.0
politica: vazao      perda      jitter      atraso
          1          0          0          0
  i  rota      vazao      perda      jitter      atraso      enlaces da
rota
  0  50      60000.00      6.9      3.00      160.00      1  4  5
matriz inuse
enlace org dest      vazão
  1     1   2         0.0
  2     1   3         0.0
  3     1   4      50000.0
  4     1   5         0.0
  5     2   3         0.0
  6     2   4         0.0
  7     2   5         0.0
  8     3   4         0.0
  9     3   5         0.0
 10     4   5      50000.0
 11     1   6         0.0
 12     2   6         0.0
 13     3   6         0.0
 14     4   6         0.0
 15     5   6         0.0

```

Figura 83 – *log* gerado pelo sistema PBQoS após o processo de admissão

Para compor a rota 50, os nós 1→4→5 foram alocados, e como pode ser observado na Figura 82 (campo inferior esquerdo), esta rota dispõe de recursos suficientes para atender a requisição feita pelo SLA:

vazão: 60000.00 kbps | **jitter:** 6.9 ms | **perda:** 3.00% | **atraso:** 160.00 ms

Vale ressaltar que os recursos consumidos no enlace 4→5 atingiram 83% de sua capacidade total disponível. E por exceder o limite de 70% de sua capacidade total disponível, as políticas de controle foram ativadas e um alerta amarelo foi enviado ao administrador da rede. Conforme as requisições vão sendo admitidas, a MIB geral da rede vai sendo atualizada de acordo com a quantidade de recursos utilizados pelas admissões. Para dar continuidade ao processo, considere uma nova requisição definida pelo SLA na Tabela 41.

Tabela 41 – Informações de uma requisição ao sistema PBQoS (2)

Nós		SLA				Configuração
Origem	Destino	Vazão (Kbps)	Perda (%)	Jitter (ms)	Atraso (ms)	Política
5	1	40000	20	10	1000	Maximizar vazão

Após realizada a requisição através da aplicação PBQoS, os resultados gerados podem ser visualizado na Figura 84.

Figura 84 – Política Maximizar Vazão – Requisição de rota (2)

De forma análoga ao processo pela requisição anterior, o sistema PBQoS admitiu a requisição com base no SLA requisitado, fornecendo assim os recursos

solicitados pelo SLA. Conforme ilustrou a Figura 84, a rota 54 dispõe de recursos suficientes para atender a requisição feita pelo SLA:

vazão: 60000.00 kbps | **jitter:** 9.7 ms | **perda:** 5.00% | **atraso:** 320.00 ms

Ao admitir a requisição, os recursos consumidos no enlace 1→4 atingiu 71% de uso dos recursos totais disponíveis, ultrapassando o limite de 70% imposto pela política de controle. Dessa forma, um alerta amarelo é enviado ao administrador da rede.

Para ilustrar o processo completo de configuração no Cenário 2, foram realizadas 30 requisições com diversos perfis entre diferentes nós de origem e destino. Após 30 requisições realizadas, diversos alertas amarelos foram emitidos. Como ilustra a Figura 85, os enlaces 2→4 e 4→5 excederam o limite de segurança de 85% de sua capacidade total disponível. Dessa forma, alertas vermelhos são emitidos ao administrador da rede, fazendo com que as políticas de manutenção sejam ativadas pelo sistema. Quando um enlace envia um alerta vermelho, o sistema PBQoS entra em um novo estado de operação e ativa as políticas de manutenção em busca de uma nova rota para melhor distribuir os recursos na rede.

Figura 85 – Política *Maximizar Vazão* – Requisição de rota (30)

De forma análoga ao apresentado pela Política Maximizar Vazão, o mesmo processo foi realizado para as demais políticas definidas neste trabalho. Um

resumo dos resultados gerados para cada política ativa pode ser visualizada na Tabela 42.

Tabela 42 – Resumo dos resultados gerados – Processo de configuração

	Requisição	Admissão	Rejeição	% admitido
Maximizar vazão	30	24	6	80,0
Minimizar perda	30	23	7	76,7
Minimizar jitter	30	23	7	76,7
Minimizar atraso	30	23	7	76,7

8.2.2 – Processo Manutenção

Conforme já descrito anteriormente, o processo de manutenção de rota visa manter e controlar os recursos alocados para atender uma determinada requisição. Quando os limites impostos pelas políticas de controle são atingidos, as políticas de controle e manutenção de rota são ativadas. Com base no SLA definido na Tabela 37, esta simulação tem como objetivo principal, ilustrar o sistema PBQoS buscando por rotas alternativas quando o limite de segurança imposto pela política de controle for atingido.

Tabela 43 – SLA utilizado para gerar carga na rede

Nós		SLA			
Origem	Destino	Vazão (Kbps)	Perda (%)	Jitter (ms)	Atraso (ms)
1	4	3000	10	8	100

Para cada política ativa, diferentes rotas foram selecionadas pelo sistema PBQoS e os resultados gerados por cada requisição podem ser visualizados através da Tabela 44.

Tabela 44 – Rotas selecionadas para cada requisição

Requisição	Maximizar Vazão	Minimizar Atraso	Minimizar Jitter	Minimizar Perdas	Visão Global de QoS
1	1→6	1→6	1→6	1→6	1→6
2	1→4→6	1→6	1→6	1→6	1→6
3	1→2→6	1→6	1→6	1→6	1→6
4	1→6	1→6	1→6	1→6	1→6
5	1→4→6	1→6	1→6	1→6	1→6
6	1→2→6	1→6	1→6	1→6	1→6
7	1→6	1→6	1→6	1→6	1→6
8	1→4→6	1→6	1→6	1→6	1→6
9	1→2→6	1→6	1→6	1→6	1→6
10	1→2→5→6	1→6	1→6	1→6	1→6
11	1→6	1→6	1→6	1→6	1→6
12	1→4→6	1→5→6	1→5→6	1→2→6	1→2→6
13	1→2→6	1→5→6	1→5→6	1→2→6	1→5→6
14	1→2→5→6	1→5→6	1→5→6	1→2→6	1→2→6
15	1→6	1→5→6	1→5→6	1→2→6	1→2→6
16	1→4→6	1→5→6	1→5→6	1→2→6	1→5→6
17	1→2→6	1→5→6	1→5→6	1→2→6	1→2→6
18	1→2→5→6	1→2→6	1→2→6	1→2→6	1→5→6
19	1→6	1→2→6	1→2→6	1→2→6	1→2→6
20	1→4→6	1→2→6	1→2→6	1→2→6	1→2→6
21	1→2→6	1→2→6	1→2→6	1→2→6	1→5→6
22	1→2→5→6	1→2→6	1→2→6	1→2→6	1→2→6
23	1→5→6	1→2→6	1→2→6	1→2→4→6	1→2→6
24	1→6	1→2→6	1→2→6	1→2→4→6	1→5→6
25	1→4→6	1→2→6	1→2→6	1→2→4→6	1→2→4→6
26	1→2→6	1→2→6	1→2→6	1→2→4→6	1→2→4→6
27	1→2→5→6	1→2→6	1→2→6	1→2→4→6	1→2→6
28	1→5→6	1→2→6	1→2→6	1→2→4→6	1→4→6
29	1→6	1→2→5→6	1→4→6	1→2→4→6	1→2→4→6
30	1→4→6	1→2→5→6	1→4→6	1→2→4→6	1→2→4→6
31	1→2→6	1→2→5→6	1→4→6	1→4→6	1→5→6
32	1→2→5→6	1→2→5→6	1→4→6	1→4→6	1→2→6
33	1→6	1→2→5→6	1→4→6	1→4→6	1→4→6
34	1→5→6	1→4→6	1→4→6	1→5→6	1→2→4→6
35	1→4→6	1→4→6	1→4→6	1→5→6	1→2→6
36	1→2→6	1→4→6	1→4→6	1→5→6	1→4→6
37	1→6	1→4→6	1→4→6	1→5→6	1→2→4→6
38	1→5→6	1→4→6	1→4→6	1→5→6	1→2→4→6
39	1→4→6	1→4→6	1→4→6	1→5→6	1→2→4→6
40	1→2→6	1→4→6	1→2→5→6	1→4→3→6	1→4→3→6
41	1→4→3→6	1→4→6	1→2→5→6		
42	1→6	1→4→6	1→2→5→6		
43	1→5→6	1→4→6	1→2→5→6		
44	1→4→6	1→4→6	1→2→5→6		
45	1→2→6	1→4→3→6	1→4→3→6		

Os resultados foram obtidos através da execução da aplicação PBQoS sobre a arquitetura proposta neste trabalho. Com base no SLA requerido e nas regras de configuração definidas pela(s) política(s) ativa(s), diversas requisições foram realizadas simultaneamente.

Com base nos resultados obtidos (Tabela 44), é possível observar que o quanto maior o número de políticas ativas simultaneamente, menor será o número de requisições admitida. Isto ocorre devido ao número de regras impostas pelas políticas, pois, quanto mais regras existirem, mais rígido será o cálculo de uma rota.

Visto que os recursos disponíveis na rede são limitados e em cada requisição admitida estes recursos vão sendo consumidos, este cenário mostrou que o sistema PBQoS foi capaz de atender diversas requisições imultâneas até que os recursos da rede fossem esgotados. No momento em que os recursos utilizados por uma rota vão sendo liberados pelas requisições, estes recursos são devolvidos ao sistema, que por sua vez, já pode utilizá-los em novas requisições.

8.3 – Considerações Finais

Como prova de conceito, este Capítulo apresentou a implementação computacional do sistema PBQoS através de dois cenários distintos. O Cenário apresentou uma topologia relativamente simples com intuito de ilustrar de forma detalhada o processo de configuração, manutenção e controle de uma rota. Por se tratar de uma rede totalmente interconectada, o Cenário 2 apresentou uma topologia mais complexa capaz de ilustrar não somente os processo de configuração, manutenção e controle, mas também a escalabilidade do algoritmo de busca de rota desenvolvido neste trabalho. Com base nesta simulação, foi possível observar o sistema PBQoS buscando por rotas alternativas, quando os limites impostos pelas políticas de controle são atingidos. Não havendo recursos disponíveis na rede em um dado momento, o sistema PBQoS não admite a requisição.

Capítulo 9

Conclusão

Definir uma arquitetura de gerenciamento para prover e gerenciar de forma eficiente os recursos da rede é primordial para o sucesso das aplicações e serviços que envolvam distribuição e transmissão de conteúdo pela rede. Neste sentido, a concepção de estratégias para distribuição eficiente de dados é uma tendência fortemente verificada nas aplicações avançadas. Tais estratégias devem identificar as rotas de distribuição e considerar aspectos como velocidade do enlace, capacidade de transmissão das redes e recursos para melhor atender uma determinada requisição.

O sistema de gerenciamento PBQoS proposto neste trabalho definiu uma arquitetura hierárquica e distribuída aplicada em um cenário de rede sobrepostas. Trata-se de uma arquitetura baseada nos padrões definidos pela IETF, nos quais é composta por elementos específicos com funções bem definidas em busca de encontrar a melhor rota capaz de atender as necessidades de uma aplicação. Para isto, tem como base os requisitos de QoS exigidos pelas aplicações da Internet avançada e faz uso das políticas de configuração, manutenção e controle para definir o funcionamento do sistema.

A implementação do sistema PBQoS se baseia em técnicas de otimização conhecidas, e define um algoritmo capaz de dar suporte ao funcionamento da arquitetura proposta. Com base nos conceitos básicos de otimização, o trabalho aqui proposto se fundamentou em princípios matemáticos que auxilia na definição da melhor rota para atender os requisitos mínimos exigidos pelo perfil de uma aplicação (SLA).

Conhecer a topologia de rede e seu estado atual é imprescindível para que o sistema PBQoS possa realizar o cálculo de uma rota com base na demanda do tráfego e definir os recursos disponíveis de forma eficiente. Dessa forma, a arquitetura de gerenciamento proposta neste trabalho visa definir, manter e controlar uma rota que seja capaz de atender não somente as necessidades de uma aplicação, mas também as expectativas dos usuários (QoE).

O sistema PBQoS dispõe de mecanismos que monitoram a rede física (Internet) freqüentemente com objetivo de mensurar continuamente os atributos

dinâmicos como vazão, taxa de perda, atrasos e variação de atraso. Dessa forma é possível fechar a malha de gerenciamento e determinar os melhores recursos para atender a demanda dos usuários.

Como prova de conceito foi desenvolvida uma aplicação capaz de configurar e administrar o sistema de gerenciamento. Para isto, utiliza o algoritmo PBQoS desenvolvido neste trabalho para realizar o processo de definição, manutenção e controle de uma rota na rede sobreposta. Este algoritmo dispõe de funcionalidades importantes para que o sistema de PBQoS opere de forma eficiente na busca pela melhor rota disponível.

Como trabalhos futuros pretende-se:

- Definir um modelo que seja capaz de se comunicar com um simulador de rede de forma a manter a MIB atualizada com dados reais ou então implementar este modelo em um ambiente real de rede, viabilizando assim o desenvolvimento de diversos serviços e aplicações;
- Aplicar esta arquitetura em um cenário real de rede e expandi-la para cenários multi-domínio.
- Definir um mecanismo que permita adotar critérios de pesos para cada política ativa no sistema;
- Como prova de conceito, apenas a capacidade nominal dos enlaces (vazão) foi levando em consideração em uma das etapas do processo de admissão de um serviço. Um serviço só é admitido, caso os recursos solicitados não excedam os limites impostos pelo sistema. Pretende-se nos próximos trabalhos, adicionar os demais requisitos de QoS (atraso, variação do atraso e perdas) no processo de admissão de um novo serviço, tornando o processo ainda mais personalizado.
- Representar graficamente o estado atual de todos os requisitos de QoS monitorados no sistema, facilitando assim a administração do sistema.
- Criar meios que permitam ao administrador, além de criar, manipular e deletar políticas através da interface gráfica, possa realizar o

chaveamento de uma rede para outra de forma dinâmica, facilitando assim o gerenciamento de vários cenários simultaneamente.

- Aumentar a complexidade das políticas, visando tornar o sistema mais autônomo.

Referências Bibliográficas

- (Almeida, 2006)** Almeida, F. L. A Management and Configuration Infrastructure for RNP's Digital Video Distribution Service. Proceedings of the IEEE International Telecommunications Symposium 2006 (ITS 2006), Fortaleza – CE.
- (Alves, 2004)** Alves, C. L. S; VoIP – Voz sobre IP. Monografia apresentada a Faculdade de Ciências Aplicadas de Minas, 2004.
- (Andersen, 2001a)** Andersen, D; Balakrishnan, H; Kaashoek, F. Morris, R. The Case for Resiliente Overlay Networks. Proceedings of IEEE Computer Society, 2001.
- (Andersen, 2001b)** Andersen, D; Balakrishnan, H; Kaashoek, F. Morris, R. Resiliente Overlay Networks. Proceedings of 18th ACM Symp. On Operation Systems Principles (SOPS), Baff, Canadá. Outubro de 2001.
- (Ausiello, 2003)** Ausiello, Giorgio; et al. Complexity and Approximation (Corrected ed.), Springer, 2003.
- (Beller, 2005)** Beller, A. G; Uma Arquitetura para Gerenciamento de QoS Baseado em Políticas. Dissertação - Pontifícia Universidade Católica do Paraná. Programa de Pós Graduação em Informática Aplicada. Curitiba, 2005. 144p.
- (Beuran, 2003)** Beuran, R; Ivanovici, M; Dobinson, B; Network Quality of Service Measurement System for Application Requirements Evaluation. Pág 380 à 387. SPECTS '03. disponível em http://www.jaist.ac.jp/~razvan/publications/qos_measurement.pdf acesso fev.2009.
- (Blake, 1998)** Blake, S. et al., “An Architecture for Differentiated Services,” IETF RFC-2475, Dec. 1998.
- (Bohn, 2003)** Bohn, W; Braun, P; Policy Based Architecture for UMTS Multimedia Domain. *Proceedings of the second IEEE International Symposium on Network Computing and Application (NCA '03)*, 2003.
- (Bondi, 2000)** André B. Bondi; 'Characteristics of scalability and their impact on performance', Proceedings of the 2nd international workshop on Software and performance, Ottawa, Ontario, Canada, 2000, ISBN 1-58113-195-X, pages 195 - 203
- (Brandino, 2009)** Brandino, W. L. Sistemas Multimídia. Disponível em <http://www.wandreson.com/download/training-networking-multimediasystems01.ppt#256,1>, Sistemas Multimídia, acesso em abril de 2009.
- (Cardoso, 1994)** Cardoso, A. A; Síntese Racional de Automatizada de Cavernas de Embarcações. Dissertação de Mestrado. Escola Politécnica da Universidade de São Paulo. São Paulo, 1994.
- (Cardoso, 1995)** Cardoso, A. A; Augusto, O. B; An Expert System for Design Ship Structures. Proceedings of 8th International Symposium on Artificial Intelligence, Monterrey, 1995.
- (Changkun, 2007)** Changkun, W. Policy-based Network Management. Proceedings on Communication Technology Proceedings, 2000. WCC - ICCT 2000, pag 101-105 vol.1. International. Disponível em http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=889177, acesso em março de 2007.
- (Coutinho, 2009)** Coutinho, L. G; Delay Tolerant Networks (DTN). Escola Politécnica da Universidade Federal do Rio de Janeiro – UFRJ, 2006. Disponível em http://www.gta.ufrj.br/grad/06_2/gustavo/inicial.htm, acesso em 02/2009.
- (Cormen, 2002)** Cormen, Thomas H., et al. Introduction to Algorithms. , 2nd ed. Cambridge: The MIT press, 2002.

- (Doval, 2003)** Doval, D; O'Mahony. *Overlay Networks - A Scalable Alternative for P2P*. IEEE Computer Society, 2003.
- (Elster, 2003)** Elster, K, H. *Modern Mathematical Methods of Optimization*. Vch Pub. 1993.
- (ETSI-TS, 2002)** ETSI TS 101329-2 (2002), *Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) Release 3; End-to-end Quality of Service in TIPHON Systems; Part 2: Definition of speech Quality of Service (QoS) classes*.
- (ITU-T, 2002)** ITU-T Recommendation F.700 (2000), *Framework Recommendation for multimedia services*.
- (Filsfils, 2007)** Filsfils, C; Evans, J. *Deploying IP and MPLS QoS for Multiservice Networks: Theory and Practice*. Morgan Kaufmann Publishers, 2007.
- (Fluckinger, 1995)** F. Fluckinger; *Understanding Networked Multimedia: Applications and Technology*; Prentice Hall, 1995.
- (Fry, 2004)** Fry, G; West, R; Adaptive routing of QoS constrained media streams over scalable overlay topologies. In 10th IEEE Real-Time and Embedded Technology and Applications Symposium, 2004.
- (Harman, 1990)** Harman, G. *The intrinsic Quality of Experience*. Action Theory and Philosophy of Mind. Princeton University. 1990.
- (Harroud, 2005)** Harroud, H; Karmouch, A. A Policy Based Context-aware Agent Framework to Support Users Mobility. Proceedings of the IEEE Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources Conference/E-Learning on Telecommunication Workshop, 2005.
- (Hetzer, 2007)** Hetzer, D. Miloucheva, I. Jonas, K. Niephaus, C. Policy based resource management for QoS aware applications in heterogeneous network environments. Proceedings of the IEEE Communications and Networking in China, 2007.
- (Hill, 1997)** Mark D. Hill, 'What is scalability?' in ACM SIGARCH Computer Architecture News, December 1990, Volume 18 Issue 4, pages 18-21, (ISSN 0163-5964) and Leticia Duboc, David S. Rosenblum, Tony Wicks, 'Doctoral symposium: presentations: A framework for modelling and analysis of software systems scalability' in Proceeding of the 28th international conference on Software engineering ICSE '06, May 2006. ISBN 1-59593-375-1, pages 949 – 952
- (Hromkovic, 2002)** Hromkovic, Juraj. *Algorithmic for Hard Problems*, Texts in Theoretical Computer Science (2nd ed.), Springer, 2002.
- (Hunt, 2008)** Hunt, R; *IP Quality of Service Architectures*. IEEE, 2001.
- (IETF, 2008)** The IETF Policy Framework Working Group: Charter available at <http://www.ietf.org/html.charters/policy-charter.html>, acesso em Nov. 2008.
- (Miras, 2002)** Miras, D; A Survey of Network QoS Needs of Advanced Internet Applications. Internet2 QoS Working Group. University College London. November 2002.
- (ITU-T, 2002)** ITU-T – International Telecommunication Union. *Transmission Systems and media, Digital systems and Networks. Quality of service and performance, End-user multimedia QoS categories*. Recommendation G.1010. Genova, 2002.
- (ITU-T, 2001)** International Telecommunication Union (ITU-T). *Network performance objectives for IP-based services*. Series Y: Global Information Infrastructure. Draft New Recommendation Y.1541, October 2001.
- (ITU-T, 1998)** International Telecommunications Union (ITU). ITU-T-G.114, *General Characteristics of International Telephone Connections and International Telephone Circuits*, 1998.

- (Kawarasaki, 2004)** Kawarasaki, M; Atarashi, R; Policy Based Content Delivery Management Using Metadata. Proceedings of the 2004 International Symposium on Applications and the Internet Workshops (SAINTW'04). IEEE, 2004.
- (Kim, 2005)** Kim, G. Kim J. e Na, J. Design and Implementation of Policy Decision Point in Policy-Based Network. Eletronics and Telecommunications Research Institute. Proceedings of the IEEE Fourth Annual ACIS International Conference on Computer and Information Science (ICIS'05).
- (Knuth, 1997)** Knuth, D. E. *The Art Of Computer Programming Vol 1. 3rd ed.*, Boston: Addison-Wesley.
- (Kowtha, 2006)** Kowtha, S; Jiang, X; Na. N-State Driven Policy-based Network Management to control end-end network behaviors. Proceedings of the Seventh IEEE International Workshop on Policies for Distributed Systems and Networks, 2006.
- (Kumar, 2001)** Kumar, V. et al. IP Telephony with H.323. Wiley, 2001.
- (Lacerda, 2007)** Lacerda, A. G et al., Serviço de Distribuição de Conteúdo em uma plataforma IPTV. II Congresso de Pesquisa e Inovação da Rede Norte Nordeste de Educação Tecnológica, 2007.
- (Lima, 2009)** Lima, R. A. Lamarca, E; Oliveira, A. H. G. Aplicações multimídia: Desenvolvimento, uso e análise de desempenho. abril/2009.
- (Lowekamp, 2003)** Bruce Lowekamp, Brain Tierney, Les Cottrell, Richard Hughes-Jones, Thilo Kielmann and Martin Swany. A Hierarchy of Network Measurements for Grid Applications and Services Documents (draft) Global Grid Forum NMWG Feb 17, 2003.
- (Macedo, 2004)** Macedo, F. D; Oliveira, B. L; Loureiro A. F. A. Integrando Redes Overlay e Redes de Sensores sem fio. Telecomunicações, vol. 7, n.1. Junho de 2004.
- (Michael, 2007)** Michael, M.; J.E. Moreira, D. Shiloach, R.W. Wisniewski (March 26, 2007). "Scale-up x Scale-out: A Case Study using Nutch/Lucene". Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4228359.
- (Mieghem, 2006)** Mieghem, P. V; Performance Analysis of Communications Networks and Systems. Cambridge University Press, 2006.
- (Mordecai, 2003)** Mordecai, A. Nonlinear Programming: Analysis and Methods. Dover Publishing, 2003.
- (Nakao, 2003)** Nakao, A; Peterson, L; Bavier, A; A Routing Underlay for Overlay Networks. Departamento f Computer Science, Princeton University. Proceedings of SIGCOMM'03, Germany, 2003.
- (Nascimento, 2009)** Nascimento, A. M; Moraes, D. M; Redes Overlay. Universidade Federal do Paraná. Disponível em <http://www.inf.ufpr.br/elias/sisdis/overlay-aldo-dionei.pdf>. acesso 02/2009.
- (Ok, 2006)** Ok, K; Hong, D. C; Chang B. S; The Design of Services Management System based on Policy-based Network Management. Proceedings of the International conference on Networking and Services. IEEE Computer Society, 2006.
- (Oqily, 2007)** Oqily, A. Karmouch, A. Automating Overlay Networks Management. School of Information Technology & Engineering (SITE). University of Ottawa. 21 IEEE. International Conference on Advanced Networking and Applications (AINA'07).
- (Overby, 2010)** Stephanie Overby – SLA (Service Level Agreement). Now!Digital Business licenciado pela International Data Group, Inc., 2010.

- (Uchoa, 2007)** Uchôa, D. C; Kulesza, R; Bressan, G; Melo, R. S; Matushima, R; Oversea: Towards a Scalable and Effective Architecture for Overlay Networks. Proceedings of XIII Brazilian Symposium on Multimedia and the Web. Gramado – RS, outubro de 2007.
- (Panos, 2000)** Panos Y. Papalambros and Douglass J. Wilde. Principles of Optimal Design : Modeling and Computation, Cambridge University Press, 2000.
- (Perlman, 1999)** Perlman, Radia. Interconnections, Second Edition: Bridges, Routers, Switches, and Internetworking Protocols. Boston: Addison Wesley, 1999.
- (RFC 2328)** Moy, J. “OSFP Version 2”, RFC 2338, April 1998.
- (RFC 2680)** Almes, G., et al. “A One-way Packet Loss Metric for IPPM”, RFC 2680, September 1999.
- (RFC 2679)** Almes, G., et al. “A One-way Delay Metric for IPPM”, RFC 2679, September 1999.
- (RFC 3393)** Demichelis, C., et al. “IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)”, RFC 3393, November 2000.
- (Rich, 1990)** Rich E; Knigh, K. Artificial Intelligence. McGraw-Hill Science/Engineering/Match; 2 edition, december, 1990.
- (Santana, 2000)** Santana, H. Qualidade de Serviço (QoS) em redes IP: Princípios básicos, Parâmetros e Mecanismos. Universidade Santa Cecília – Unisanta, 2000.
- (Silveira, 2009)** Silveira, R. M. Redes de Alta Velocidade e as Aplicações Multimídia. Disponível em <http://www.redes.usp.br/conteudo%5Cdocumentos%5Crav%20e%20aplica%C3%A7%C3%B5es%5CRAVeAplic.pdf> acesso abril/2009.
- (Sloman, 1994)** M. Sloman, “Policy Driven Management For Distributed Systems,” J. Net. And Sys.Mgmt., vol. 2, no. 4, Dec. 1994, pp. 333–60.
- (Sreenivas, 2009)** Sreenivas, D; Policy Control for IPTV domain. Nokia Siemens Networks. Bangalore, Índia.
- (Strassner, 2004)** Strassner, J. C. *Policy-Based Network Management – Solutions for the Next Generation*. Editora Elsevier, 2004.
- (Strassner, 2007)** Strassner, J; Souza. J. N; Raymer, D; Samudrala, S; Davy, S; Barrett, K. The Design of a new Policy Model to Support Ontology-Driven Reasoning for Autonomic Networking. Proceedings of the IEEE Network Operations and Management Symposium, LANOMS 2007. Petrópolis – RJ.
- (Subramanian, 2003)** Subramanian, L; Stoica, I; Balakrishnan, H; Katz, H. R; OverQoS: *Offering Internet QoS Using Overlays*. ACM SIGCOMM Computer Communications Review, 2003.
- (Teixeira, 2004)** Teixeira, G. Carvalho, P. CORDENA – Um framework para gestão de redes baseadas em políticas. Universidade do Ninho. 2004.
- (Thomas, 2001)** Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*, Second Edition. Section 22.3: Depth-first search, pp. 540–549. MIT Press and McGraw-Hill, 2001.
- (Ushida, 2006)** Ushida, M; Kamei, S; Kawahara, R; Abe, T. QoS-Aware Overlay Routing with Limited Number of Alternative route candidates and Its Evaluation. IEICE Trans. Commun. Set. 2006.
- (Vegesna, 2001)** Vegesna, S. IP Quality of Service. EUA, Cisco Press, 2001.

(Verma, 2002) Verma, D. C; Simplifying Network Administration Using Policy-Based Management. IBM Thomas J. Watson Research Center. IEEE Network. 2002.

(Verma, 2000) Verma, D. C. Policy-Based Networking – Architecture and Algorithms. New Riders Publishing, 2000.

(Wang, 2001) Wang, Z. Internet QoS: Architectures and Mechanisms for Quality of Service. Bell Labs, Lucent Technologies - USA, 2001.

(Wright, 2009) Wright, M. Using Policies for Effective Network Management. International Journal of Network Management Int. J. Network Mgmt. 9, 118-125, 1999.

(Yang, 2008) Yang X. S. Introduction to Mathematical Optimization: From Linear Programming to Metaheuristics, Cambridge Int. Science Publishing. 2008

(Zambenet, 2003) Zambenet, L. G; Vaguetti, L; Almeida, M. J. B; Tarouco, L. M. R. A PBNM System for Integrated QoS and Multicast Management. Proceedings of the 4th International Workshop on Policies for Distributed Systems and Networks (POLICY'03), IEEE, 2003.