

Universidade Estadual de Campinas - **UNICAMP**
Faculdade de Engenharia Elétrica e de Computação - **FEEC**
Departamento de Telemática - **DT**

Qualidade de Serviço em Rede IP Utilizando Lógica Fuzzy

Autor: Raulison Alves Resende

Mestre em Engenharia Elétrica (Stricto Sensu, UNICAMP - Campinas/SP, 2001)

Especialista em Ciência da Computação (Lato Sensu, UCG - Goiânia/GO, 1997)

Tecnólogo em Processamento de Dados (Graduação, IUESO - Goiânia/GO, 1995)

Orientador: Prof. Dr. Akebo Yamakami

Tese de Doutorado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: **Telecomunicações e Telemática**.

Banca Examinadora

Prof. Dr. Akebo Yamakami (presidente) ... DT/FEEC/UNICAMP, Campinas - SP
Prof. Dr. Adriano Domingos Neto ISCA Faculdades, Limeira - SP
Prof. Dr. Helio Waldman DECOM/FEEC/UNICAMP, Campinas - SP
Prof. Dr. João de Deus Lima FANAT/UERN, Mossoró - RN
Prof. Dr. José Patrocínio da Silva CESET/UNICAMP, Limeira - SP
Prof. Dr. Yuzo Iano DECOM/FEEC/UNICAMP, Campinas - SP

Campinas, São Paulo, Brasil
31 de julho de 2006

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

R311q Resende, Raulison Alves
Qualidade de serviço em rede IP utilizando lógica fuzzy
/ Raulison Alves Resende. --Campinas, SP: [s.n.], 2006.

Orientador: Akebo Yamakami
Tese (doutorado) - Universidade Estadual de Campinas,
Faculdade de Engenharia Elétrica e de Computação.

1. Intranet (Redes de computação). 2. Redes de
computação - Protocolos. 3. Telecomunicações - Trafego.
4. Provedores de serviços da Internet. 5. Sistemas de
controle ajustável. 6. Analise de redes (Planejamento). 7.
Lógica difusa. I. Yamakami, Akebo. II. Universidade
Estadual de Campinas. Faculdade de Engenharia Elétrica e
de Computação. III. Título.

Titulo em Inglês: Quality of service in IP network using fuzzy logic

Palavras-chave em Inglês: Fuzzy logic, Linguistic rule, Shortest path, Quality of
service (QoS) routing

Área de concentração: Telecomunicações e Telemática.

Titulação: Doutor em Engenharia Elétrica

Banca examinadora: Adriano Domingos Neto, Helio Waldman, João de Deus Lima,
José Patrocinio da Silva e Yuzo Iano

Data da defesa: 31/07/2006

Agradecimentos

Ao meu orientador, Prof. Dr. Akebo Yamakami meus sinceros agradecimentos por ter me aceito como orientado em 1999 e todas as orientações dadas neste período (mestrado e doutorado) de sete anos ininterruptos de trabalho;

Ao Prof. Dr. Ivanil S. Bonati pelas suas orientações, pelas muitas horas de discussão, o que levou ao aperfeiçoamento deste trabalho é também à minha formação profissional e acadêmica;

Ao Pesquisador e colega do CPqD, Dr. Antonio C. Lavelha pelas suas sugestões neste documento e também pelos artigos publicados em conjunto;

Ao amigo Dr. Adriano D. Neto pela grande colaboração na estruturação e sugestões neste trabalho;

Os meus sinceros agradecimentos aos membros da banca examinadora desta tese;

Ao Centro de Pesquisa e Desenvolvimento em Telecomunicações (CPqD), pelo apoio técnico e financeiro e a todos os colegas de trabalho desta instituição;

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pelo apoio financeiro parcial;

Às famílias Adriano Neto & Teresa, Sr. Antonio & Sra. Wilma, Dirson & Célia e João de Deus & Zuleika meu grande abraço de agradecimento;

Aos meus amigos Adrian Ferreira, Fabiano Padua, Helen Teixeira, José Patrocínio, Júlio Cesar Magro, Luiz Bonani, Marcos Siqueira, Nelson Uto, Rangel Arthur, Rubia Mara e Sandro Rossi;

Aos demais colegas e funcionários da FEEC-UNICAMP, os quais, cada um da sua forma, contribuíram para a minha formação acadêmica;

A todos quanto direta ou indiretamente contribuíram para este trabalho, o meu muito obrigado.

A Deus

Salmo 26:5 - Assim, no dia mau ele me esconderá na sua tenda, Ocultar-me-á no recôndido de seu tabernáculo, Sobre um rochedo me erguerá.

Obrigado por tudo sempre!

A Nossa Senhora

Eclesiástico 24:24 - Sou a mãe do puro amor, do temor (de Deus), Da ciência e da santa esperança.

Obrigado por todas as graças sempre!

Dedicatória

A minha esposa, Luciana B. Leite Resende e meus dois anjos, Maria Luísa L. Resende e Isadora Valeriana L. Resende, pelo amor, carinho, felicidade e apoio nos momentos difíceis;

Aos meus pais, Manoelino de Resende e Maria de Fátima A. Resende, que sempre lutaram pelo meu futuro;

Aos meus sogros, Adão Leite e Dinorá Leite, pelo apoio nesta minha caminhada;

Aos meus irmãos Leonardo A. Resende e Anderson A. Resende pelos incentivos;

Aos meus tios, tias, primos, primas, cunhados, cunhadas, sobrinhos e sobrinhas, pelo carinho nessa jornada;

E em memória da minha irmã, Gisele A. Resende, que sempre torceu pelo meu sucesso.

...
Você não sabe o quanto eu caminhei
Pra chegar até aqui
Percorri milhas e milhas antes de dormir
Eu nem cochilei
Os mais belos montes escalei
Nas noites escuras de frio chorei, ei, ei, ieh
ei, ei, ei, ei, ei, ei, iehh

...
A Estrada - Cidade Negra (Toni Garrido / Lazão / Da Gama / Bino)

Resumo

A teoria de conjuntos *fuzzy* provê um ferramental para desenvolver processos aproximados de raciocínio quando a informação disponível é incerta, incompleta, imprecisa ou vaga. Com as novas ferramentas emergentes na área de inteligência computacional, tais como lógicas não padronizadas, redes neurais e raciocínio simbólico, esta nova teoria é um acréscimo bem-vindo ao repertório de ferramentas tradicionais apropriadas. Neste sentido, esta tese propõe três estratégias para o provimento de qualidade de serviço em redes IP, usando lógica *fuzzy*. Como primeira proposta é implementado um módulo inteligente para gerenciamento da QoS por meio da criação de políticas para roteadores do tipo DiffServ. A segunda estratégia consiste em um algoritmo *fuzzy* que resolve o problema de roteamento baseado em restrição de tempo com mínimo custo. Esta proposta apresentou resultados satisfatórios quando comparada com a solução sem lógica *fuzzy*. E, por último, se propõe um controlador de admissão de conexão para tratar as informações com certo grau de incertezas em redes IP com MPLS. O controlador apresenta as seguintes vantagens: flexibilização da admissão de conexão e possibilidade de incluir mais informações da rede e do tráfego na tomada da decisão sem aumentar consideravelmente a complexidade do controlador.

Palavras-chave: lógica *fuzzy*, regra lingüística, caminho mínimo, qualidade de serviço e roteamento.

Abstract

Fuzzy set theory provides a machinery for carrying out approximate reasoning processes when available information is uncertain, incomplete, imprecise, or vague. With the emergence of new tools in the area of computational intelligence, such as non standard logics, neural networks, and symbolic reasoning, this new theory is a welcome addition to the repertoire of traditional appropriate tools. In this sense, this thesis proposes three different strategies to apply Fuzzy Logic to provide QoS in IP Networks. The first one is the implementation of an intelligent, policy-based module for QoS management. The module permits the creation of policies to configure QoS in routers and a practical implementation has been performed for a DiffServ router. The second strategy proposes a fuzzy algorithm to solve the problem of routing based on time restrictions with minimum costs. The results obtained from this algorithm were satisfactory when compared to similar solutions that did not involve fuzzy logic. Finally, the third strategy proposes a Connection Admission Controller to process information that contains a high level of uncertainty in IP networks over MPLS. The controller offers the following advantages: flexibility for connections admission and the possibility of including more network and traffic information in the decision making process without considerably increasing the controller complexity.

Keywords: fuzzy logic, linguistic rule, shortest path, quality of service (QoS), and routing.

Sumário

Lista de Figuras	xv
Lista de Tabelas	xvii
Glossário	xix
1 Introdução	1
Introdução	1
1.1 Contextualização	1
1.2 Proposta de pesquisa	3
1.2.1 Definição do problema	3
1.3 Contribuições da tese	4
1.4 Organização da tese	4
2 Gerenciamento da QoS em Redes IP com Base em Políticas	5
2.1 Contribuição	5
2.2 Descrição do sistema de gestão de QoS por meio de políticas	6
2.2.1 Interfaces do sistema QoSM	6
2.3 Funcionamento do Módulo Controlador <i>Fuzzy</i>	9
2.3.1 Dados de entrada	10
2.3.2 Regras <i>fuzzy</i>	10
2.3.3 Fluxo básico	11
2.3.4 Fluxos alternativos	15
2.3.5 Parâmetros de saída	15
2.4 Resultados	17
3 Roteamento Baseado em Restrições	19
3.1 Contribuição	19
3.2 Formulação do problema de roteamento	19
3.2.1 Elementos dos algoritmos	19
3.2.2 Formulação do problema e algoritmo de Ford-Bellman-Moore	20
3.3 Problema de roteamento baseado em restrições	22
3.3.1 Problema de roteamento sem usar a lógica <i>fuzzy</i>	22
3.3.2 Algoritmo sem usar a lógica <i>fuzzy</i>	23

3.3.3	Problema de roteamento baseado em lógica <i>fuzzy</i>	24
3.3.4	Algoritmo proposto	25
3.4	Comparação de desempenho	27
3.4.1	Modelo de rede e tráfego	27
3.5	Proposta de controlador nebuloso de admissão de conexão	32
3.5.1	Arquitetura do controlador nebuloso	32
3.5.2	Funções de pertinência do controlador	34
3.5.3	Variáveis lingüísticas do controlador	34
3.5.4	O conjunto de regras <i>fuzzy</i>	38
3.6	Ambiente de teste	38
3.6.1	Sobre a ferramenta de testes	38
3.6.2	Cenários simulados	38
3.6.3	Resultados	40
4	Conclusões e trabalhos futuros	45
4.1	Conclusões	45
4.2	Sugestões para trabalhos futuros	46
	Referências bibliográficas	47
A	Trabalhos publicados pelo autor	51
A.1	Publicações relacionadas ao capítulo 2	51
A.2	Publicações relacionadas ao capítulo 3	51
B	Teoria <i>fuzzy</i>	53
B.1	Base de regras <i>fuzzy</i>	53
B.1.1	<i>Fuzzyficação</i>	53
B.1.2	Inferência	53
B.1.3	<i>Desfuzzyficação</i>	54
B.1.4	Teoria de possibilidade	54
B.1.5	Número <i>fuzzy</i>	55
B.2	Controlador <i>fuzzy</i>	56
C	Gerência de redes por políticas	59
C.1	Gerência de redes	59
C.1.1	Políticas	60
C.1.2	Arquitetura de gerenciamento por políticas	60
C.1.3	Ponto de decisão de políticas	61
C.1.4	Ponto de instalação de políticas	61
C.1.5	Repositório de políticas	61
D	Arquivos XML utilizados pelo módulo	63
D.1	entrada01.xml	63
D.2	entrada02.xml	63
D.3	saida01.xml	64

Lista de Figuras

2.1	Arquitetura do sistema de gestão de QoS.	6
2.2	Diagrama de blocos do módulo controlador <i>fuzzy</i>	9
2.3	Estrutura da política quando há condição <i>fuzzy</i>	11
2.4	Função triangular.	12
2.5	Função de pertinência da variável lingüística Ocupação do <i>link</i> (entrada).	12
2.6	Função de pertinência da variável lingüística Perda de Pacotes fim-a-fim (entrada).	13
2.7	Função de pertinência da variável lingüística Valor Atraso máximo (entrada).	13
2.8	Função de pertinência da variável lingüística Período do dia (entrada).	14
2.9	Função de pertinência para saída <i>crisp</i> , Largura de banda a ser ajustada (BA) por meio do Corretor <i>fuzzy</i> de Banda.	16
3.1	Modelo de rede de comunicação de dados simétrica com etiquetas não- <i>fuzzy</i>	27
3.2	Modelo de rede de comunicação de dados simétrica com rótulos <i>fuzzy</i>	29
3.3	Arquitetura do controlador nebuloso de admissão de conexão.	33
3.4	Funções de pertinência para a variável de entrada - Ocupação do enlace.	35
3.5	Funções de pertinência para a variável de entrada - Requisição da aplicação.	36
3.6	Funções de pertinência para a variável de saída - Resposta à solicitação.	37
3.7	Topologia usada na simulação.	39
3.8	Número de pacotes perdidos em função do tempo para a taxa média de chegada (conexões / s): (a) $l = 1$	40
3.9	Número de pacotes perdidos em função do tempo para a taxa média de chegada (conexões / s): (b) $l = 3$	41
3.10	Número de pacotes perdidos em função do tempo para a taxa médias de chegada (conexões / s): (c) $l = 5$	41
3.11	Relação entre número total de pacotes perdidos e número total de pacotes gerados em função da taxa média de chegada das conexões.	42
3.12	Percentual de conexões bloqueadas em função da taxa média de chegada das conexões.	43
B.1	Número <i>fuzzy</i>	56
C.1	Arquitetura de gerência por políticas em três camadas.	60

Lista de Tabelas

1.1	Tipos de dados de rede sensíveis a parâmetros de QoS.	2
2.1	Função de Pertinência para entrada <i>fuzzy</i> , Ocupação do <i>link</i> - (OL).	12
2.2	Função de Pertinência para entrada <i>fuzzy</i> , Taxa de Perda de Pacotes fim-a-fim - (PP).	13
2.3	Função de Pertinência para entrada <i>fuzzy</i> , Valor Atraso máximo - (AM).	14
2.4	Função de Pertinência para entrada <i>fuzzy</i> , Período do dia - PD.	14
2.5	Função de pertinência para saída <i>crisp</i> , Largura de banda a ser ajustada - BA por meio do Corretor <i>fuzzy</i> de Banda.	16
3.1	Informações sobre os valores não <i>fuzzy</i> da rede de comunicação de dados (Figura 3.1).	28
3.2	Caminhos de <i>s</i> até <i>d</i> com tempo fim-a-fim e menor custo, restrição de tempo não <i>fuzzy</i> = 10.	28
3.3	Informações sobre os valores <i>fuzzy</i> da rede de comunicação de dados (Figura 3.2).	29
3.4	Caminhos de <i>s</i> até <i>d</i> com tempo fim-a-fim e menor custo, restrição de tempo Fuzzy = (5, 10, 20).	30
3.5	Caminhos de <i>s</i> até <i>d</i> com tempo fim-a-fim e menor custo, restrição de tempo não <i>fuzzy</i> = 60.	30
3.6	Caminhos de <i>s</i> até <i>d</i> com tempo fim-a-fim e menor custo, restrição de tempo = (50, 60, 70).	31
3.7	Resultados da comparação de performance entre os algoritmos AJTnF e AJTF.	31
3.8	Funções de pertinência para variáveis de entrada - Ocupação do enlace.	34
3.9	Funções de pertinência para variáveis de entrada - requisição da aplicação.	35
3.10	Funções de pertinência para variáveis de saída - resposta à solicitação.	36
3.11	Apresentação das regras de inferência.	38

Glossário

AAA	- <i>Administration, Authorization, and Authentication</i>
AJTF	- Algoritmo Janela de Tempo Fuzzy
AJTnF	- Algoritmo Janela de Tempo não Fuzzy
BPFBR	- Bloqueio Parcial Fuzzy com Banda Residual
CAC	- <i>Connection Admission Control</i>
CD	- Controlador Difuso
CLI	- <i>Command Line Interface</i>
COPS	- <i>Common Open Policy Service</i>
CORBA	- <i>Common Object Request Broker Architecture</i>
CPU	- <i>Central Processing Unit</i>
CR-LDP	- <i>Constraint-based LDP</i>
DEN-ng	- <i>Directory Enabled Network - Next Generation</i>
DiffServ	- <i>Differentiated Services</i>
DVD	- <i>Digital Versatile Disc</i>
FINEP	- Financiadora de Estudos e Projetos
FIFO	- <i>First-In-First-Out</i>
IHC	- Interação Homem-Computador
IP	- <i>Internet Protocol</i>
INPI	- Instituto Nacional da Propriedade Industrial
ISO	- <i>International Standard Organization</i>
ISP	- <i>Internet Service Providers</i>
LAN	- <i>Local Area Network</i>
LDAP	- <i>Lightweight Directory Access Protocol</i>
LDP	- <i>Label Distribution Protocol</i>
LER	- <i>Label Edge Router</i>
LSP	- <i>Label Switched Path</i>
LSR	- <i>Label Switch Router</i>
MNS	- <i>MPLS Network Simulator</i>
MPLS	- <i>Multi-Protocol Label Switching</i>
ms	- milissegundo
NS	- <i>Network Simulator</i>
OSI	- <i>Open System Interconnection</i>

PBNM	- <i>Policy-based Network Management</i>
PCIM	- <i>Policy Core Information Mode</i>
PDP	- <i>Policy Decision Point</i>
QoS	- <i>Quality of Service</i>
QoSM	- <i>QoS Management</i>
RMON	- <i>Remote Monitoring Network</i>
SN	- <i>Shuffle Net</i>
SNMP	- <i>Simple Network Management Protocol</i>
SQL	- <i>Structured Query Language</i>
Tags	- Comando utilizado para definir a formatação de um documento
TFTP	- <i>Trivial File Transfer Protocol</i>
UDP	- <i>User Datagram Protocol</i>
VPN	- <i>Virtual Private Network</i>
WAN	- <i>Wide Area Network</i>
WWW	- <i>World Wide Web</i>
XML	- <i>eXtensible Markup Language</i>

Capítulo 1

Introdução

O texto apresentado a seguir tem o propósito de delinear o problema tratado neste trabalho e justificar a sua importância no contexto da Tecnologia da Informação.

1.1 Contextualização

Nos últimos anos vem ocorrendo um grande desenvolvimento e uma ampla disseminação das aplicações em rede de comunicação, como, por exemplo, áudio e vídeo. Novas aplicações de redes multimídia - vídeo de entretenimento, voz sobre IP (*Internet Protocol*), rádio por Internet, páginas www (*World Wide Web*) multimídia, teleconferências, jogos interativo, mundos virtuais, aprendizagem à distância e dentre outras aplicações - são anunciadas continuamente [1]. Ao tentar utilizar um serviço dessa rede, os seus usuários esperam que ela esteja sempre disponível e querem sempre ter a sensação de respostas instantâneas e de disponibilidade plena dos serviços. Falhas, indisponibilidades freqüentes e demora nas respostas provocam frustração e irritação. Os grandes ambientes corporativos também passaram a depender de forma muito mais intensa dos serviços fornecidos pelas redes. Devido a esse crescimento e dependência das redes de computadores, os provedores de serviços têm dado maior importância a alguns quesitos, tal como, confiança, eficiência, qualidade de serviço, engenharia de tráfego e roteamento baseado em restrições, devido às necessidades de conseguir e manter novas chamadas.

Os provedores de redes estão cada vez mais preocupados em analisar, de forma crítica, qualquer aspecto que possa estar relacionado ao ambiente operacional do seu *backbone*. Eles estão procurando obter oportunidades de crescimento e otimização de recursos com mínimo custo para atender os níveis mínimos e desejáveis dos serviços solicitados com relação a: atrasos máximos (de transporte, e para estabelecer/encerrar a conexão), *jitter*, capacidade da conexão (banda), confiabilidade (de transporte, e para estabelecer/encerrar a conexão), prioridade em relação às demais conexões, proteção (contra acesso não autorizado ao tráfego que flui pela conexão), distância e perda de pacotes. Assim, garante-se o contrato no caminho entre origem e destino do tráfego de pacotes na rede (chamadas ou demandas).

A eficiência do roteamento é um componente chave do sucesso ou da falha final de uma rede de telecomunicações. Muitos pedidos de conexões podem ser aceitos, mesmo sendo a rede incapaz de suportar o atendimento desses pedidos. Isto pode resultar em uma significativa

deterioração do desempenho. Entretanto, se utilizada a política de qualidade de serviço pode-se providenciar um algoritmo capaz de identificar tais caminhos que possam ou não atender os pedidos de conexão os quais a rede suporte.

O Roteamento Baseado em Restrições tem sido tema de diversas pesquisas [2], [3]. O Problema de Caminho Mínimo com Janela de Tempo (PCMJT) e alguns pertencentes à classe *NP-hard* [4] e [5] são considerados de difícil solução. O estado da arte de roteamento em Telecomunicações está sendo a otimização nas procuras do melhor caminho de uma rede e do tempo necessário para percorrê-lo.

O roteamento baseado na qualidade de serviço (QoS - *Quality of Service*) pode ser classificado em quatro classes principais: otimização do enlace, restrição no enlace e otimização do caminho [6]. Os roteamentos, com foco no enlace, usam funções *custo* que estimam a ocupação da banda ou do *buffer* [7] e [8]. Os roteamentos que consideram o caminho usam métricas aditivas, tais como o tempo de transferência.

Devido à natureza das variáveis envolvidas no problema de otimização de rede, optamos pela lógica *fuzzy*, por ser a ferramenta existente mais indicada no tratamento desse tipo de variáveis. Várias aplicações de rede exigem o mínimo de QoS com algumas solicitações, conforme os dados ilustrados na Tabela 1.1, extraída de [9].

Tipo do Tráfico	Sensibilidade			
	Largura de banda	Perda de pacotes	Atraso	Jitter
Voz	Muito Baixa	Média	Alta	Alta
E-commerce	Baixa	Alta	Alta	Baixa
Transações	Baixa	Alta	Alta	Baixa
E-mail	Baixa	Alta	Baixa	Baixa
Telnet	Baixa	Alta	Média	Baixa
WWW aleatório	Baixa	Média	Média	Baixa
WWW importante	Média	Alta	Alta	Baixa
Transferência de Arquivos	Alta	Média	Baixa	Baixa
Vídeo conferência	Alta	Média	Alta	Alta
Multicasting	Alta	Alta	Alta	Alta

Tabela 1.1: Tipos de dados de rede sensíveis a parâmetros de QoS.

É difícil definir precisamente qualidade de serviço. Em geral, assume significados diferentes para pessoas distintas. Para a ISO/OSI (*International Standard Organization's Open System Interconnect*), QoS é definida como o efeito coletivo do desempenho de um serviço, o qual determina o grau de satisfação de um usuário do serviço [10]. Essa definição é bastante genérica e deve ser melhor definida para o problema específico que se deseja tratar. No caso de aplicações multimídia, alguns parâmetros de QoS podem possuir um componente subjetivo já que a qualidade de áudio e vídeo está relacionada com a percepção dos usuários, que é uma medida variável, por exemplo, atraso fim-a-fim baixo e alta perda de pacotes.

Para lidar com incertezas não-estatísticas, como subjetividade, foi introduzida pelo Professor Zadeh [11] a teoria dos conjuntos *fuzzy*. Um conjunto *fuzzy* é uma generalização da teoria dos

conjuntos tradicional para resolver os paradoxos gerados à partir da classificação “verdadeiro ou falso” da lógica clássica. Tradicionalmente, uma proposição lógica tem dois extremos: ou “completamente verdadeiro” ou “completamente falso”. Entretanto, na lógica *fuzzy*, uma premissa varia em grau de verdade de 0 a 1, o que leva a ser parcialmente verdadeira ou parcialmente falsa, possuindo ainda a opção de modelar termos e expressões lingüísticas, além de seu uso em raciocínio aproximado. A aplicabilidade da lógica *fuzzy* deriva da sua capacidade em inferir conclusões e gerar respostas baseadas em informações vagas, ambíguas e qualitativamente incompletas e imprecisas. Neste aspecto, os sistemas de base *fuzzy* operam de forma semelhante à dos humanos. A lógica das rotinas de programação é representada de maneira muito simples e natural, levando à construção de sistemas compreensíveis e de fácil manutenção.

1.2 Proposta de pesquisa

1.2.1 Definição do problema

Como oferecer e garantir QoS em redes de computadores por meio de roteamento e políticas de CAC (*Connection Admission Control*)?

Os atuais avanços tecnológicos na área de redes de comunicação de dados têm propiciado um crescente aumento das taxas de transmissão, tornando possível ampliar o raio de ação da QoS. Tradicionalmente, o que se tem em redes é o modelo do melhor esforço, no qual a grande preocupação reside em entregar os dados corretamente para a outra entidade participante da comunicação. Quanto às aplicações, QoS é fundamental, principalmente quando se trata de multimídia e tempo real, uma vez que estas exigem sincronismos entre as diversas mídias. Por exemplo, numa vídeoconferência o som deve estar sincronizado com a imagem, ou seja, deve haver sincronia entre as palavras e os movimentos dos lábios daqueles que aparecem falando nas imagens de mídias. Já em outras aplicações, pode-se negociar com a rede a probabilidade da queda de conexão (para admitir mais usuários na rede) em detrimento da velocidade de transmissão e do sincronismo, mantendo um certo grau de confiabilidade na conexão.

Para se falar em QoS deve-se definir parâmetros de qualidade, por meio dos quais pode-se verificar se os requisitos de QoS estão sendo atendidos para uma aplicação. Os parâmetros de QoS devem ser escolhidos de acordo com o ambiente usado, pois devido às suas características particulares, redes de clientes e redes hierárquica dos ISPs (*Internet Service Providers*) necessitam negociar parâmetros diferentes. QoS é uma maneira de se expressar as características desejadas pela aplicação por meio de um conjunto de parâmetros pré-definidos. Porém, os seus parâmetros, do ponto de vista da aplicação, são mais subjetivos (*fuzzy*), pois tratam, por exemplo, do problema de decidir se a imagem em uma vídeoconferência deve ser em cor ou em preto-e-branco, se o som deve ter qualidade de telefone ou de DVD (*Digital Versatile Disc*), se pode haver ou não interrupção da comunicação durante um *handoff*, entre outras coisas. Desse modo, deve existir um mecanismo que traduza tais desejos em parâmetros de mais baixo nível, os quais possam ser negociados pelas entidades de rede, tais como taxa de transmissão ou atraso. Portanto, QoS é algo que deve ser tratado por todo o sistema. Após a escolha dos valores dos parâmetros de QoS pela aplicação e sua posterior tradução em parâmetros que a rede possa “compreender”, deve-se efetivamente iniciar a negociação com os elementos na rede envolvidos

na comunicação para verificar se é possível prover a qualidade desejada. Se a negociação com os outros elementos de rede for bem sucedida, é preciso que os recursos necessários para se garantir a QoS sejam reservados ao longo do caminho da comunicação. Durante a comunicação deve-se monitorar também a QoS e, caso seja violada, tomar as providências necessárias. Além disso, ao longo da comunicação a aplicação deve poder renegociar parâmetros de QoS.

Como a QoS é garantida às aplicações por meio de reserva de recursos ao longo da rede, como largura de banda, *buffers* nos roteadores, ciclos de CPU (*Central Processing Unit*), entre outros, torna-se importante que não haja desperdício de recursos, pois desse modo pode-se atender a mais usuários simultaneamente. Essa racionalidade no uso dos recursos também é importante devido ao fato de que o usuário estará eventualmente pagando pela sua utilização e, portanto, deve-se procurar minimizar os custos.

1.3 Contribuições da tese

Para contribuir com a questão de como oferecer e garantir QoS em redes IP, neste trabalho são apresentadas as seguintes ferramentas, que foram desenvolvidas utilizando a lógica *fuzzy*:

1. Módulo inteligente para gerenciamento da QoS baseado em políticas [12], [13];
2. Algoritmo *fuzzy* de roteamento baseado em restrição de tempo com mínimo custo [14], [15], [16];
3. Controlador de admissão de conexão para tratar as informações com certo grau de incertezas em redes IP com MPLS (*Multi-Protocol Label Switching*) [17], [18].

1.4 Organização da tese

O restante desta tese está organizado como se segue. No Capítulo 2 é apresentada uma proposta de um módulo inteligente para gerenciamento da QoS, baseado em políticas [12], [13]. No Capítulo 3 são apresentados um algoritmo *fuzzy* que resolve o problema de roteamento baseado em restrição de tempo com mínimo custo [14], [15], [16] e um controlador de admissão de conexão, para tratar as informações com certo grau de incertezas em redes IP com MPLS [17], [18]. No Capítulo 4 são apresentadas as conclusões da tese e as propostas de continuidade da pesquisa aqui desenvolvida. No Apêndice A são apresentados os trabalhos publicados pelo autor durante o processo de doutoramento. No Apêndice B são apresentados os conceitos e definições da lógica *fuzzy* utilizados na nossa pesquisa. No Apêndice C são apresentados conceitos sobre gerência de redes por políticas utilizados na contribuição apresentada no Capítulo 2. No Apêndice D são apresentados os dados de entrada e saída no formato XML (*Extensible Markup Language*) referentes à proposta apresentada no Capítulo 2.

Capítulo 2

Gerenciamento da QoS em Redes IP com Base em Políticas

Neste capítulo é apresentado um módulo de gerenciamento da QoS que foi desenvolvido baseado em lógica *fuzzy* e em políticas (vide apêndice C). O módulo proposto configura os mecanismos da QoS na rede, por meio da criação de políticas para roteadores do tipo DiffServ (*Differentiated Services*). Além de garantir a QoS para pacotes que trafegam pelo roteador, ele pode decidir, por meio de resultados obtidos do monitoramento dos níveis, sobre a configuração da QoS em níveis apropriados para o atendimento das regras do negócio.

2.1 Contribuição

O procedimento para a geração de dados conduzidos por regras *fuzzy* foi usado no desenvolvimento de um controlador *fuzzy*. Sistemas baseados no conceito de lógica *fuzzy* têm sido utilizados com sucesso em diversas áreas, tais como: fabricação de eletrodomésticos, indústria automobilística, sistemas de auxílio à tomada de decisão e controle industrial dentre outros. Nestas aplicações, as técnicas tradicionais podem ter dificuldades de processamentos e até mesmo impossibilidade de atingir níveis desejados de eficiência. As principais causas são: as complexidades intrínsecas do sistema e a falta de agregação do conhecimento especializado de operadores humanos. Muitas vezes as rotinas são expressas de forma imprecisa, vaga e, às vezes, em situações em que o sistema possui dinâmica pouco conhecida. No entanto, na medida em que a complexidade do processo aumenta, torna-se difícil estabelecer a configuração ideal de alguns dos componentes de um sistema *fuzzy*. Desta forma, pode-se, então, aplicar ferramentas cuja função primordial é fornecer, após uma fase de aprendizagem, a solução procurada.

A contribuição deste capítulo consiste no uso da lógica *fuzzy* para implementação de um módulo Ponto de Decisão de Políticas (PDP - *Policy Decision Point*) de inteligência para tomada de decisões, a partir de um controlador difuso - CD. Para propósitos de validação da implementação foram realizadas simulações do principal elemento do módulo, responsável pelas ações na rede, baseando-se nas condições e eventos das políticas e em parâmetros que caracterizam o estado da rede. Recentes publicações [12], [13] mostram a coesão da técnica aqui proposta.

2.2 Descrição do sistema de gestão de QoS por meio de políticas

A arquitetura do sistema é composta por um conjunto de módulos que auxiliam na tomada de decisão. Estes componentes funcionais são apresentados na Figura 2.1.

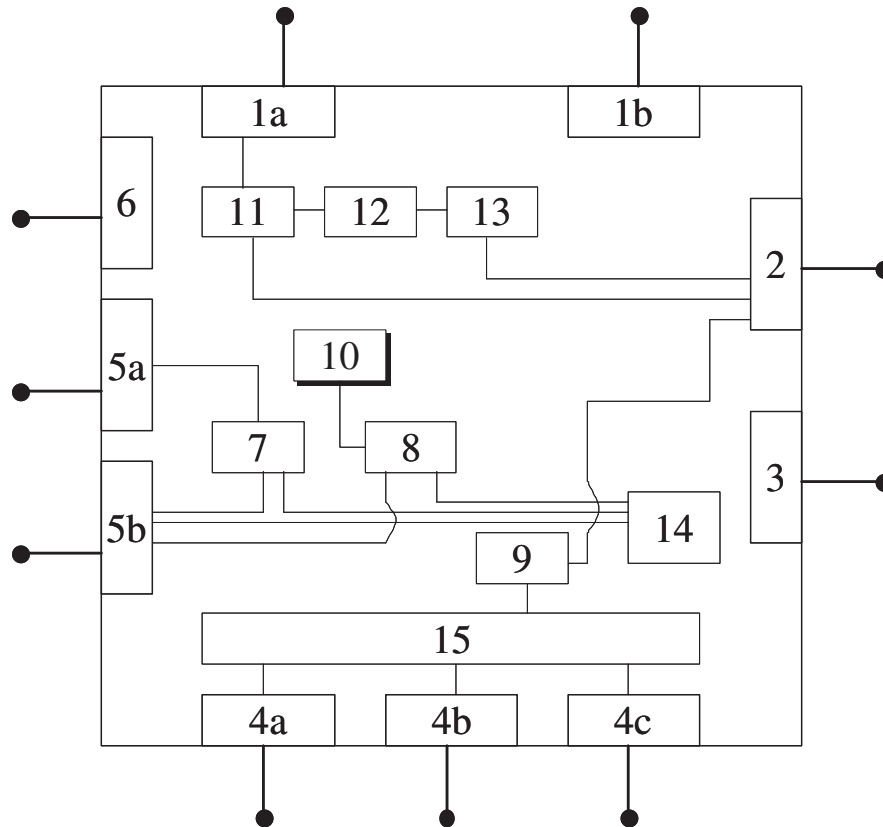


Figura 2.1: Arquitetura do sistema de gestão de QoS.

Os módulos 1 - 6 são as interfaces externas do sistema. Os módulos 11 - 15 implementam as funcionalidades de edição, compilação e detecção de conflitos entre políticas, geração de eventos, avaliação de condições e envio de ações de configuração aos devidos elementos de rede.

2.2.1 Interfaces do sistema QoSM

- Módulo 1a - Interface com IHC (Interação Homem-Computador) *client side presentation*
Esta interface permite que a edição de políticas seja desacoplada do tipo de linguagem utilizada para a implementação da IHC.
- Módulo 1b - Interface com o gerente integrador
Esta interface permite que o sistema QoSM (*QoS Manager*) opere de forma integrada com os demais módulos de gerência, garantindo a não duplicação de funcionalidades.

- Módulo 2 - Interface com repositório de políticas

Como o modelo de informações para representação de políticas é o DEN-NG (*Directory Enabled Network - Next Generation*), as possibilidades de repositório de políticas consistem do banco de dados SQL (*Structured Query Language*) ou diretório LDAP (*Lightweight Directory Access Protocol*). Assim, a interface com o repositório depende do tipo de repositório a ser utilizado.

- Módulo 3 - Interface com repositório da máquina de estados das políticas

Idem a (2)

- Módulo 4a - Interface com roteadores WAN

Esta interface permite a comunicação com elementos de rede por meio dos protocolos de configuração SNMP (*Simple Network Management Protocol*), CLI (*Command Line Interface*) e TFTP (*Trivial File Transfer Protocol*).

- Módulo 4b - Interface com equipamento genérico

Baseada em *scripts* CLI, esta interface permite que o operador elabore *scripts* a serem executados nos equipamentos via CLI.

- Módulo 4c - Interface com *switches* LAN (*Local Area Network*) Esta interface permite a comunicação com elementos de rede por meio dos protocolos configuração SNMP, CLI e TFTP.

- Módulo 5a - Interface de coleta de estatísticas da rede

Esta interface deve permitir a coleta de estatísticas de alguns elementos da rede. Esta coleta deve ser pontual e definida pelos módulos controlador de eventos e analisador de condições. Os mecanismos utilizados para monitoração são protocolos Netflow, SNMP (Trap e Get), sistema de logs e RMON (*Remote Monitoring Network*).

- Módulo 5b - Interface com sistema AAA (*Administration, Authorization, and Authentication*)

Esta interface permite a coleta de dados do sistema de autenticação em uso na rede. Inicialmente o mecanismo suportado é o *Radius accounting*.

- Módulo 6 - Interface com Repositório topológico

Opera como uma interface LDAP ou SQL.

- Módulo 7 - Analisador e gerador de eventos

Pela análise dos descritores de eventos armazenados no repositório de políticas, este módulo configura monitores de eventos e informa ao módulo controlador da máquina de estados das políticas sobre a ocorrência de novo evento, para avaliar a condição da política em TRUE ou FALSE;

- Módulo 8 - Avaliador de condições

Ao ser informado pelo módulo controlador da máquina de estados das políticas sobre condição de avaliação para determinada política; avaliador de condições procede a avaliação da condição. Esta avaliação pode ser feita por meio de uma comparação simples ou por meio do acionamento do módulo de inteligência, que deverá usar técnicas mais avançadas para avaliação da condição em questão.

- Módulo 9 - Gerador de ações

No modelo DEN-ng, as ações são executadas para os casos de condições avaliadas como verdadeiras ou falsas. Em qualquer um destes casos, o controlador da máquina de estados das políticas inicializa o módulo gerador de ações para execução da ação. Neste caso, o módulo gerador de ações simplesmente empacota os dados relativos à ação, se necessário, adiciona informações topológicas e sobre dispositivos, encaminhando ao despachador e tradutor de ações que por sua vez encaminha a ação para os equipamentos devidos, de acordo com o escopo e objetivo da política em questão.

- Módulo 10 - Controlador *Fuzzy* proposto com contribuição em tecnologia da informação

Este módulo auxilia na avaliação de condições que necessitam de regras complexas de avaliação e está descrito na seção 2.3.

- Módulo 11 - Editor de políticas

Este módulo permite a edição de políticas via interface Web. A interface permite que os dados que serão preenchidos sobre as políticas sejam configurados de forma amigável. O editor de políticas ainda interage com o compilador e detector de conflitos entre políticas para adição ou edição de políticas existentes.

- Módulo 12 - Compilador de políticas

O compilador *Ponder* consiste de um analisador de sintaxe, um analisador de semântica e um gerador de código Java para políticas de *obligation* e *refrain*. O compilador usa uma especificação em *Ponder* e gera um código intermediário que contém os vários objetos de política na especificação. O compilador suporta um mecanismo para armazenamento dos objetos gerados em um serviço de diretórios usando LDAP

- Módulo 13 - Detector de conflitos entre políticas

Este módulo detecta conflitos entre as políticas ativadas no sistema e novas políticas a serem aplicadas

- Módulo 14 - Controlador da máquina de estados das políticas

O módulo é responsável por manter os estados das políticas atualizados no sistema e auxiliar o módulo de inteligência sempre que houver necessidade de aplicação ou substituição de políticas ativas

- Módulo - 15 Despachador e tradutor de ações

Este módulo traduz o código intermediário em um conjunto de comandos na linguagem específica do dispositivo de rede utilizado.

2.3 Funcionamento do Módulo Controlador *Fuzzy*

A estrutura básica do módulo controlador *fuzzy* é mostrada na Figura 2.2. Ela é composta principalmente por dados de entrada, regras *fuzzy* e parâmetros de saída.

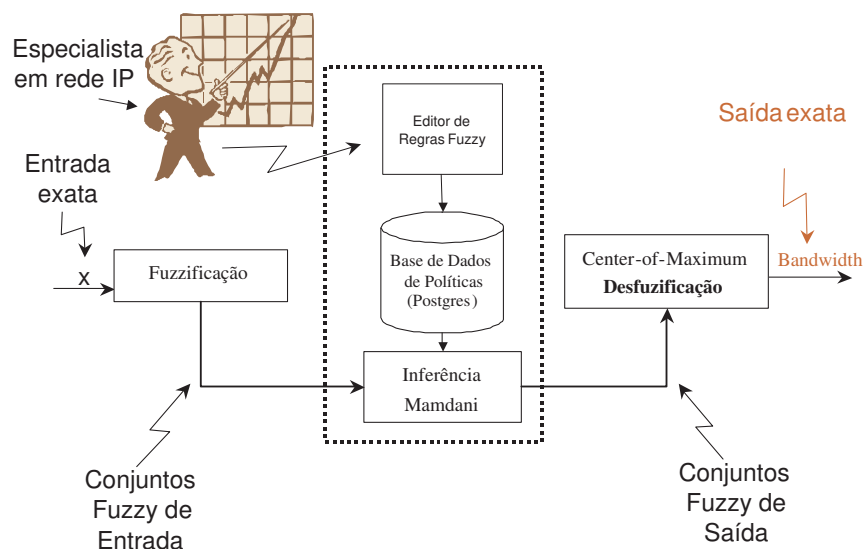


Figura 2.2: Diagrama de blocos do módulo controlador *fuzzy*.

O módulo atua na reconfiguração de um nó Diffserv em função de eventos ocorridos na rede agindo de forma semelhante a um Corretor de Banda (*Bandwidth Broker*) [19]. Baseado na análise *fuzzy* de medidas pré-estabelecidas coletadas da rede, a aplicação Corretor *fuzzy* de Banda (*Bandwidth fuzzy Broker*) determina o valor de largura de banda a ser ajustada para o nó.

A análise *fuzzy* implementa a lógica *fuzzy* (nebulosa), baseada na teoria dos Conjuntos *fuzzy*. A lógica nebulosa possibilita um tratamento das implicações lógicas seguindo regras naturais de raciocínio, analisando condições. Seus fundamentos advêm da teoria dos conjuntos nebulosos (*fuzzy sets*), que permite a manipulação de expressões verbais abstratas tais como: pouco, muito congestionado e outras. No entanto, embora seja possível modelar fenômenos por meio da lógica *fuzzy*, é necessário um processamento computacional sobre esse modelo. É na etapa de “fuzzyficação” que tais informações são convertidas em variáveis *fuzzy* para então ocorrer a formulação e execução de uma estratégia de controle. As grandezas provenientes do domínio do mundo real, captadas por monitores (dados de entrada - crisp input), sofrem essa conversão por meio da definição de um conjunto de funções linguísticas de pertinência, que descrevem a entidade no domínio de abrangência.

A definição satisfatória da quantidade e grau de superposição (0 - totalmente não-membro e 1 - totalmente membro) entre as funções de pertinência é fundamental na implementação de um controlador nebuloso. Ela influencia diretamente na fase seguinte do processo: a inferência ou lógica de tomada de decisão. Nesta fase, um conjunto de regras (base de conhecimento) do tipo IF ... THEN ... descrevem a dependência entre as variáveis lingüísticas de entrada e a de saída. Essa relação é determinada heurísticamente. A inferência consiste de duas etapas: agregação, quando se avalia a parte **IF** de cada regra, por meio dos operadores “AND” e “OR”. Esta etapa fornece o menor (MIN) ou maior (MAX) dos dois valores de pertinência comparados; a etapa de composição, pondera as diferentes conclusões das regras ativas sob os operadores “AND” e “OR”, provendo o menor ou maior dos parâmetros entre estes. Após a inferência da ação a ser tomada, necessita-se de uma tradução do valor lingüístico para a variável numérica de saída, que pode representar atividades como ajustar o limite Máximo de utilização de banda, por classes de serviços, ou acionar uma válvula. Este passo é conhecido como “*defuzzyficação*”.

2.3.1 Dados de entrada

A execução das entradas possui as seguintes etapas: Fluxos básico e alternativo que fazem uso dos seguintes dados de entrada:

1. Nome da política;
2. Arquivo XML “entrada01.xml” contendo o valor máximo de banda da classe (MAX_BW_CLASS) e modo de inferência (MODE_INFERENCIA);
3. Arquivo XML “entrada02.xml” contendo os valores iniciais das funções de pertinência do controlador *fuzzy*.
4. Medidas de entrada:
 - Ocupação do *link* - OL;
 - Perda de pacotes - PP;
 - Atraso máximo - AM;
 - Período do dia - PD.

2.3.2 Regras *fuzzy*

O conhecimento que relaciona as diversas variáveis de entrada e seus respectivos valores gera uma base de regras *fuzzy* no formato If-Then. Podem ser definidas as seguintes regras de inferência (Condição *fuzzy*), baseadas nas medidas de entrada:

1. **If** (Ocupação do *link* é Baixa) **AND** (Atraso máximo fim-a-fim é Médio) **AND** (Período do dia é à Tarde) **AND** (Perda de pacotes é Média) **Then** (Corretor *fuzzy* de Banda é Pouco positivo)
2. **If** (Ocupação do *link* é Baixa) **OR** (Atraso máximo fim-a-fim é Baixo) **AND** (Período do dia é à Tarde) **OR** (Perda de pacotes é Média) **Then** (Corretor *fuzzy* de Banda é Pouco negativo)

2.3.3 Fluxo básico

O fluxo básico consiste em:

1. Carregar os arquivos entrada01.xml e entrada02.xml;
2. Obter as medidas de entrada;
3. Recuperar, verificar e obter a regra mostrada na Figura 2.3;

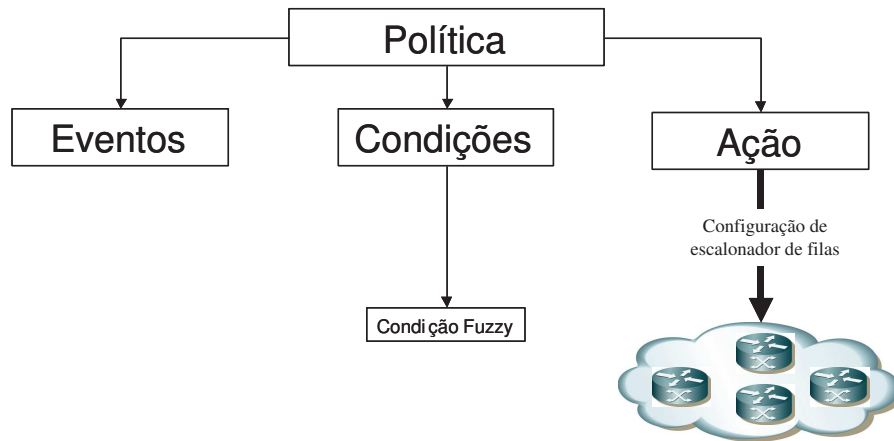


Figura 2.3: Estrutura da política quando há condição *fuzzy*.

A Figura 2.3 acima ilustra, na forma de diagrama de blocos, uma política composta por Eventos, Condições *fuzzy* e Ação. A ação à ser executada nos roteadores IP é a configuração de escalonador de filas [20].

4. Calcular grau de pertinência (α corte).

As funções de pertinência selecionadas são contínuas do tipo triangulares, equação (2.1) e figura 2.4, em razão de possuírem codificação simples, eficiente e apresentar os mesmos resultados de outras funções [21], [22], [23].

$$\text{trimf}(x; a, b, c) = \begin{cases} 0, & \text{if } x \leq a \\ \frac{x-a}{b-a}, & \text{if } x \in (a, b) \\ \frac{c-x}{c-b}, & \text{if } x \in [b, c) \\ 0, & \text{if } x \geq c \end{cases} \quad (2.1)$$

onde x é o dado de entrada, a é o limitante inferior, b é o valor modal e c é o limitante superior. Neste capítulo, a partir deste ponto, todas as variáveis com índices a , b e c representam limitante inferior, valor modal e limitante superior das suas funções de pertinência, respectivamente.

Fazendo uso da equação (2.1):

- (a) É calculado o grau de pertinência de cada função do conjunto *fuzzy* “Ocupação do *link*”, representada pela Figura 2.5 e Tabela 2.1.

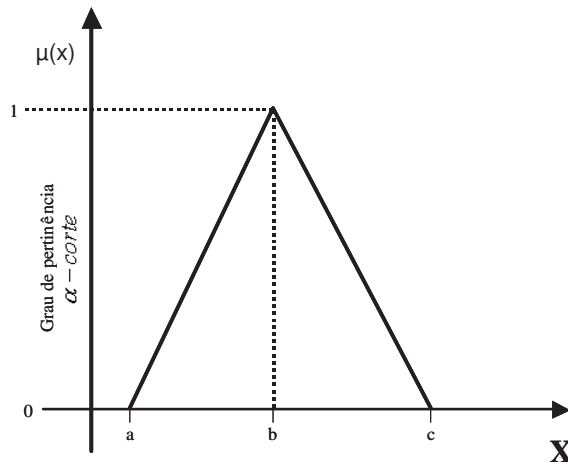
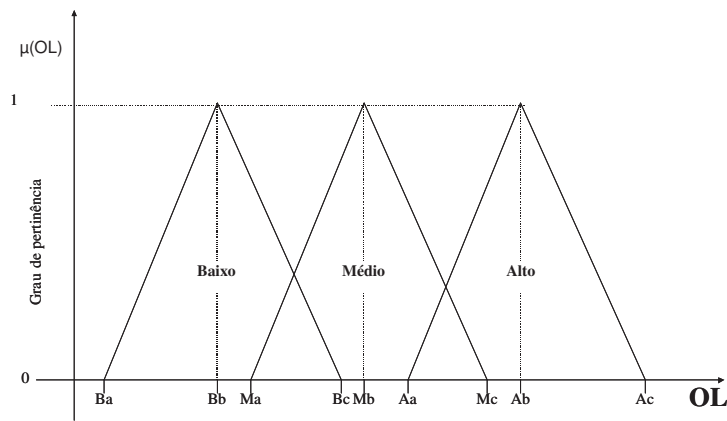


Figura 2.4: Função triangular.

Figura 2.5: Função de pertinência da variável lingüística Ocupação do *link* (entrada).

Variável lingüística	Tipo da função	Parâmetros da função	Valores
Baixo	trimf	$(x; B_a, B_b, B_c)$	$(x; 0, 32, 96)$
Médio	trimf	$(x; M_a, M_b, M_c)$	$(x; 64, 128, 192)$
Alto	trimf	$(x; A_a, A_b, A_c)$	$(x; 160, 224, 256)$

Tabela 2.1: Função de Pertinência para entrada *fuzzy*, Ocupação do *link* - (OL).

- (b) É calculado o grau de pertinência de cada função do conjunto *fuzzy* “Perda de pacotes”, representada pela Figura 2.6 e Tabela 2.2.
- (c) É calculado o grau de pertinência de cada função do conjunto *fuzzy* “Atraso máximo”, representada pela Figura 2.7 e Tabela 2.3.
- (d) É calculado o grau de pertinência de cada função do conjunto *fuzzy* “Período do dia”, representada pela Figura 2.8 e Tabela 2.4.

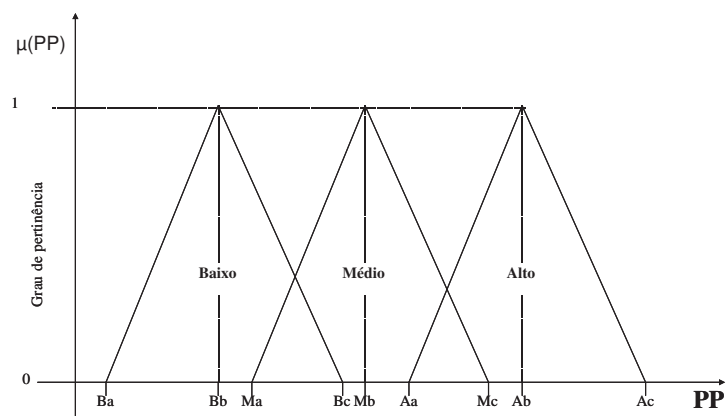


Figura 2.6: Função de pertinência da variável linguística Perda de Pacotes fim-a-fim (entrada).

<i>Variável linguística</i>	<i>Tipo da função</i>	<i>Parâmetros da função</i>	<i>Valores</i>
Baixo	trimf	$(x; B_a, B_b, B_c)$	$(x; 2, 6, 10)$
Médio	trimf	$(x; M_a, M_b, M_c)$	$(x; 7, 14, 21)$
Alto	trimf	$(x; A_a, A_b, A_c)$	$(x; 17, 25, 33)$

Tabela 2.2: Função de Pertinência para entrada *fuzzy*, Taxa de Perda de Pacotes fim-a-fim - (PP).

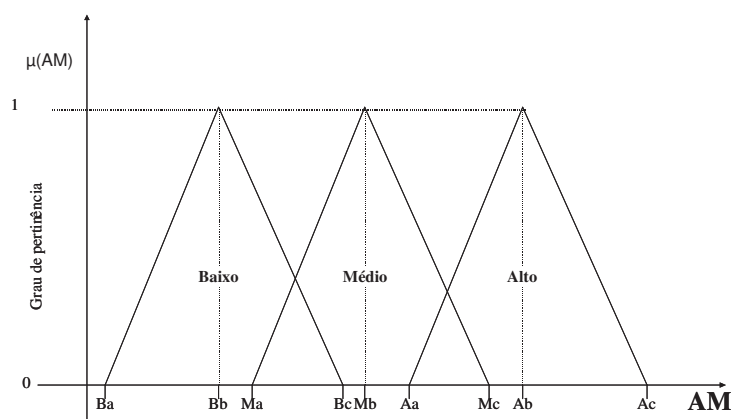


Figura 2.7: Função de pertinência da variável linguística Valor Atraso máximo (entrada).

Os valores atribuídos aos parâmetros das funções de pertinência, foram extraídos de [24], [25], [26], [9] e são armazenados no arquivo entrada02.xml.

<i>Variável lingüística</i>	<i>Tipo da função</i>	<i>Parâmetros da função</i>	<i>Valores</i>
Baixo	trimf	$(x; B_a, B_b, B_c)$	$(x; 0, 75, 150)$
Médio	trimf	$(x; M_a, M_b, M_c)$	$(x; 100, 175, 250)$
Alto	trimf	$(x; A_a, A_b, A_c)$	$(x; 200, 275, 300)$

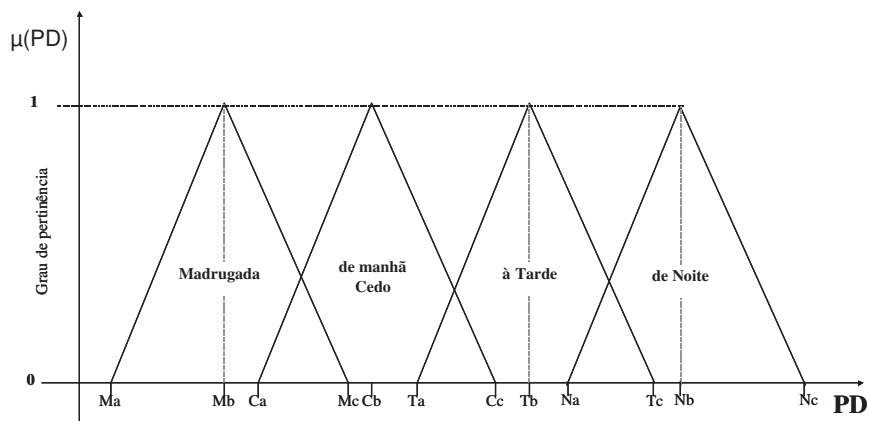
Tabela 2.3: Função de Pertinência para entrada *fuzzy*, Valor Atraso máximo - (AM).

Figura 2.8: Função de pertinência da variável lingüística Período do dia (entrada).

<i>Variável lingüística</i>	<i>Tipo da função</i>	<i>Parâmetros da função</i>	<i>Valores</i>
Madrugada	trimf	$(x; M_a, M_b, M_c)$	$(x; 00 : 00, 03 : 00, 06 : 00)$
de manhã Cedo	trimf	$(x; C_a, C_b, C_c)$	$(x; 05 : 00, 09 : 00, 12 : 00)$
à Tarde	trimf	$(x; T_a, T_b, T_c)$	$(x; 11 : 00, 15 : 00, 18 : 00)$
de Noite	trimf	$(x; N_a, N_b, N_c)$	$(x; 17 : 00, 21 : 00, 24 : 00)$

Tabela 2.4: Função de Pertinência para entrada *fuzzy*, Período do dia - PD.

2.3.4 Fluxos alternativos

No processo de tratamento dos dados de entrada, o fluxo alternativo acontece quando:

1. É identificado mais de uma regra na condição *fuzzy*;
2. O módulo não identifica uma função de pertinência com resultado maior que zero;
3. Não há regra ativada na Condição *fuzzy*;
4. Mais de uma regra ativada para a condição *fuzzy* que está sendo analisada;
5. O módulo obtém valor MAX para a variável MODE_INFERENCE;
6. O resultado do cálculo de Defuzzificação (Valor de banda) é maior que o valor MAX_BW_CLASS recuperado do arquivo `entrada01.xml`;
7. O resultado do cálculo de defuzzificação é menor que zero;
8. O módulo não consegue carregar o arquivo `entrada02.xml`;
9. O módulo não consegue obter o valor MODE_INFERENCE do arquivo `entrada01.xml`;
10. O módulo não consegue obter o valor MAX_BW_CLASS do arquivo `entrada01.xml`;
11. Falha na recuperação da Condição *fuzzy* do repositório de políticas;
12. Não possuem valores as medidas de entrada;
13. O módulo não encontra no arquivo `entrada02.xml` os valores esperados para os conjuntos *fuzzy*;
14. Ocorre falhas no registro das mensagens de erro no arquivo de log.

2.3.5 Parâmetros de saída

Os resultados de saída podem agrupar em:

- Resultado da *defuzzificação*: valor de largura de banda (decremento ou incremento);
- Mensagem de erro para o arquivo de log do módulo.

Os resultados da *defuzzificação* compõem o seguinte fluxo de saída:

1. O sistema obtém do arquivo `entrada02.xml` carregado no passo 3 da seção 2.3.1 do dados de entrada, os parâmetros para cada função de pertinência do conjunto *fuzzy* “Largura de banda a ser ajustada”, e os inicializa com seus respectivos valores, apresentados na Figura 2.9 e Tabela 2.5.

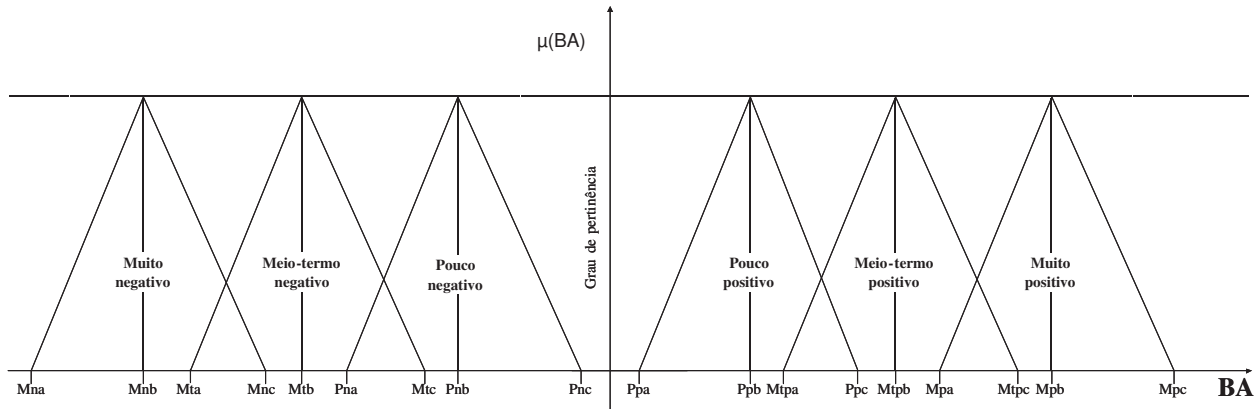


Figura 2.9: Função de pertinência para saída *crisp*, Largura de banda a ser ajustada (BA) por meio do Corretor *fuzzy* de Banda.

<i>Variável lingüística</i>	<i>Tipo da função</i>	<i>Parâmetros da função</i>	<i>Valores</i>
Pouco positivo - Pp	trimf	$(x; Pp_a, Pp_b, Pp_c)$	$(x; 2, 5, 8)$
meio-termo positivo - Mtp	trimf	$(x; Mtp_a, Mtp_b, Mtp_c)$	$(x; 6, 9, 12)$
Muito positivo - Mp	trimf	$(x; Mp_a, Mp_b, Mp_c)$	$(x; 10, 13, 16)$
Pouco negativo - Pn	trimf	$(x; Pn_a, Pn_b, Pn_c)$	$(x; -2, -5, -8)$
meio-termo negativo - Mt	trimf	$(x; Mt_a, Mt_b, Mt_c)$	$(x; -6, -9, -12)$
Muito negativo - Mn	trimf	$(x; Mn_a, Mn_b, Mn_c)$	$(x; -10, -13, -16)$

Tabela 2.5: Função de pertinência para saída *crisp*, Largura de banda a ser ajustada - BA por meio do Corretor *fuzzy* de Banda.

2. Através do Centro-do-Máximo é gerada a saída como valor numérico [27], [28]. A saída discreta é a média ponderada dos máximos, cujos pesos são os resultados da inferência. O valor da *defuzzificação* é determinado por meio da equação (2.2).

$$u = \frac{\sum_{i=1}^N u_i \sum_{k=1}^n \mu_{O,k}(u_i)}{\sum_{i=1}^N \sum_{k=1}^n \mu_{O,k}(u_i)} \quad (2.2)$$

onde $\mu_{O,k}$ e u_i indicam os pontos em que ocorrem máximos (alturas) das funções de pertinência de saída, e os valores resultantes da inferência de cada regra, respectivamente, N é o número de funções de pertinência de saída. A escolha deste método foi por ser de fácil implementação e os resultados obtidos não justificaram a complexidade adicionada quando comparados com os demais métodos [27].

3. O sistema verifica se o resultado do cálculo de defuzzificação obtido no passo 5 da Subseção 2.3.3, do fluxo básico, é maior que zero;
4. O sistema verifica se o resultado do cálculo de defuzzificação, é menor ou igual ao MAX_BW_CLASS (em porcentagem), obtido no passo 3 da subseção 2.3.1;

5. O sistema gera o arquivo saida01.xml com o resultado do cálculo de defuzzificação e o encaminha para um Sistema de Gerência de redes;
6. O arquivo saida01.xml é então usado como dado de entrada de um Sistema de Gerência de redes para gestão de Falhas, Configuração e Desempenho. Este sistema, permite:
 - Visualização topológica geral da rede supervisionada e de forma hierarquizada;
 - Navegabilidade (top-down e bottom-up) até o elemento de rede;
 - Visualização por domínios tecnológicos e administrativos, consulta às configurações, monitora falhas e parâmetros de desempenho da rede e disponibiliza informações por meio de relatórios customizados;
 - Integração com outros sistemas de suporte de operação;
 - Suporte de múltiplos domínios tecnológicos;
 - Suporte de protocolos para comunicação (SNMP, CLI e outros);
 - Com recursos gerenciados e protocolos de comunicação (XML, CORBA e outros) com outros sistemas.

2.4 Resultados

Esta pesquisa foi financiada pela FINEP (Financiadora de Estudos e Projetos) com o intuito de desenvolver um programa que faz gerenciamento de qualidade de Serviço em equipamentos roteadores de forma automática de acordo com a mudança do estado da rede e das políticas de QoS criadas no sistema. O título do programa é QoSM - QoS Manager, foi criado em 26 de janeiro de 2005 na linguagem Java e foi solicitado ao INPI (Instituto Nacional da Propriedade Industrial) o seu registro, Classificação do Campo de Aplicação TC02 e Classificação do Tipo de Programa CD05, Nome comercial QoSM, Versão a ser protegida 1.0 - 018060064943.

Capítulo 3

Roteamento Baseado em Restrições

3.1 Contribuição

Neste capítulo, é proposto um algoritmo de roteamento que considera como métricas o custo nos enlaces e restrições no caminho. O algoritmo encontra um conjunto de caminhos que atende as exigências mínimas solicitadas (banda e tempo) entre um par de nós, origem-destino, de redes de telecomunicações. Recentes publicações do autor [14], [15], [16], [17], [18], mostram a coerência das técnicas aqui propostas.

3.2 Formulação do problema de roteamento

Um dos problemas a ser resolvido na camada de rede do modelo OSI (*Open Systems Interconnection*) é a escolha do caminho que os pacotes devem seguir para interconexão das estações de origem e de destino [6], [1]. Nas redes que transportam informações do tipo datagramas, a decisão de roteamento é tomada para cada pacote. Para rede do tipo circuito virtual, o caminho a ser seguido é definido no estabelecimento do circuito virtual, e todos os pacotes da mensagem seguem este caminho.

A eficiência do roteamento é um componente chave do sucesso de uma rede de telecomunicações. Muitos pedidos de conexões podem ser aceitos, mesmo sendo a rede incapaz de suportar o atendimento desses pedidos. Isto pode resultar na deterioração do desempenho. Entretanto, se forem utilizados mecanismos de controle de Admissão de Conexões - CAC (*Connection Admission Control*) e política de qualidade de serviço pode-se prover um algoritmo capaz de identificar caminhos capazes de atender os pedidos de conexão.

3.2.1 Elementos dos algoritmos

Etiquetas

Para resolver o problema de caminho de custo mínimo com janela de tempo, as rotas têm outro atributo importante que consiste no tempo de chegada até o nó i . Para tratar simultaneamente as variáveis de tempo t_i e as variáveis de custo c_i associadas a cada nó i ,

utiliza-se uma etiqueta (c_i, t_i) , para rotular cada nó i . Uma etiqueta (c_i, t_i) é atribuída ao nó i se, e somente se, existir um caminho de custo c_i , saindo da origem s e chegando ao nó i , no tempo t_i .

Relação de ordem das etiquetas

Define-se a relação de ordem parcial $\prec$ para o nó i como

$$(c_i^1, t_i^1) \prec (c_i^2, t_i^2) \Leftrightarrow (t_i^1 \leq t_i^2) \text{ e } (c_i^1 \leq c_i^2).$$

Uma etiqueta pertencente ao conjunto irredutível Q_i é denominada *etiqueta mínima* se não existir outra inferior a ela, segundo a relação de ordem parcial. As etiquetas mínimas guardam os caminhos da origem s ao nó i . Assim, pode-se reduzir o conjunto de etiquetas de cada nó, retirando aquelas que não são mínimas e, conservando o conjunto das etiquetas irredutíveis dentro do tempo de atendimento permitido. O conjunto irredutível associado ao nó i é ordenado em função da variável de tempo t_i , estritamente crescente, e os custos c_i em ordem decrescente.

Teoria de possibilidade

Seja $G = (N, M)$ um grafo no qual N denota o conjunto de nós aos quais são associadas janelas de tempo *fuzzy* e M o conjunto de arcos. Cada arco $(i, j) \in M$ tem um tempo *fuzzy* $\tilde{t}_{ij}$ de transferência e um custo *fuzzy* $\tilde{c}_{ij}$ associados $\in \mathbb{R}^2$. Sejam $\tilde{t}^1, \tilde{t}^2; \tilde{c}^1, \tilde{c}^2$; para $\tilde{t}^1 \neq \tilde{t}^2$ e $\tilde{c}^1 \neq \tilde{c}^2$ dois pares de números *fuzzy*. Os números *fuzzy* $\tilde{t}^1$ e $\tilde{c}^1$ têm grau de possibilidade menor do que $\tilde{t}^2$ e $\tilde{c}^2$, respectivamente, conforme a expressão 3.1:

$$Poss(\tilde{t}^1 \leq \tilde{t}^2) = \sup_{(u \leq v)} \{\min\{\mu_{\tilde{t}^1}(u), \mu_{\tilde{t}^2}(v)\}\} \quad (3.1)$$

onde $\mu_{\tilde{t}^1}$ e $\mu_{\tilde{t}^2}$ são as funções de pertinência de $\tilde{t}^1$ e $\tilde{t}^2$, respectivamente. Mais detalhes da teoria de possibilidade são apresentados no Apêndice B.

3.2.2 Formulação do problema e algoritmo de Ford-Bellman-Moore

Ford-Bellman-Moore [29] formula o problema de caminho mínimo (sem janela de tempo) e determina para cada nó uma etiqueta c_i que representa o custo de ir da origem s para o nó i , utilizando um dado caminho.

Conforme a formulação de Ford-Bellman-Moore, seja $F(i) = \{j : (i, j) \in M\}$ o conjunto de nós sucessores dos nós $i \in L$. A árvore de caminho mínimo, saindo da origem s para todos os outros nós da rede, é obtida associando-se a cada nó uma etiqueta melhorada a cada iteração do algoritmo. O método das etiquetas consiste na escolha de um nó i , pertencente ao conjunto L . Em seguida se retira o nó i do conjunto L e tenta-se melhorar todos os sucessores do nó i . Os nós melhorados são inseridos no conjunto L . Assim, quando o conjunto L for vazio, a etiqueta c_i , associada ao nó i , representa o custo do caminho mínimo, saindo da origem s e terminando no nó i .

Decorrente de sua formulação Ford-Bellman-Moore estabelece o seguinte algoritmo. Na aplicação do algoritmo as etiquetas são melhoradas recursivamente até que os valores ótimos sejam obtidos.

Passo 1: Iniciação

- O custo na origem s : $c_s \leftarrow 0$
- O custo nos demais nós: $c_i \leftarrow \infty, \quad i \neq s$
- $L \leftarrow \{s\}$

Passo 2: Exploração de $F(i)$

- Escolha um nó $i \in L$
- $\forall j \in F(i)$ tal que $c_j > c_i + c_{ij}$

$$c_j \leftarrow c_i + c_{ij}$$

$$L \leftarrow L \cup \{j\}$$

Passo 3: Resolução do conjunto L

- $L \leftarrow L - \{i\}$
- Se $L = \emptyset$

então FIM.

Caso contrário, retorne ao Passo 2.

Neste algoritmo é associada a cada nó uma etiqueta progressivamente melhorada. Quando se acrescenta ao problema as restrições de horários (janela de tempo) não é mais possível conservar unicamente a etiqueta melhorada para cada nó, porque o par (c_i, t_i) não tem ordem total no $\mathbb{R}^2$. Então, é necessário preservar um conjunto de etiquetas (c_i^α, t_i^α) para $\alpha \geq 1$, onde α armazena os possíveis caminhos associados às suas respectivas etiquetas. Este conjunto é finito, porque a quantidade de nós é finita, determinando um número finito de caminhos possíveis, saindo da origem s e chegando ao nó i . Cada caminho produz uma etiqueta para cada nó i que será armazenada em uma lista Q_i , que representa um conjunto irredutível de etiquetas.

Dois algoritmos de roteamento vêm sendo explorados: algoritmo de Dijkstra e de Ford-Bellman-Moore [29]. Enquanto o algoritmo de Dijkstra tem natureza sequencial, o algoritmo Ford-Bellman-Moore é flexível a uma implementação distribuída [30]. Para o pior caso, a complexidade do algoritmo de Dijkstra é $O(M \log N)$ e para o algoritmo de Ford-Bellman-Moore $O(HM)$, onde N denota o conjunto de nós, M , o conjunto de arcos e H é o número máximo de interações do algoritmo, para $H \gg N - 1$ (teoricamente), e $H < N$ na prática. Logo, existe uma menor complexidade, usando o algoritmo de Ford-Bellman-Moore.

Como a proposta deste capítulo é um algoritmo de roteamento, considerando conjuntamente o custo nos enlaces (*least-cost*) e restrições no caminho (*delay-constrained*), se torna necessário fazer o tratamento do problema de roteamento, considerando mais de uma restrição.

3.3 Problema de roteamento baseado em restrições

3.3.1 Problema de roteamento sem usar a lógica *fuzzy*

Com a finalidade de formular o problema de roteamento, considere $G = (N, M)$ um grafo no qual N denota o conjunto de nós aos quais são associados janelas de tempo, e M , o conjunto de arcos. Cada arco $(i, j) \in M$ tem um tempo t_{ij} de transferência e um custo c_{ij} associado. O tempo t_{ij} é a soma dos tempos de enfileiramento (recebimento, processamento e transmissão) e propagação do nó i para o nó j . Um arco (i, j) é considerado no caminho somente quando for possível satisfazer a restrição $(t_i + t_{ij}) \leq b_j$, sendo t_i o tempo para acessar o nó i .

O problema de roteamento pode ser formulado matematicamente como:

$$x_{ij} = \begin{cases} 1, & \text{se } (i, j) \in M \text{ é utilizado na conexão,} \\ 0, & \text{caso contrário.} \end{cases}$$

$$\begin{aligned} &\text{Minimizar} && \sum_{(i,j) \in M} c_{ij} x_{ij} \\ &\text{sujeito a} \end{aligned}$$

$$\forall j \in N, \quad \sum_{i \in N} x_{ij} - \sum_{i \in N} x_{ji} = \begin{cases} -1, & \text{para } j = s, \\ 1, & \text{para } j = d, \\ 0, & \text{caso contrário,} \end{cases} \quad (3.2)$$

$$\begin{aligned} &t_s = a_s \\ &t_j = x_{ij}(t_i + t_{ij}) + (1 - x_{ij})a_j; \quad i \in N, j \in N - \{s\}, \end{aligned} \quad (3.3)$$

$$a_i \leq t_i \leq b_i; \quad i \in N, \quad (3.4)$$

$$x_{ij} \in \{0, 1\} \quad (i, j) \in M.$$

A restrição (3.2) assegura que a conexão que sai do nó s chega ao nó d . A restrição (3.3) estabelece a compatibilidade requerida entre as rotas e os tempos, e a restrição (3.4) estabelece o intervalo de tempo em que o nó pode ser acessado.

Portanto, trata-se de encontrar um caminho de custo mínimo entre um nó origem s e um nó destino d na rede residual, tal que satisfaça a banda de transmissão solicitada e o tempo de transferência dos pacotes entre os nós contidos nas janelas temporais de restrição $[a_i, b_i]$, especificadas para cada roteador i do caminho escolhido. Este problema é conhecido como: *path-constrained path-optimization routing or delay-constrained least-cost routing* [6].

3.3.2 Algoritmo sem usar a lógica *fuzzy*

Segundo a formulação do problema de Ford-Bellman-Moore, o algoritmo de caminho mínimo para uma rede com restrições de atraso nos nós, armazena em cada nó pares de etiqueta (c_i, t_i) , ou seja, uma rota de custo mínimo, chegando ao nó i no tempo t_i .

O conjunto irredutível associado ao nó i contém $\alpha(i)$ pares de etiquetas: (c_i^α, t_i^α) , onde $\alpha = 1, 2, \dots, \alpha(i)$, com

$$a_i \leq t_i^1 < t_i^2 < \dots < t_i^{\alpha(i)} \leq b_i \text{ e } c_i^1 > c_i^2 > \dots > c_i^{\alpha(i)}.$$

Seja $Q_i = (c_i^\alpha, t_i^\alpha)$, $\alpha = 1, 2, \dots, \alpha(i)$, um conjunto irredutível de etiquetas associadas ao nó i . Assim, unindo o arco (i, j) aos caminhos que saem da origem s e chegam em i , obtemos o conjunto Q_j associado ao nó j por meio da seguinte aplicação:

$$F_{ij}(Q_i) = \cup_{\alpha} f_{ij}\{(c_i^\alpha, t_i^\alpha)\}$$

onde

$$f_{ij}\{(c_i^\alpha, t_i^\alpha)\} = \begin{cases} (\max\{a_j, t_i^\alpha + t_{ij}\}, c_i^\alpha + c_{ij}), & (t_i^\alpha + t_{ij}) \leq b_j, \\ 0, & \text{caso contrário.} \end{cases}$$

A definição de f_{ij} permite um tempo de espera no nó j , quando o tempo de chegada for menor que o limitante inferior da janela de tempo. O novo conjunto $F_{ij}(Q_i)$ deve ser reduzido, pois etiquetas não mínimas podem ser geradas.

Denota-se por $F_{ij}(Q_i) \cup Q_j$ a redução de $F_{ij}(Q_i) \cup Q_j$, conservando apenas as etiquetas mínimas. As etiquetas associadas a um nó estão localizadas em uma lista com os tempos em ordem crescente.

A seguir é descrito o Algoritmo Janela de Tempo não Fuzzy (AJTnF), [31] adaptado para resolver problemas de caminho com custo mínimo com restrição tempo em redes de comunicação de dados:

Passo 1: *Iniciação*

$$P_i = \begin{cases} \{(0, 0)\}, & i = s, \\ \emptyset, & \forall_i \in N, i \neq s, \end{cases}$$

$$Q_i = \emptyset, \quad \forall_i \in N;$$

- P_i denota o conjunto irreduzível de etiquetas permanentes para o nó i ;
- Q_i denota o conjunto de etiquetas candidatas para o nó i .

Passo 2: *Procurar a etiqueta corrente*

- **Procurar** $F(Q)$, uma etiqueta $\{(t_i^\alpha, c_i^\alpha)\}$, de custo mínimo, no conjunto $Q = \cup_{i \in N} (Q_i - P_i)$, de forma lexicografica.
Se $(Q = \emptyset)$, **Pare**.
Calcular o limite superior de $B(Q)$.

Passo 3: *Procurar a próxima etiqueta para ser tratada*

- **Procurar** um elemento de $B(Q)$.
Se $B(Q)$ é um conjunto vazio, **Retorne ao Passo 1**.

Passo 4: *Tratamento da etiqueta (t_i^α, c_i^α)*

- **Para** todos os sucessores de j **até** o nó i **Faça**
Início:
Se $t_i^\alpha + t_{ij} \leq b_j$ (janela de tempo satisfeita)
Então $(t_j, c_j) = (\max(a_j, t_i^\alpha + t_{ij}), c_i^\alpha + c_{ij})$
 $Q_j = (Q_j \cup \{(t_j, c_j)\})$
Fim.
 $P_i = P_i \cup \{(t_i^\alpha, c_i^\alpha)\}$
Retorne ao Passo 2.

3.3.3 Problema de roteamento baseado em lógica fuzzy

Com a finalidade de formular o problema de roteamento considere $G = (N, M)$ um grafo, no qual N denota o conjunto de nós com janelas de tempo *fuzzy* associadas e M , o conjunto de arcos. Cada arco $(i, j) \in M$ tem um tempo *fuzzy* $\tilde{t}_{ij}$ de transferência e um custo *fuzzy* $\tilde{c}_{ij}$ associado. O tempo *fuzzy* $\tilde{t}_{ij}$ é a soma dos tempos *fuzzy* de enfileiramento (recebimento, processamento e transmissão) e propagação do nó i para o nó j . Um arco (i, j) é considerado no caminho somente quando for possível satisfazer a restrição $(\tilde{t}_i + \tilde{t}_{ij}) \leq \tilde{b}_j$, sendo $\tilde{t}_i$ o tempo *fuzzy* para acessar o nó i .

O problema de roteamento *fuzzy* é formalizado matematicamente como:

$$x_{ij} = \begin{cases} 1, & \text{se } (i, j) \in M \text{ é utilizado na conexão,} \\ 0, & \text{caso contrário.} \end{cases}$$

$$\begin{aligned}
& \text{Minimizar} && \sum_{(i,j) \in M} \tilde{c}_{ij} x_{ij} \\
& \text{sujeito a} && \\
& \forall j \in N, && \sum_{i \in N} x_{ij} - \sum_{i \in N} x_{ji} = \begin{cases} -1, & \text{para } j = s, \\ 1, & \text{para } j = d, \\ 0, & \text{caso contrário.} \end{cases} \quad (3.5) \\
& && \tilde{t}_s = \tilde{a}_s \\
& && \tilde{t}_j = x_{ij}(\tilde{t}_i + \tilde{t}_{ij}) + (1 - x_{ij})\tilde{a}_j; \quad i \in N, j \in N - \{s\}, \quad (3.6) \\
& && \tilde{a}_i \leq \tilde{t}_i \leq \tilde{b}_i; \quad i \in N, \quad (3.7) \\
& && x_{ij} \in \{0, 1\} \quad (i, j) \in M.
\end{aligned}$$

A restrição (3.5) garante que a conexão que sai do nó s chega ao nó d . A restrição (3.6) estabelece a compatibilidade requerida entre as rotas e os tempos *fuzzy*, e a restrição (3.7) estabelece o intervalo de tempo *fuzzy* no qual o nó pode ser acessado.

Portanto, trata-se de encontrar um caminho de custo *fuzzy* mínimo entre um nó origem s e um nó destino d na rede residual tal que satisfaça a banda de transmissão solicitada e o tempo *fuzzy* de transferência dos pacotes entre os nós que estejam contidos nas janelas temporais *fuzzy* de restrição $[\tilde{a}_i, \tilde{b}_i]$, especificadas para cada roteador i do caminho escolhido.

3.3.4 Algoritmo proposto

Baseado em lógica *fuzzy*, o algoritmo de caminho mínimo para uma rede com as restrições de atraso nos nós é descrito, de modo similar ao algoritmo AJTnF. O algoritmo proposto [16] armazena em cada nó pares de etiqueta $(\tilde{c}_i, \tilde{t}_i)$, ou seja, uma rota de custo mínimo *fuzzy*, chegando ao nó i , no tempo *fuzzy* $\tilde{t}_i$.

O conjunto irredutível associado ao nó i contém $\alpha(i)$ pares de etiquetas: $(\tilde{c}_i^\alpha, \tilde{t}_i^\alpha)$, onde $\alpha = 1, 2, \dots, \alpha(i)$, com

$$\tilde{a}_i \leq \tilde{t}_i^1 < \tilde{t}_i^2 < \dots < \tilde{t}_i^{\alpha(i)} \leq \tilde{b}_i \quad \text{e} \quad \tilde{c}_i^1 > \tilde{c}_i^2 > \dots > \tilde{c}_i^{\alpha(i)}.$$

Seja $Q_i = (\tilde{c}_i^\alpha, \tilde{t}_i^\alpha)$, $\alpha = 1, 2, \dots, \alpha(i)$, um conjunto irredutível de etiquetas associados ao nó i . Assim, unindo o arco (i, j) aos caminhos que saem da origem s e chegam em i , obtemos o conjunto Q_j associado ao nó j por meio da seguinte aplicação:

$$F_{ij}(Q_i) = \cup_{\alpha} f_{ij}\{(\tilde{c}_i^\alpha, \tilde{t}_i^\alpha)\}$$

onde

$$f_{ij}\{(\tilde{c}_i^\alpha, \tilde{t}_i^\alpha)\} = \begin{cases} (\max\{\tilde{a}_j, \tilde{t}_i^\alpha + \tilde{t}_{ij}\}, \tilde{c}_i^\alpha + \tilde{c}_{ij}), & (\tilde{t}_i^\alpha + \tilde{t}_{ij}) \leq \tilde{b}_j, \\ 0, & \text{caso contrário.} \end{cases}$$

A definição de f_{ij} permite um tempo de espera no nó j , quando o tempo de chegada é menor que o limitante inferior da janela de tempo. O novo conjunto $F_{ij}(Q_i)$ deve ser reduzido, pois etiquetas não mínimas podem ser geradas.

Denota-se por $\overline{F_{ij}(Q_i)} \cup Q_j$ a redução de $F_{ij}(Q_i) \cup Q_j$, conservando apenas as etiquetas mínimas. As etiquetas associadas a um nó são localizadas em lista com os tempos em ordem crescente.

A seguir é apresentado o Algoritmo Janela de Tempo Fuzzy (AJTF), proposto para resolver problemas de caminho com custo mínimo *fuzzy* com restrição de tempo *fuzzy* em redes de comunicação de dados:

Passo 1: *Iniciação*

$$P_i = \begin{cases} \{(0, 0)\}, & i = s, \\ \emptyset, & \forall i \in N, i \neq s, \end{cases}$$

$$Q_i = \emptyset \quad \forall i \in N$$

- P_i denota o conjunto irreduzível de etiquetas permanentes para o nó i ;
- Q_i denota o conjunto de etiquetas candidatas para o nó i .

Passo 2: *Procurar a etiqueta corrente*

- **Procurar** $F(Q)$, uma etiqueta $\{(\tilde{t}_i^\alpha, \tilde{c}_i^\alpha)\}$ de custo mínimo, no conjunto $Q = \cup_{i \in N} (Q_i - P_i)$, de forma lexicografica. **Se** $(Q = \emptyset)$, **Pare**.
Calcular o limitante superior de $B(Q)$.

Passo 3: *Procurar a próxima etiqueta para ser tratada*

- **Procurar** um elemento de $B(Q)$.
Se $B(Q)$ é um conjunto vazio, **Retorne ao passo 1**.

Passo 4: *Tratamento da etiqueta $(\tilde{t}_i^\alpha, \tilde{c}_i^\alpha)$*

- **Para** todos sucessores de j **Até** o nó i **Faça**
Início:
Se $\tilde{t}_i^\alpha + \tilde{t}_{ij} \leq \tilde{\Omega}_j$ (janela de tempo satisfeita)
Então $(\tilde{t}_j, \tilde{c}_j) = (\max(\tilde{\alpha}_j, \tilde{t}_i^\alpha + t_{ij}), \tilde{c}_i^\alpha + \tilde{c}_{ij})$
 $Q_j = (Q_j \cup \{(\tilde{t}_j, \tilde{c}_j)\})$
Fim.
 $P_i = P_i \cup \{(\tilde{t}_i^\alpha, \tilde{c}_i^\alpha)\}$
Retorne ao Passo 3.

3.4 Comparação de desempenho

Nesta seção os resultados numéricos são apresentados de modo a comparar o desempenho dos dois modelos descritos anteriormente, o modelo sem usar a lógica *fuzzy* e a proposta baseada em lógica *fuzzy*.

3.4.1 Modelo de rede e tráfego

Para a rede de seis nós da Figura 3.1, os pares $(c_{i,j}, t_{i,j})$, rotulados nos enlaces, indicam o custo e o tempo - valores unitários, respectivamente. Estes pares estão representados na Tabela 3.1. Na Figura 3.2, as etiquetas $(\tilde{c}_{i,j}, \tilde{t}_{i,j})$ são números *fuzzy*, representados na Tabela 3.3. Estes valores serão utilizados nas simulações. As Tabelas 3.2, 3.4, 3.5 e 3.6 mostram os caminhos encontrados pelos dois algoritmos. A Tabela 3.7 mostra que o número de conexões atendidas pelo algoritmo *fuzzy* proposto é maior.

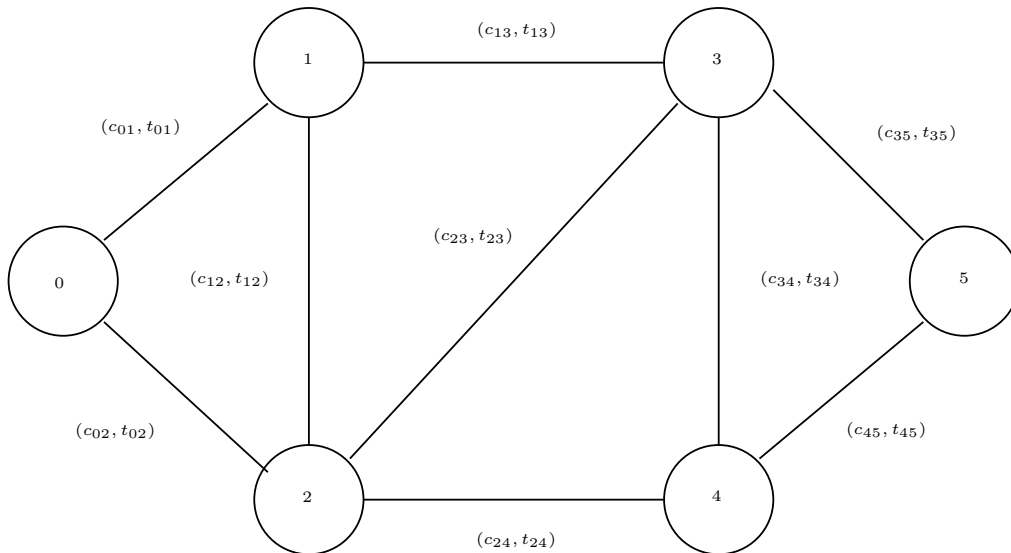


Figura 3.1: Modelo de rede de comunicação de dados simétrica com etiquetas não-*fuzzy*.

Enlace	não- <i>fuzzy</i>		
	Custo - c	Tempo de propagação - t	Largura de banda - unitário
(0,1)/(1,0)	20	63	7
(0,2)/(2,0)	20	20	3
(1,2)/(2,1)	20	41	6
(1,3)/(3,1)	24	15	2
(2,3)/(3,2)	22	9	6
(2,4)/(4,2)	24	57	2
(3,4)/(4,3)	19	41	4
(3,5)/(5,3)	30	75	4
(4,5)/(5,4)	25	75	4

Tabela 3.1: Informações sobre os valores não *fuzzy* da rede de comunicação de dados (Figura 3.1).

Caminho de s até d		Não Fuzzy		
Origem - s	Destino - d	Caminho	Tempo	Custo
1	3	-	-	-
1	2	-	-	-
0	2	-	-	-
2	0	-	-	-
2	1	-	-	-
3	1	-	-	-
3	2	[3 - 2]	9,0	22,0
2	3	[2 - 3]	9,0	22,0

Tabela 3.2: Caminhos de s até d com tempo fim-a-fim e menor custo, restrição de tempo não *fuzzy* = 10.

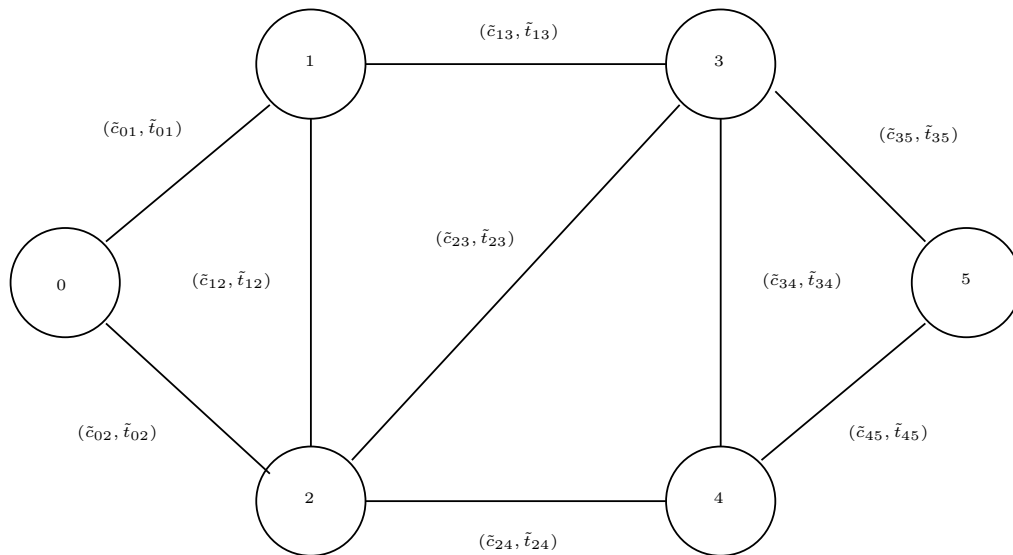


Figura 3.2: Modelo de rede de comunicação de dados simétrica com rótulos *fuzzy*.

Enlace	<i>fuzzy</i>						
	Custo - $\tilde{c}$			Tempo de propagação - $\tilde{t}$			Largura de banda - unitário
(0,1)/(1,0)	10	20	30	58	63	68	7
(0,2)/(2,0)	10	20	30	15	20	25	3
(1,2)/(2,1)	15	20	35	38	41	46	6
(1,3)/(3,1)	14	24	15	12	15	18	2
(2,3)/(3,2)	11	22	21	8	9	10	6
(2,4)/(4,2)	10	24	28	54	57	62	2
(3,4)/(4,3)	17	19	20	38	41	46	4
(3,5)/(5,3)	21	30	40	70	75	80	4
(4,5)/(5,4)	12	25	31	65	75	85	4

Tabela 3.3: Informações sobre os valores *fuzzy* da rede de comunicação de dados (Figura 3.2).

Caminho de s até d		Fuzzy		
Origem - s	Destino - d	Caminho	Tempo	Custo
1	3	[1 - 3]	(12,0; 15,0; 18,0)	(14,0; 15,0; 24,0)
1	2	[1 - 3 - 2]	(20,0; 24,0; 28,0)	(25,0; 36,0; 46,0)
0	2	[0 - 2]	(10,0; 20,0; 30,0)	(15,0; 20,0; 25,0)
2	0	[2 - 0]	(15,0; 20,0; 25,0)	(10,0; 20,0; 30,0)
2	1	[2 - 3 - 1]	(20,0; 24,0; 28,0)	(25,0; 36,0; 46,0)
3	1	[3 - 1]	(12,0; 15,0; 18,0)	(14,0; 15,0; 24,0)
3	2	[3 - 2]	(8,0; 9,0; 10,0)	(11,0; 21,0; 22,0)
2	3	[2 - 3]	(8,0; 9,0; 10,0)	(11,0; 21,0; 22,0)

Tabela 3.4: Caminhos de s até d com tempo fim-a-fim e menor custo, restrição de tempo Fuzzy = (5, 10, 20).

Caminho de s até d		Não Fuzzy		
Origem - s	Destino - d	Caminho	Tempo	Custo
5	4	-	-	-
0	4	-	-	-
4	0	-	-	-
5	3	-	-	-
4	5	-	-	-
3	5	-	-	-
1	0	[1 - 3 - 2 - 0]	44,0	66,0
0	2	[0 - 2]	20,0	20,0
4	1	[4 - 3 - 1]	56,0	43,0
3	2	[3 - 2]	9,0	22,0
1	3	[1 - 3]	15,0	24,0
2	4	[2 - 3 - 4]	50,0	41,0
1	2	[1 - 2]	41,0	20,0
0	3	[0 - 2 - 3]	29,0	42,0
3	4	[3 - 4]	41,0	19,0
2	3	[2 - 3]	9,0	22,0
2	0	[2 - 0]	20,0	20,0
2	1	[2 - 1]	41,0	20,0
3	0	[3 - 2 - 0]	29,0	42,0
3	1	[3 - 2 - 1]	50,0	42,0

Tabela 3.5: Caminhos de s até d com tempo fim-a-fim e menor custo, restrição de tempo não fuzzy = 60.

Caminho de s até d		Fuzzy		
Origem - s	Destino - d	Caminho	Tempo	Custo
5	4	[5 - 4]	(65,0; 75,0; 85,0)	(12,0; 25,0; 31,0)
0	4	[0 - 2 - 3 - 4]	(61,0; 70,0; 81,0)	(38,0; 61,0; 71,0)
4	0	[4 - 3 - 2 - 0]	(61,0; 70,0; 81,0)	(38,0; 61,0; 71,0)
5	3	[5 - 3]	(70,0; 75,0; 80,0)	(21,0; 30,0; 40,0)
4	5	[4 - 5]	(65,0; 75,0; 85,0)	(12,0; 25,0; 31,0)
3	5	[3 - 5]	(70,0; 75,0; 80,0)	(21,0; 30,0; 40,0)
1	0	[1 - 2 - 0]	(53,0; 61,0; 71,0)	(25,0; 40,0; 65,0)
0	2	[0 - 2]	(15,0; 20,0; 25,0)	(10,0; 20,0; 30,0)
4	1	[4 - 3 - 1]	(50,0; 56,0; 64,0)	(31,0; 35,0; 43,0)
3	2	[3 - 2]	(8,0; 9,0; 10,0)	(11,0; 21,0; 22,0)
1	3	[1 - 3]	(12,0; 15,0; 18,0)	(14,0; 15,0; 24,0)
2	4	[2 - 3 - 4]	(46,0; 50,0; 56,0)	(28,0; 41,0; 41,0)
1	2	[1 - 3 - 2]	(20,0; 24,0; 28,0)	(25,0; 36,0; 46,0)
0	3	[0 - 2 - 3]	(23,0; 29,0; 35,0)	(21,0; 42,0; 51,0)
3	4	[3 - 4]	(38,0; 41,0; 46,0)	(17,0; 19,0; 20,0)
2	3	[2 - 3]	(8,0; 9,0; 10,0)	(11,0; 21,0; 22,0)
2	0	[2 - 0]	(15,0; 20,0; 25,0)	(10,0; 20,0; 30,0)
2	1	[2 - 3 - 1]	(20,0; 24,0; 28,0)	(25,0; 36,0; 46,0)
3	0	[3 - 2 - 0]	(23,0; 29,0; 35,0)	(21,0; 42,0; 51,0)
3	1	[3 - 1]	(12,0; 15,0; 18,0)	(14,0; 15,0; 24,0)

Tabela 3.6: Caminhos de s até d com tempo fim-a-fim e menor custo, restrição de tempo = (50, 60, 70).

Não Fuzzy - AJTnF		Fuzzy - AJTF			
Tempo - ω	Bloqueio	Tempo - $\tilde{\omega}$		Bloqueio	
10	0,93	5	10	20	0,63
20	0,77	10	20	30	0,58
30	0,77	20	30	40	0,51
40	0,77	30	40	50	0,43
50	0,63	40	50	60	0,43
60	0,58	50	60	70	0,11
70	0,53	60	70	80	0,11
80	0,35	70	80	90	0,03
90	0,33	80	90	100	0,00
100	0,32	90	100	110	0,00
110	0,28	100	110	120	0,00

Tabela 3.7: Resultados da comparação de performance entre os algoritmos AJTnF e AJTF.

3.5 Proposta de controlador nebuloso de admissão de conexão

Um dos grandes desafios ao gerenciar as redes IP é garantir qualidade de serviço oferecido, utilizando a infra-estrutura de maneira otimizada. Uma das soluções encontradas é o uso da Engenharia de Tráfego com MPLS. Porém, as caracterizações das demandas e do estado da rede são tarefas difíceis, considerando que as demandas do tráfego de dados e o estado da rede mudam dinamicamente e de forma aleatória. Neste trabalho, é proposto um Controlador de Admissão de Conexão usando lógica *fuzzy* - BPFBR, baseada em regras lingüísticas, para tratar as informações com grau de incertezas em redes IP com MPLS, visando oferecer qualidade de serviço aos usuários.

O mecanismo de controle de admissão de conexão e determinação da rota, proposto neste trabalho, visa regular a quantidade de conexões, minimizando o congestionamento em uma rede IP com MPLS. Deste modo, o mecanismo busca garantir qualidade de serviço oferecida e suavizar as perdas de pacotes, otimizando a utilização da infra-estrutura da rede.

De acordo com os resultados da simulação apresentados na Tabela 3.7, o emprego da lógica *fuzzy* permite maior flexibilidade no processo de admissão de conexão e a possibilidade de incluir mais informações da rede e do tráfego na tomada de decisão, sem aumentar consideravelmente a complexidade do controlador.

3.5.1 Arquitetura do controlador nebuloso

O estado da rede e a ação de controle são determinados considerando o grau de incerteza. Neste contexto, o controlador nebuloso proposto seleciona uma rota não congestionada para a provável aplicação que vai ingressar na rede.

O controlador é composto de três módulos: o primeiro módulo (M1) é responsável pela seleção de um caminho com a menor taxa de utilização por meio do algoritmo Dijkstra [29], [32]. O segundo módulo (M2) é responsável pela decisão de aceitar ou não a aplicação com a taxa solicitada ou fornecer uma nova taxa dentro da realidade da rede, na situação em que foi aceito o pedido de conexão. O terceiro módulo (M3) tem como objetivo fazer o estabelecimento do Label Switched Path (LSP) entre uma origem e um destino. A arquitetura desse controlador nebuloso é mostrada na Figura 3.3.

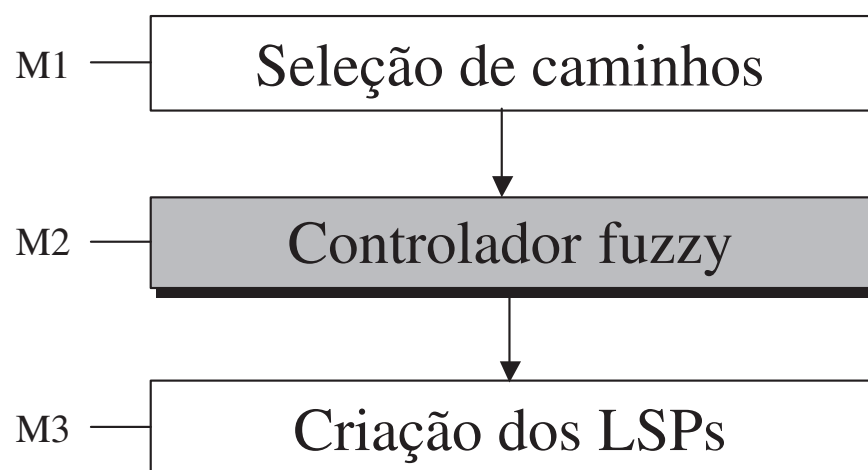


Figura 3.3: Arquitetura do controlador nebuloso de admissão de conexão.

3.5.2 Funções de pertinência do controlador

As funções de pertinência *fuzzy* utilizadas são contínuas do tipo triangular, $\text{trimf}(x;a,m,b)$, e trapezoidal, $\text{trapmf}(x;a,m,n,b)$, definidas conforme as equações 3.8 e 3.9, respectivamente,

$$\text{trimf}(x;a,m,b) = \begin{cases} 0, & \text{se } x \leq a, \\ \frac{x-a}{m-a}, & \text{se } x \in (a,m], \\ \frac{b-x}{b-m}, & \text{se } x \in [m,b), \\ 0, & \text{se } x \geq b, \end{cases} \quad (3.8)$$

$$\text{trapmf}(x;a,m,n,b) = \begin{cases} 0, & \text{se } x \leq a, \\ \frac{x-a}{m-a}, & \text{se } x \in (a,m], \\ 1, & \text{se } x \in [m,n], \\ \frac{b-x}{b-n}, & \text{se } x \in [n,b), \\ 0, & \text{se } x \geq b, \end{cases} \quad (3.9)$$

onde m e n denotam valores modais e a e b denotam os limites inferior e superior, respectivamente. Estas funções foram escolhidas por serem implementadas com código mais simples e eficiente.

3.5.3 Variáveis lingüísticas do controlador

Na implementação do controlador foram adotadas a ocupação do enlace e a requisição da aplicação como variáveis de entrada, e para a variável de saída, uma resposta de solicitação (SIM, NÃO) ou a taxa de qualidade de serviço que o sistema pode garantir. A variável lingüística "Ocupação do enlace" está apresentada na Tabela 3.8, a variável "Requisição da aplicação" na Tabela 3.9, e a "Resposta à solicitação" na Tabela 3.10. A representação gráfica das funções de pertinência para cada uma das variáveis estão ilustradas nas Figuras 3.4, 3.5 e 3.6, respectivamente.

Variável lingüística	Valor (Mbit/s)	Tipo da função	Faixa da função [a,m,n,b]
OC1	1,0	trapmf	[0,5,1,0;1.5;2,0]
OC2	2,0	trapmf	[1.5,2,0;2.5;3,0]
OC3	3,0	trapmf	[2.5,3,0;3.5;4,0]
OC4	4,0	trapmf	[3.5,4,0;4.5;5,0]
OC5	5,0	trapmf	[4.5,5,0;5.5;6,0]
OC6	6,0	trapmf	[5.5,6,0;6.5;7,0]
OC7	7,0	trapmf	[6.5,7,0;7.5;8,0]
OC8	8,0	trapmf	[7.5,8,0;8.5;9,0]
OC9	9,0	trapmf	[8.5,9,0;9.5;10,0]

Tabela 3.8: Funções de pertinência para variáveis de entrada - Ocupação do enlace.

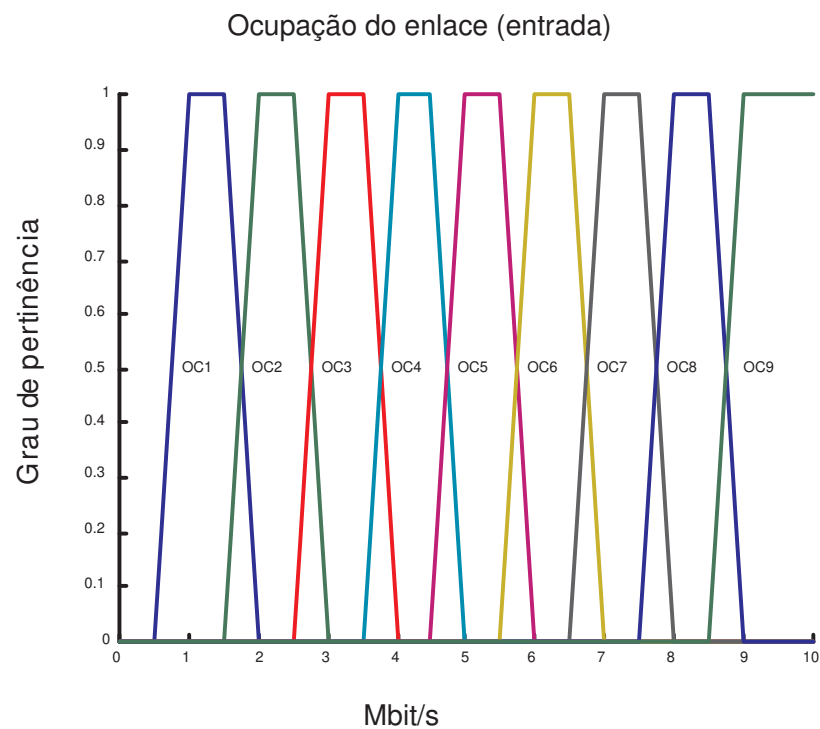


Figura 3.4: Funções de pertinência para a variável de entrada - Ocupação do enlace.

Variável lingüística	Valor (Mbit/s)	Tipo da função	Faixa da função [a,m,n,b]
T03	0,3	trapmf	[0,0;0,3,0.7;1,0]
T1	1,0	trapmf	[0,7,1,0;1.5;2,0]
T2	2,0	trapmf	[1.5,2,0;2.5;3,0]
T3	3,0	trapmf	[2.5,3,0;3.5;4,0]
T4	4,0	trapmf	[3.5,4,0;4.5;5,0]
T5	5,0	trapmf	[4.5,5,0;5.5;6,0]
T6	6,0	trapmf	[5.5,6,0;6.5;7,0]
T7	7,0	trapmf	[6.5,7,0;7.5;8,0]
T8	8,0	trapmf	[7.5,8,0;8.5;9,0]
T9	9,0	trapmf	[8.5,9,0;9.5;10,0]

Tabela 3.9: Funções de pertinência para variáveis de entrada - requisição da aplicação.

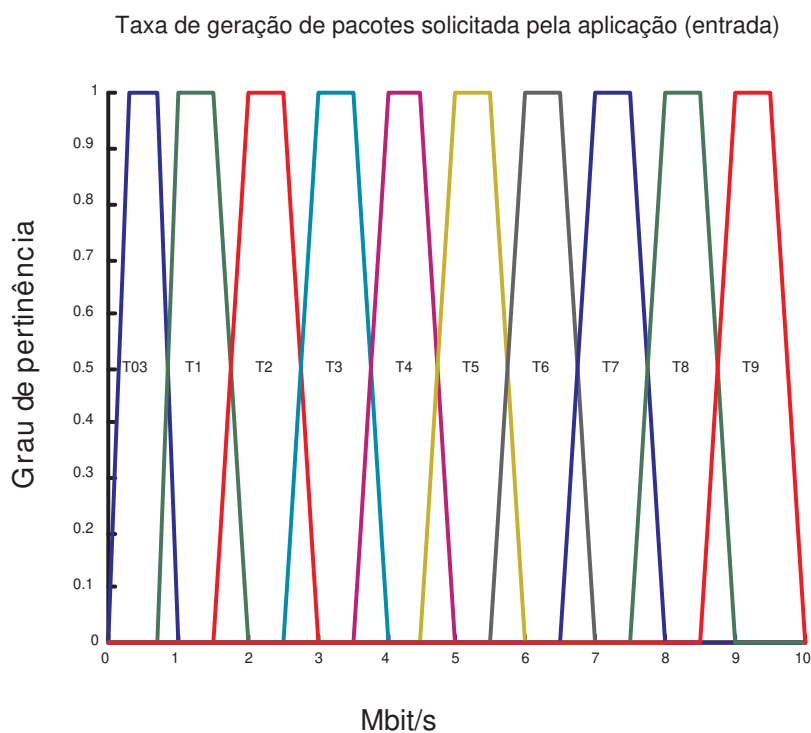


Figura 3.5: Funções de pertinência para a variável de entrada - Requisição da aplicação.

Variável lingüística	Valor (Mbit/s)	Tipo da função	Faixa da função $[a,m,n,b]$
SIM	conexão aceita	trapmf	$[-2,0;-2,0;-1,0;0,3]$
qK	0,5	trimf	$[0,0;0,5;0,8]$
uM	1,0	trimf	$[0,5;1,0;1,5]$
dM	2,0	trimf	$[1,0;2,0;2,5]$
tM	3,0	trimf	$[2,0;3,0;3,5]$
qM	4,0	trimf	$[3,0;4,0;4,5]$
cM	5,0	trimf	$[4,0;5,0;5,5]$
sM	6,0	trimf	$[5,0;6,0;6,5]$
NÃO	conexão recusada	trapmf	$[6,0;9,0;11,0;11,0]$

Tabela 3.10: Funções de pertinência para variáveis de saída - resposta à solicitação.

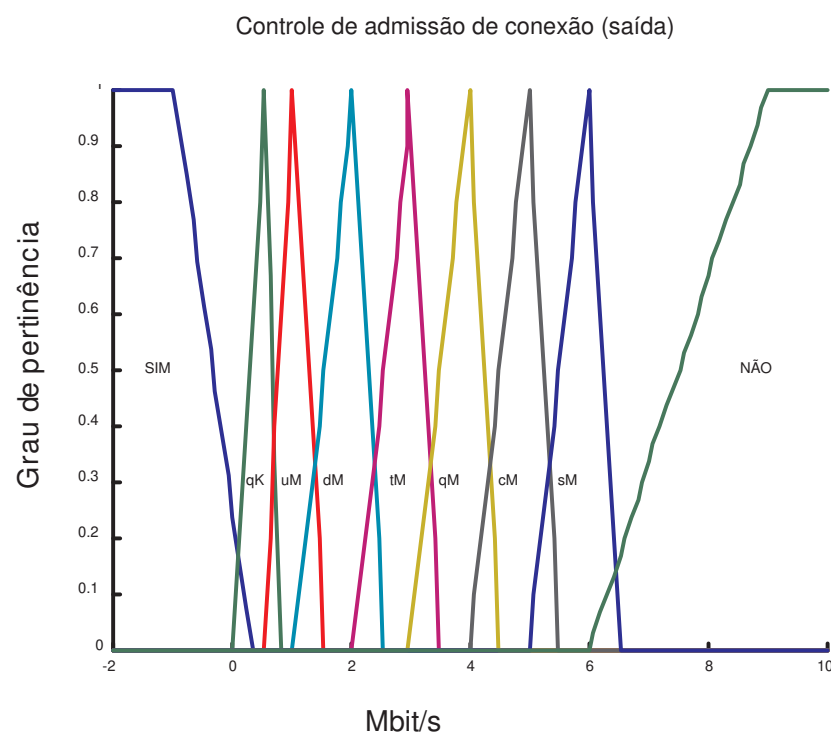


Figura 3.6: Funções de pertinência para a variável de saída - Resposta à solicitação.

3.5.4 O conjunto de regras *fuzzy*

As regras de inferência que relacionam as diversas variáveis linqüísticas e seus respectivos valores estão resumidos na Tabela 3.11. Estas regras deram origem ao conjunto de regras *fuzzy*, no formato **Se-Então**, semelhante às apresentadas na Subseção 2.3.2 do Capítulo 2. O processo de inferência utilizado na simulação é conhecido como Mandani [33] e [28].

Ocupação do <i>link</i>	Requisição da aplicação									
	T03	T1	T2	T3	T4	T5	T6	T7	T8	T9
OC1	SIM	SIM	SIM	SIM	SIM	SIM	SIM	SIM	SIM	NÃO
OC2	SIM	SIM	SIM	SIM	SIM	SIM	SIM	SIM	sM	NÃO
OC3	SIM	SIM	SIM	SIM	SIM	SIM	SIM	sM	sM	NÃO
OC4	SIM	SIM	SIM	SIM	SIM	SIM	cM	cM	cM	NÃO
OC5	SIM	SIM	SIM	SIM	SIM	qM	qM	qM	qM	NÃO
OC6	SIM	SIM	SIM	SIM	tM	tM	tM	tM	tM	NÃO
OC7	SIM	SIM	SIM	dM	dM	dM	dM	dM	dM	NÃO
OC8	SIM	SIM	qK	qK	qK	qK	qK	qK	qK	NÃO
OC9	NÃO	NÃO	NÃO	NÃO	NÃO	NÃO	NÃO	NÃO	NÃO	NÃO

Tabela 3.11: Apresentação das regras de inferência.

3.6 Ambiente de teste

3.6.1 Sobre a ferramenta de testes

Para o teste experimental do controlador, foi utilizado o simulador de rede NS-2.1b7a - *Network Simulator* versão 2.1b7a (NS). O NS-2 é um simulador desenvolvido pela Berkely UC, escrito em C++ e OTcl e possui o módulo MNS (*MPLS Network Simulator*) [34]. O NS-2 permite testar: a troca de etiquetas, os módulos: LDP e CR-LDP, como um LSP é criado e destruído. Sendo a funcionalidade MPLS do simulador modular, gratuito (com código aberto e documentações) foi considerado adequado para realizar as simulações desejadas.

3.6.2 Cenários simulados

A topologia usada nas simulações pode ser vista na Figura 3.7. Esta topologia, conhecida como *Shuffle Net* (SN), é comumente utilizada nas redes de pacotes. Para a topologia de rede, os roteadores são LERs (*Label Edge Routers*) e LSRs (*Label Switch Router*) ao mesmo tempo. A rede foi simulada com os fluxos partindo de todos nós 1 - 8 para todos os nós. Todos os pacotes utilizam o UDP (*User Datagram Protocol*), como protocolo da camada de transporte, têm um comprimento de 576 bytes e recebem uma única etiqueta de 4 bytes do MPLS. Nos cenários simulados, todos os enlaces obedecem ao mesmo dimensionamento, (10 Mbit/s com

5 ms de tempo de propagação e os LSR/LER (*Label Switching Router/Label Edge Router*) têm filas do tipo FIFO (*First-In-First-Out*) para 400 pacotes ou 230.400 bytes). O número de circuitos virtuais foi tomado igual ao número de fontes existentes na rede (56). Foram utilizados dois tipos de fontes de transmissão aleatoriamente: VBR (*Variable Bit Rate*) com tráfego, obedecendo à distribuição exponencial; e CBR (*Constant Bit Rate*) cuja taxa média de transmissão, também aleatória, tem distribuição uniforme no intervalo 500kbit/s - 9Mbit/s.

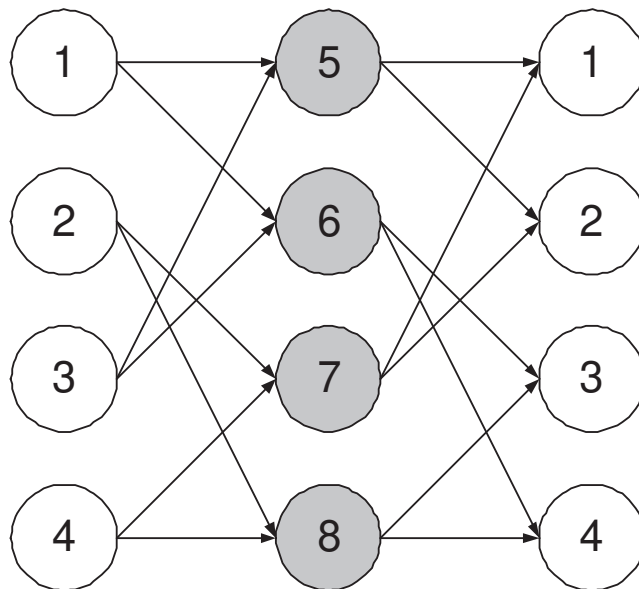


Figura 3.7: Topologia usada na simulação.

Na topologia de rede, cada fonte inicia sua transmissão em um instante aleatório, seguindo uma distribuição poissoniana com taxas médias de chegada - λ variando de 1 a 5 conexões por segundo. O tempo de permanência de cada fonte foi gerado a partir de uma distribuição exponencial negativa com duração média de 0,25 s.

3.6.3 Resultados

Para comparar o desempenho do controlador *fuzzy* proposto foram implementadas outras políticas de distribuição do tráfego e controle de admissão de conexões na rede IP/MPLS, a saber: distribuição do tráfego na rede utilizando o menor caminho, com função custo baseada na banda residual sem bloqueio (SB) e com bloqueio (BBR).

As Figuras 3.8, 3.9 e 3.10 mostram o número de pacotes perdidos em função do tempo, para algumas taxas médias de chegada. Como pode ser visto, o controlador *fuzzy* apresenta uma perda menor de pacotes e uma melhor uniformidade ao longo da simulação, quando comparado à política de bloqueio total utilizando banda residual (BBR). Isso se deve ao fato de o controlador *fuzzy* oferecer uma menor rigidez nos critérios de decisão de admissão de conexão.

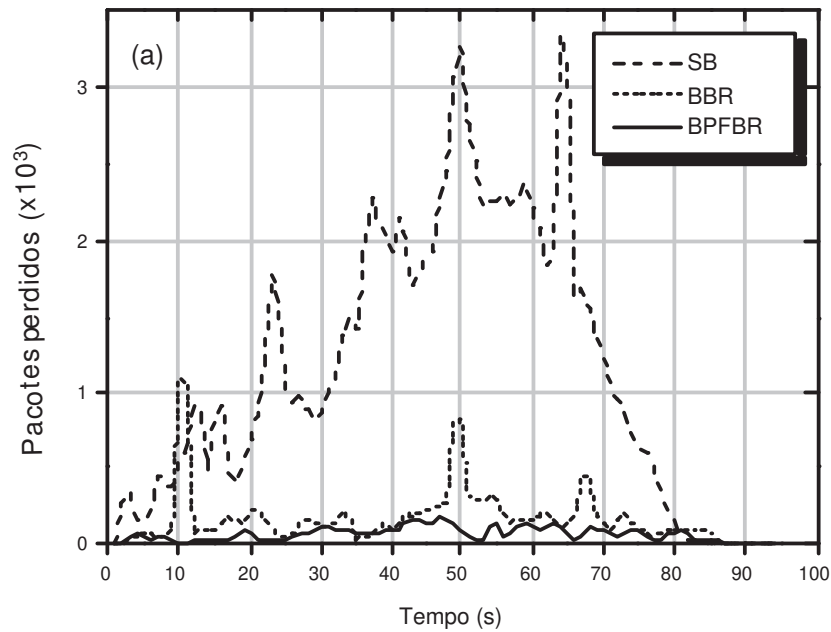


Figura 3.8: Número de pacotes perdidos em função do tempo para a taxa média de chegada (conexões / s): (a) $l = 1$.

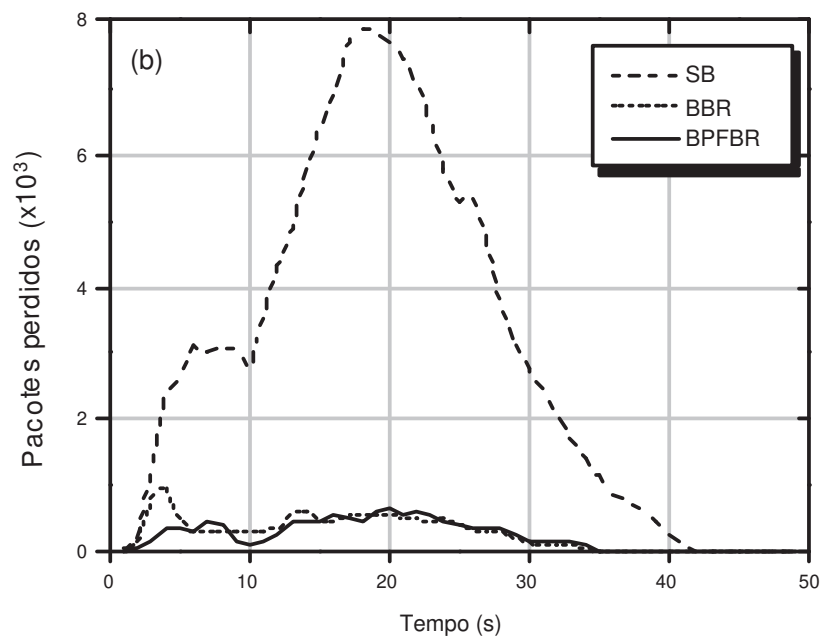


Figura 3.9: Número de pacotes perdidos em função do tempo para a taxa média de chegada (conexões / s): (b) $l = 3$.

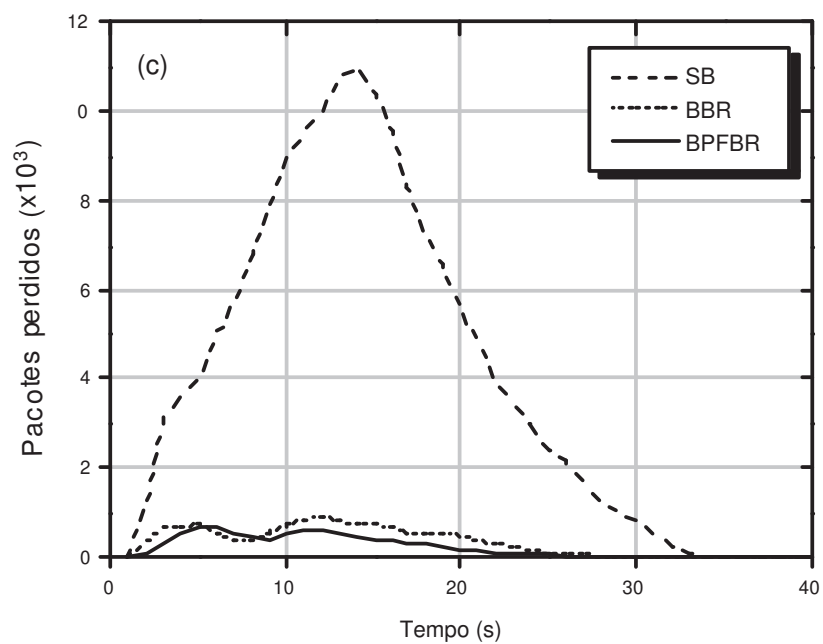


Figura 3.10: Número de pacotes perdidos em função do tempo para a taxa médias de chegada (conexões / s): (c) $l = 5$.

Na Figura 3.11 é ilustrada a relação entre o número total de pacotes perdidos e o número total de pacotes gerados em função da taxa média de chegada das conexões. Para as taxas simuladas, o controlador *fuzzy* apresentou um percentual de pacotes perdidos, na média, abaixo de 5% e um ganho da ordem de 35% em relação à política BBR.

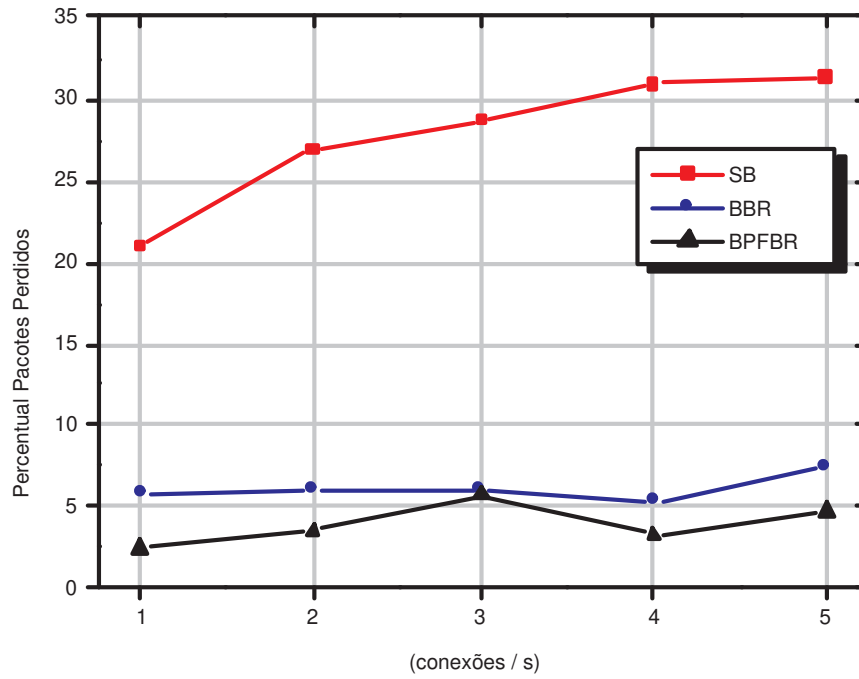


Figura 3.11: Relação entre número total de pacotes perdidos e número total de pacotes gerados em função da taxa média de chegada das conexões.

Um resultado que também deve ser analisado é o percentual de conexões bloqueadas. Um dos objetivos de QoS, utilizando controle de admissão, é o de garantir baixa perda de pacotes para as conexões ingressantes e para as já existentes, mas ao mesmo tempo tentar oferecer acesso à rede para o maior número possível de aplicações. Isso pode ser visto na Figura 3.12, que mostra o percentual de conexões bloqueadas em função da taxa média de chegada das conexões. Nesta figura se pode notar que o controlador *fuzzy* BPFBR (Bloqueio Parcial Fuzzy com Banda Residual) sempre apresenta menor bloqueio, comparado à política BBR. Isto ocorre porque quando uma aplicação requisita uma conexão com taxa maior que a banda residual, o controlador *fuzzy* oferece uma nova taxa, que não compromete os contratos de QoS das aplicações existentes na rede.

De acordo com os resultados da simulação apresentados nas Figuras 3.12, 3.11, o emprego da lógica *fuzzy* permite maior flexibilidade no processo de admissão de conexão e a possibilidade de incluir mais informações da rede e do tráfego na tomada de decisão, sem aumentar consideravelmente a complexidade do controlador.

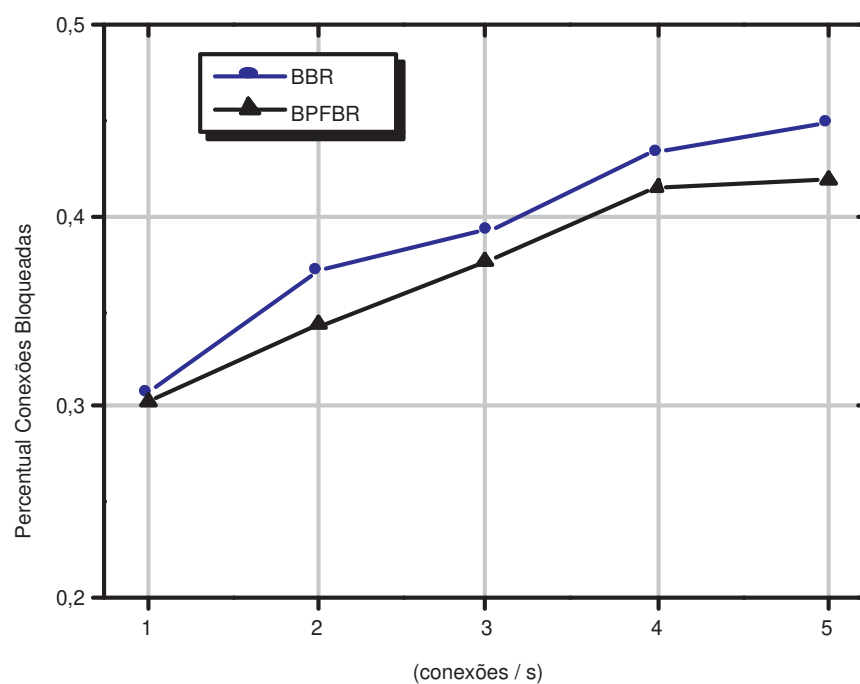


Figura 3.12: Percentual de conexões bloqueadas em função da taxa média de chegada das conexões.

Capítulo 4

Conclusões e trabalhos futuros

4.1 Conclusões

Neste trabalho foram estudadas formas de prover QoS em redes IP por meio da lógica *fuzzy* devido a incertezas na rede. Verificou-se que as técnicas propostas apresentaram boas propriedades para métodos interativos (*real time*) e baixa complexidade computacional e, conseqüentemente, simples exigências de recursos computacionais de processamento (AMD Athlon 1.6GHz e memória RAM de 640Mb, este foi o recurso utilizado nas simulações) em termos de CPU e memória.

No Capítulo 2 foi apresentada a proposta de um módulo de inteligência, utilizando lógica *fuzzy*, para gerenciamento da QoS baseado em políticas. Essa proposta é diferenciada das soluções proprietárias atuais por possuir um módulo inteligente para auxiliar o administrador na tomada de decisões e por ser um sistema de gerenciamento de QoS com o ciclo fechado - criação de políticas, configuração, monitoração, avaliação dos resultados da monitoração, tomada de decisões, reconfiguração de QoS quando necessário.

No Capítulo 3, para o problema restrição de tempo com menor custo (*delay-constrained least-cost routing problem*) é descrita uma proposta de algoritmo para as redes IP, tendo como principal alteração métricas aditivas tratadas como variáveis *fuzzy*. Essa adaptação conseguiu a redução no número de comparações de soluções e o descarte de caminhos que não participam do conjunto solução. Simulações mostraram que o algoritmo proposto tem um desempenho melhor que o correspondente algoritmo *crisp*, com respeito à taxa de perda de chamada. Verificou-se também que a proposta de controle de admissão de conexão baseado em regras que utiliza conceitos de lógica *fuzzy*, aplicado a redes IP com MPLS, apresentou uma maior flexibilização da admissão de conexão e possibilidade de incluir mais informações da rede e do tráfego na tomada da decisão, sem aumentar consideravelmente a complexidade do controlador. Além disso, por tratar conceitos lingüísticos (mais, menos, maior e outros) de forma parecida com a do ser humano, a programação do sistema *fuzzy* é bastante natural. Finalmete, deve ser notado que o algoritmo proposto pode ser facilmente estendido para considerar outros requisitos de QoS *fuzzy*, além de custo e atraso.

4.2 Sugestões para trabalhos futuros

Apontam-se os seguintes itens como prováveis trabalhos futuros:

1. A partir da experiência adquirida com o desenvolvimento desta pesquisa, iniciar uma nova fase para provisionamento de QoS por meio da lógica *fuzzy* que leve em conta os estudos do Kuipers em [35] referente ao problema MCP (*Multiple Constrained Path*);
2. Estudar as funções *fuzzy* que melhor representam as métricas aditiva, multiplicativa e côncava [36];
3. Estudar algoritmos genéticos (AG) e redes neurais (RN) com o objetivo de criar sistema de suporte a decisão para definição de políticas de gerência de rede.

Referências Bibliográficas

- [1] J. Kurose e K. W. Ross. *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison Wesley, 3 edition, 2004.
- [2] Y. Yang e L. Zhang e J. K. Muppala e S. T. Chanson. Bandwidth-delay constrained routing algorithms. *Computer Networks*, 42:503–520, July 2003.
- [3] X. Masip-Bruin e S. Sànchez-López e J. Solé-Pareta e J. Domingo-Pascual. Qos routing algorithms under inaccurate routing information for bandwidth constrained applications. In *ICC 2003 - IEEE International Conference on Communications*, volume 3, pages 1743 – 1748, Anchorage, Alaska, May 2003. IEEE.
- [4] F. A. Kuipers e P. V. Mieghem. The impact of correlated link weights on qos routing. In *INFOCOM 2003 - Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1425 – 1434, San Francisco, California - USA, March/April 2003. IEEE.
- [5] T. Korkmaz M. Krunz F. Kuipers, P. Van Mieghem. An overview of constraint-based path selection algorithms for qos routing. *IEEE Communications Magazine*, 40:50 – 55, December 2002.
- [6] S. Chen e K. Nahrstedt. An overview of quality of service routing for next-generation high-speed networks: problems and solutions. *IEEE Networks*, 12:64 – 79, November/December 1998.
- [7] B. Fortz e J. Rexford e M. Thorup. Traffic engineering with traditional ip routing protocols. *IEEE Communications Magazine*, 40:118–124, October 2002.
- [8] B. Fortz e M. Thorup. Optimizing ospf/is-is weights in a changing world. *IEEE Journal on Selected Areas in Communications*, 20:756–767, May 2002.
- [9] A. Dutta-Roy. The cost of quality in internet-style networks. *IEEE Spectrum*, 37(9):57–62, February 2000.
- [10] ISO-IEC JTC1-SC21. Information Technologoy - Quality of Service: Framework - Final CD. Technical report, July 1995.
- [11] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.

- [12] M. A. Siqueira e N. A. Nassif e R. A. Resende e A. E. Silva e M. Lima-Marques. Policy-based architecture for qos management in enterprise ip networks. In *IM2005 9th IFIP/IEEE International Symposium on Integrated Network Management*, volume y, pages xx – xx, Nice, France, May 2005. IFIP/IEEE.
- [13] R. A. Resende e M. A. Siqueira e N. A. Nassif e A. E. Silva e M. Lima-Marques. Quality of service control in ip networks using fuzzy logic for policy condition evaluation. In *FUZZ-IEEE 2005, 14th IEEE International Conference on Fuzzy Systems - Biggest Little Conference in the World*, volume 2, pages 448 – 453, Reno, Nevada, USA, May 2005. IEEE.
- [14] R. A. Resende e R. M. Oliveira e F. J. L. Padua e A. Yamakami e I. S. Bonatti. Roteamento com restrições utilizando janela de tempo em redes de comunicação. In *CRC2004 7ª Conferência sobre Redes de Computadores*, Leiria, Portugal, Outubro 2004.
- [15] R. A. Resende e R. M. Oliveira e F. J. L. Padua e A. Yamakami e I. S. Bonatti e A. C. Lavelha. An Algorithm for Quality-of-Service Unicast Routing in Data Communication Networks. In *ICT2006 - 13th International Conference on Telecommunications*, Funchal, Madeira island, Portugal, 9 - 12, May 2006. IEEE/IEE.
- [16] R. A. Resende e A. C. Lavelha e A. Yamakami e I. S. Bonatti. A Fuzzy Algorithm to Solve the Problem of QoS Unicast Routing in IP Networks. In *ITS2006 - 5th International Telecommunications Symposium*, Fortaleza - Ceará, Brazil, 3 - 6, September 2006. SBrT/IEEE - Accepted.
- [17] R. A. Resende e S. M. Rossi e A. Yamakami e L. H. Bonani. Controle de admissão de conexão em rede IP/MPLS usando lógica fuzzy baseada em regras lingüísticas. In *Workcomp'2002 - V Workshop de Computação ITA*, São José dos Campos - SP, Brasil, Outubro 2002. ITA - Instituto Tecnológico de Aeronáutica.
- [18] R. A. Resende e S. M. Rossi e A. Yamakami e L. H. Bonani e E. Moschim. Traffic engineering with mpls using fuzzy logic for application in ip networks. In *FUZZ-IEEE 2003 - 12th IEEE International Conference on Fuzzy Systems*, volume 2, pages 1146 – 1151, St. Louis, MO, USA, May 2003. IEEE.
- [19] K. Nichols e V. Jacobson e L. Zhang. A two-bit differentiated services architecture for the internet. IETF, Network Working Group, Request for Comments (RFC) - RFC2638, July 1999.
- [20] O. Wendell. *CISCO DQOS Exam Certification Guide - IP Telephony Self-Study*. CISCO Press, 1 edition, 2003.
- [21] W. Pedrycz e F. Gomide. *An introduction of the fuzzy sets: analysis and design*. A Bardford Book, 1998.
- [22] Feng-Tse Lin. A shortest-path network problem in a fuzzy environment. In *FUZZ-IEEE 2001 - 10th IEEE International Conference on Fuzzy Systems*, volume 3, pages 1096 – 1100, Melbourne, Australia, December 2001. IEEE.

- [23] S. Okada. Fuzzy shortest path problems incorporating interactivity among paths. *Fuzzy Sets and Systems*, 142(3):335–357, 2004.
- [24] ETSI TS 101 329-2 v1.1.1: Telecommunications and internet protocol harmonization over networks (TIPHON); end to end quality of service in TIPHON systems; part 2: Definition of quality of service (QoS) classes., 2000.
- [25] ETSI TR 101 329-6 v1.1: Telecommunications and internet protocol harmonisation over networks (TIPHON); end to end quality of service in TIPHON systems; part 6: Actual measurement of network and terminal characteristics and performance parameters in TIPHON networks and their influence on voice quality.
- [26] TR 101 329 v2.1.1: Telecommunications and internet protocol harmonisation over networks (TIPHON); general aspects of quality of service (QoS)., 1999.
- [27] M. G. Simões e I. Shaw. *Controle e Modelagem Fuzzy*. Edgard Blucher, 1 edition, 1999.
- [28] H. T. Nguyen. *A First Course in Fuzzy Logic*. IE-CRC PRESS, 2 edition, 1999.
- [29] T. H. Cormen e C. E. Leiserson e R. L. Rivest. *Introduction to Algorithms*. MIT Press, 2 edition, 2001.
- [30] T. H. Cormen e C. E. Leiserson e R. L. Rivest. *Introduction to Algorithms*. Cambridge, MA, 1990.
- [31] C. B. Cunha e J. Swaitb. New dominance criteria for the generalized permanent labelling algorithm for the shortest path problem with time windows on dense graphs. *International Transactions in Operational Research - Elsevier*, pages 139–157, March 2000.
- [32] R. K. Ahuja e T. L. Magnanti e J. B. Orlin. *Networks Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., New Jersey, 1993.
- [33] E. H. Mamdani e N. Baaklini. Prescriptive methods for deriving control policy in a fuzzy logical controller. *Electron. Lett.*, (11):625–626, 1975.
- [34] G. Ahn e C. Woojik. Design e implementation of mpls network simulator supporting ldp and cr-ldp. In *ICON 2000 - 8th IEEE International Conference on Networks*, pages 441–446, Singapore, September 2000. IEEE.
- [35] F. A. Kuipers. *Quality of Service Routing in the Internet - Theory, Complexity and Algorithms*. Ph.d. thesis, Delft University Press, The Netherlands, September 2004.
- [36] X. Xiao e L. M. Ni. Internet qos: the big picture. *IEEE Network Magazine*, 13(2):8–18, March/April 1999.
- [37] M. T. Takahashi. *Contribuições ao Estudo de Grafos Fuzzy: teoria e algoritmos*. Doutorado em engenharia elétrica, Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, Departamento de Telemática, Campinas - São Paulo, Brasil, Abril 2004.

- [38] S. Ghosh e Q. Razouqi e H. J. Schumacher e A. Celmins. A survey of recent advances in fuzzy logic in telecommunications networks and new challenges. *IEEE Transactions on Fuzzy Systems*, 6:443 – 447, August 1998.
- [39] A. S. Moussa e L. J. Kohout. A new network state information updating policy using fuzzy tolerance classes and fuzzified precision. In *NAFIPS 2002 - North American Fuzzy Information Processing Society*, pages 541 – 545, New Orleans, Louisiana, USA, June 2002. IEEE.
- [40] J. Jantzen. Design of fuzzy controllers. Technical report, Technical University of Denmark: Dept. of Automation, 1998.
- [41] M. M. R. Akbarzadeh-T e E. Tunstel e Kumbala, K. e Jamshidi. Soft computing paradigms for hybrid fuzzy controllers: Experiments and applications. In *WCCI98 - In IEEE World Congress on Computational Intelligence*, volume 2, pages 1200–1205, Anchorage, Alaska, USA, May 1998. IEEE.
- [42] J. C. Strassner. *Policy-Based Network Management – Solutions for the Next Generation*. 1 edition, 2003.
- [43] J. C. Strassner. *Directory Enabled Networks*. 1 edition, 1999.
- [44] RFC3060 - policy core information model, version 1 specification. Technical report, IETF, Network Working Group, 2001.
- [45] M. Wahl e T. Howes e J. Strassner e S. Kille. RFC2251 - lightweight directory access protocol, version 3. Technical report, IETF, Network Working Group, December 1997.
- [46] D. W. Chadwick. *Understanding X.500 - The Directory*. Chapman & Hall, 1 edition, 1996.
- [47] Zongmin MA. *Fuzzy Database Modeling With XML*, volume 29. Springer Verlag Ny, 1 edition, February 2005.

Apêndice A

Trabalhos publicados pelo autor

A.1 Publicações relacionadas ao capítulo 2

1. M. A. Siqueira, N. A. Nassif, R. A. **Resende**, A. E. Silva e M. Lima-Marques, "Policy-Based Architecture for QoS Management in Enterprise IP Networks". *IM2005 9th IFIP/IEEE International Symposium on Integrated Network Management*, Nice, France, May, 16-19, 2005.
2. R. A. **Resende**, M. A. Siqueira, N. A. Nassif, A. E. Silva e M. Lima-Marques, "Quality of Service control in IP Networks using Fuzzy Logic for Policy Condition Evaluation". *FUZZ-IEEE 2005 - Biggest Little Conference in the World*, Reno, Nevada, USA, May 22-25, 2005.

A.2 Publicações relacionadas ao capítulo 3

1. R. A. **Resende**, A. C. Lavelha, A. Yamakami e I. S. Bonatti, "A Fuzzy Algorithm to Solve the Problem of QoS Unicast Routing in IP Networks". *ITS2006 - 5th International Telecommunications Symposium*, Fortaleza - Ceará, Brazil, September 3 - 6, 2006 - SBrT/IEEE.
2. R. A. **Resende**, R. M. Oliveira, F. J. L. Pádua, A. Yamakami, I. S. Bonatti e A. C. Lavelha, "An Algorithm for Quality-of-Service Unicast Routing in Data Communication Networks". *ICT2006 - 13th International Conference on Telecommunications*, Funchal, Madeira island, Portugal, May 9 - 12, 2006 IEEE/IEE.
3. R. A. **Resende**, R. M. Oliveira, F. J. L. Pádua, A. Yamakami e I. S. Bonatti, "Roteamento com Restrições Utilizando Janela de Tempo em Redes de Comunicação". *CRC2004 7ª Conferência sobre Redes de Computadores*, Leiria - Portugal, 7 e 8 de Outubro de 2004.
4. R. A. **Resende**, S. M. Rossi, A. Yamakami, L. H. Bonani e E. Moschim. "Traffic engineering with MPLS using fuzzy logic for application in IP networks". *Fuzzy Systems, 2003. FUZZ '03. The 12th IEEE International Conference*, St. Louis, MO, USA, 25-28 May 2003, Page(s):1146 - 1151, vol.2.

5. R. A. **Resende**, S. M. Rossi, A. Yamakami e L. H. Bonani, "Controle de Admissão de Conexão em Rede IP/MPLS Usando Lógica Fuzzy Baseada em Regras Lingüísticas", Anais do Workcomp'2002 - V Workshop de Computação ITA, São José dos Campos - SP, Brasil, 15 a 16 de Outubro de 2002.

Apêndice B

Teoria *fuzzy*

Neste apêndice são apresentados elementos e conceitos da lógica *fuzzy* utilizados na implementação das contribuições desta tese. Para mais detalhes da teoria *fuzzy* remetemos o leitor às referências bibliográficas [21], [28], [37].

B.1 Base de regras *fuzzy*

O conhecimento humano pode ser representado na forma de regras *fuzzy* "IF-THEN". A base de regras *fuzzy* consiste de um conjunto de regras "IF-THEN", sendo considerada o "núcleo" de um sistema *fuzzy*, uma vez que todos os outros componentes são usados para implementar as regras de modo eficiente e razoável [27].

B.1.1 *Fuzzyficação*

Fuzzyficação é o processo de associar ou calcular um valor para representar um grau de pertinência da entrada em um ou mais grupos qualitativos, chamados de conjuntos difusos. O grau de pertinência é determinado por uma função de pertinência definida com base na experiência ou intuição. Funções de pertinência são o meio pelo qual um controlador é sintonizado para alcançar respostas desejadas a partir determinadas entradas [28].

B.1.2 Inferência

O SE (IF) de uma regra contém uma ou mais condições, chamadas de antecedentes, que constituem uma premissa; o ENTÃO (THEN) contém uma ou mais ações chamadas de consequentes [21].

O antecedente corresponde diretamente aos graus de pertinência calculados durante o processo de *fuzzyficação*. Cada antecedente tem um grau de pertinência indicado para ele como resultado da *fuzzyficação*. Durante a avaliação das regras (a inferência), a intensidade é calculada com base em valores dos antecedentes e formam as saídas difusas da regra [27], [28].

B.1.3 Desfuzzyficação

A avaliação das regras associa potências (intensidade) para cada ação específica na atividade de inferência. A *desfuzzyficação* deve ser executada por duas razões: decifrar o significado de ações vagas (difusas), utilizando funções de pertinência; resolver os conflitos entre ações durante certas condições na avaliação das regras [28]. A desfuzzyficação converte um conjunto difuso de saída de um módulo, em um valor numérico correspondente [27]. Este processo é importante para decifrar o significado das ações difusas, usando funções de pertinência, e resolver conflitos entre ações de competição [28].

B.1.4 Teoria de possibilidade

Seja um grafo $G = (N, M)$, no qual N denota o conjunto de nós aos quais são associadas janelas de tempo *fuzzy*, e M , o conjunto de arcos. Cada arco $(i, j) \in M$ tem um tempo *fuzzy* $\tilde{t}_{ij}$ de transferência e um custo *fuzzy* $\tilde{c}_{ij}$ associados $\in \mathbb{R}^2$. Sejam dois números *fuzzy* $\tilde{t}^1$ e $\tilde{t}^2$, $\tilde{t}^1 \neq \tilde{t}^2$ e $\tilde{c}^1$ e $\tilde{c}^2$, $\tilde{c}^1 \neq \tilde{c}^2$. Podemos dizer que $\tilde{t}^1$ e $\tilde{c}^1$ tem um grau de possibilidade de ser menor do que $\tilde{t}^2$ e $\tilde{c}^2$, respectivamente [23], [37].

$$\tilde{w} = Poss\left(\sum_{ij \in \tilde{t}^1} \tilde{c}_{ij} \leq \sum_{ij \in \tilde{t}^2} \tilde{c}_{ij}\right) = \sup_{(u \leq v)} \min\{\mu_{\tilde{t}^1}(u), \mu_{\tilde{t}^2}(v)\} \quad (\text{B.1})$$

Para encontrar uma solução *fuzzy* utilizando a teoria de possibilidade, teríamos que encontrar todas as soluções que possuem algum grau de possibilidade de ser a solução ótima e comparar estas soluções para obter o grau de possibilidade de cada uma (Okada, 2001). O grau de possibilidade é dado pela fórmula:

$$D_t = \min_{t^k \in t} \{Poss\left(\sum_{ij \in t^k} c_{ij} \leq \sum_{ij \in t} c_{ij}\right)\} \quad (\text{B.2})$$

e o grau de possibilidade de cada arco é dado pela equação

$$D_{ij} = \max_{t^k | ij \in t^k} \{D_{t^k}\}. \quad (\text{B.3})$$

Isso torna o problema de difícil resolução pois, além de ter que enumerar todas as soluções, a comparação entre elas torna o problema NP-completo.

Proposição B.1.1. *Sejam dois números fuzzy normalizados, $\tilde{a}$ e $\tilde{b}$. Temos que, se*

$$\inf_{u | \mu_{\tilde{a}}(u)=1} \{u\} \leq \sup_{v | \mu_{\tilde{b}}(v)=1} \{v\}, \text{ então } Poss(a \leq b) = 1.$$

Demonstração. Seja a definição do grau de possibilidade entre dois número *fuzzy*:

$$Poss(a \leq b) = \sup_{u \leq v} \{\min\{\mu_a(u), \mu_b(v)\}\}$$

Suponha que $u = \inf_{u | \mu(u)=1} \{u\}$ e $v = \sup_{v | \mu(v)=1} \{v\}$, então $u \leq v$ é satisfeito. Com isso, $\mu_a(u) = \mu_b(v) = 1 \implies \min\{\mu_a(u), \mu_b(v)\} = 1$. Portanto, $Poss(a \leq b) = 1$. $\square$

Teorema B.1.1. *Seja t^0 uma solução do grafo crisp G_C , com valores modais de G sendo os custos do grafo crisp. Seja a solução fuzzy t^* , associada à solução crisp t^0 , com custo igual a $\tilde{\omega}^* = \sum_{ij \in \tilde{t}^*} \tilde{c}_{ij}$. Se $\tilde{t}^*$ for a solução ótima do problema, então $\text{Poss}(\sum_{ij \in \tilde{t}^*} \tilde{c}_{ij}) \leq \sum_{ij \in \tilde{t}^k} \tilde{c}_{ij} = 1$, sendo $\tilde{t}^k$ qualquer solução em G .*

Demonstração. Suponha que exista um t^k tal que $\text{Poss}(\sum_{ij \in \tilde{t}^*} \tilde{c}_{ij}) \leq \sum_{ij \in \tilde{t}^k} \tilde{c}_{ij} < 1$. Pela proposição B.1.1, temos que $\omega_{c^*} \leq \omega_{c_k}$. Mas, como ω_c é o menor valor encontrado para o grafo G_C , então $\omega_{c^*} \leq \omega_k$ chegando a uma contradição. Portanto, $\text{Poss}(\sum_{ij \in \tilde{t}^*} \tilde{c}_{ij}) \leq \sum_{ij \in \tilde{t}^k} \tilde{c}_{ij} = 1$ $\square$

Proposição B.1.2. *Sejam $\tilde{a}$ e $\tilde{b}$ dois números fuzzy. Se $\sup\{\text{supp}\{\tilde{b}\}\} < \inf\{\text{supp}\{\tilde{a}\}\}$, então $\text{Poss}(a < b) = 0$.*

Demonstração. Seja um número fuzzy $\tilde{a}$. Temos que $\mu(u) = 0$, se $u \notin (\underline{a}, \bar{a})$, onde $\underline{a}$ e $\bar{a}$ são os limitantes inferior e superior, respectivamente, do número fuzzy. Se $u \in [\underline{a}, \bar{a}]$, então $\mu(u) \geq 0$. Como $\bar{b} = \sup\{\text{supp}\{\tilde{b}\}\} < \inf\{\{\tilde{a}\}\} = \underline{a}$, então

$$u \geq \underline{a} > \bar{b} \geq v \Rightarrow \exists v \in (\underline{b}, \bar{b}) \mid u \leq v \Rightarrow \mu_{\bar{b}}(v) = 0$$

Portanto,

$$\min\{\mu_{\bar{a}}(u), \mu_{\bar{b}}\} = 0, \forall u, v \Rightarrow \text{Poss}(a \leq b) = 0.$$

$\square$

B.1.5 Número *fuzzy*

Um número *fuzzy* triangular $\tilde{\omega}$ pode ser representado por $(\acute{\omega}, \bar{\omega}, \grave{\omega})$, onde as variáveis $\acute{\omega}$ e $\grave{\omega}$ identificam, respectivamente, os limites inferior e superior da base do triângulo, para um grau de pertinência $\mu_A(\omega) = 0$, a variável $\bar{\omega}$ corresponde à altura do triângulo, para $\mu_A(\omega) = 1$

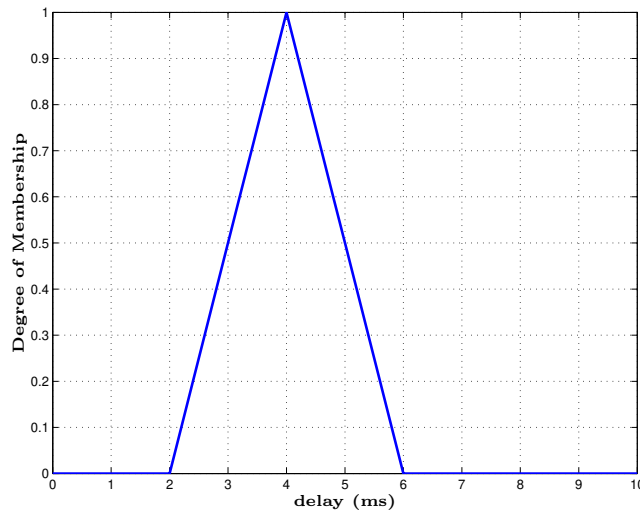
A Figura B.1, ilustra uma restrição de tempo máximo ω em milissegundo (*ms*), para $\acute{\omega} = 2$, $\bar{\omega} = 4$, e $\grave{\omega} = 6$.

Definição B.1. *Número fuzzy é uma quantidade fuzzy A que satisfaz as seguintes condições:*

1. $A(x) = 1$ para exatamente um x ;
2. O suporte $\{x : A(x) > 0\}$ de A é limitado;
3. Os α -cortes de A são intervalos fechados.

Proposição B.2. *O seguinte é válido:*

1. Números reais são números fuzzy.
2. Um número fuzzy é quantidade convexa;
3. Um número fuzzy é semicontínuo superior;

Figura B.1: Número *fuzzy*

4. Se A é um número fuzzy com $A(r) = 1$, então A é não-decrescente em $(-\infty, r]$ e não-crescente em $[r, \infty)$.

Demonstração. : Vide [28].

□

Teorema B.3. Se A e B são números fuzzy então $A + B$, $A \cdot B$ e $A - B$ também são números fuzzy.

Demonstração. : Vide [28]

□

B.2 Controlador *fuzzy*

A teoria *fuzzy* é utilizada para representar modelos de raciocínio impreciso e para tomar decisões em ambientes de incertezas e imprecisões [11]. Essa teoria é uma ferramenta robusta para se representar várias facetas do conhecimento humano [38], [18], [39].

Os Controladores difusos (CDs) são baseados na teoria dos conjuntos difusos introduzida por [11], [33]. Um conjunto difuso A do domínio X é definido pela $\mu_A: X \rightarrow [0, 1]$. A função de pertinência μ_A associa um grau de compatibilidade com conceito expresso por A a cada elemento $x \in X$: se $\mu_A(x) = 1$, diz-se que x é completamente compatível com A ; se $\mu_A(x) = 0$, diz-se que x é completamente incompatível com A ; e se $0 < \mu_A(x) < 1$, então é dito que x é parcialmente compatível com A , com um grau $\mu_A(x)$. Cada elemento x representa uma variável lingüística; X é o universo de discurso; o conjunto difuso A é rotulado por um valor lingüístico (e.g., cheio, vazio, bom, regular, ruim). Como qualquer função, pode ser discreta ou contínua. No caso discreto, o universo de discurso X é representado por um número de amostras aos quais são associados graus de pertinência arbitrários para os conjuntos difusos. Tais valores são intuitivos, baseados na experiência do projetista. No caso contínuo, os graus de pertinência são obtidos para todos os elementos de X usando-se uma função matemática conhecida.

A teoria clássica de conjuntos pode ser vista como um subconjunto da teoria dos conjuntos difusos. Enquanto que na teoria dos conjuntos difusos a compatibilidade do elemento x de um domínio X com um subconjunto A pode ser expressa com graus variando de 0 a 1, na Teoria clássica de Conjuntos, essa compatibilidade pode ser expressa apenas por 0 e 1. A lógica difusa pode ser descrita simplesmente como um cálculo com palavras em vez de números e o controle difuso pode ser descrito simplesmente como um controle com sentenças em vez de equações. As sentenças são regras empíricas [40] no formato:

SE <condições> **ENTÃO** <conclusões>.

Os CDs podem ser usados para controle da QoS uma vez que eles são robustos na presença de perturbações, fáceis de projetar e implementar e eficientes para sistemas que envolvem variáveis contínuas [41]. Em um CD usado para adaptação da QoS, o conjunto de informações de realimentação assume o papel de variáveis lingüísticas (entradas do CD), podendo ser parâmetros da QoS das diferentes camadas da arquitetura do SMD (Sistema de Monitoração de Desempenho). Os valores lingüísticos representam uma granulação dos dados, onde os valores podem pertencer a conjuntos difusos sobrepostos, o que resulta em uma adaptação gradual.

Apêndice C

Gerência de redes por políticas

Neste apêndice são apresentados conceitos sobre gerência de redes por políticas que servem de base para o entendimento e implementação das contribuições apresentadas no Capítulo 2 desta tese.

O conceito de gerência de rede com base em políticas PBNM (*Policy-based Network Management*) é apresentado como solução para o mapeamento de políticas de negócios da empresa, como forma de obtenção de configuração automatizada da rede, reduzindo a necessidade de intervenção do operador [42].

C.1 Gerência de redes

As plataformas de gerência convencionais usam elementos de rede com configurações específicas de baixo nível para cada equipamento com bastante detalhamento. Como exemplo destes mecanismos citamos os sistemas de gerência de redes SDH (*Synchronous Digital Hierarchy*) e SONET (*Synchronous Optical Network*), que são operadas segmento por segmento, requerendo técnicas não automatizadas de engenharia de tráfego. Porém, com o aumento de competitividade entre as operadoras, este tipo de operação nas redes se torna ineficiente, dificultando a manutenção dos níveis de serviço especificados nos SLAs (*Service Level Agreement*) dinâmicos, que exigem um mínimo QoS dos usuários.

A gerência de redes tende a um novo paradigma voltado para os serviços que a rede oferece, e não para os equipamentos que compõem a rede. Com isso, a plataforma de gerência opera em um nível mais alto, possibilitando uma visão geral da rede, com ferramentas automatizadas que realizam operações mais rápidas e eficientes. Tais operações incluem configurações dos mecanismos de QoS, engenharia de tráfego, gerência de SLAs e VPNs (Virtual Private Networks). A necessidade de gerência dinâmica também surge em ambientes de redes corporativas, que estão se tornando quase tão complexas quanto as redes de operadoras. Isto porque essas redes suportam diversos tipos de tecnologias das camadas 1 e 2 do modelo OSI (Open System Interconnection), comutadores com suporte para mecanismos de QoS. Além disso, as operadoras lançam novos tipos de serviços, permitindo ao cliente moldar sua solução de acesso WAN (Wide Area Networks) ao cenário da rede.

C.1.1 Políticas

No contexto de gerência de redes, o padrão DEN-ng define uma política como sendo formada por uma ou mais regras que governam como os recursos da rede serão utilizados. Por sua vez, regras são formadas por condições e ações. Ações devem ser tomadas para modificar o estado dos elementos de rede, caso as condições da política sejam satisfeitas [43], [42].

Um sistema de gerência por políticas transforma essas políticas em alterações dinâmicas na configuração da rede, e oferece solução de controle automático de mecanismos específicos na rede. Políticas oferecem abstrações para simplificar o gerenciamento de vários mecanismos na rede, tais como QoS e mecanismos de segurança. Por outro lado, políticas são uma forma de traduzir contratos de uso da rede, em parâmetros de configuração específicos. Os elementos que compoem a politica são: eventos; condições e ações.

C.1.2 Arquitetura de gerenciamento por políticas

A arquitetura de gerenciamento por políticas que vem sendo proposta por diversos órgãos de padronização, incluindo o IETF (Internet Engineering Task Force), o DMTF (Distributed Management Task Force) e o TMF (Tele Management Forum), possibilita o mapeamento de políticas de alto nível sobre elementos de rede por meio de um modelo de gerência em três camadas, mostrado na Figura C.1, permitindo escalabilidade e consistência na configuração dos elementos [44]. Decisões de políticas são transportadas entre "servidores de políticas, PDPs (Policy Decision Points)" e os PEPs (Policy Enforcement Points) instalados nos dispositivos de rede gerenciados. Vários PDPs podem acessar um repositório comum de informações representando políticas da rede por meio do protocolo LDAP (Lightweight Directory Access Protocol) [45]. A arquitetura de diretórios LDAP possibilita escalabilidade, distribuição, redundância e alto desempenho no armazenamento e busca de informações de gerência da rede.

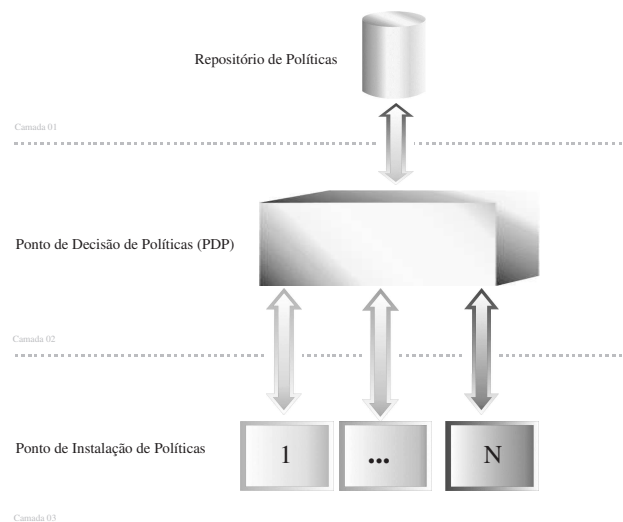


Figura C.1: Arquitetura de gerência por políticas em três camadas.

Em um cenário típico da interação entre as entidades (Repositório de políticas, PDP e PEP),

o PEP precisa verificar a possibilidade de acesso de uma determinada aplicação a um recurso da rede. Neste caso, ele formula um pedido e o envia ao PDP. O PDP retorna a decisão, ficando o PEP responsável pela realização dos procedimentos necessários para que ela seja cumprida. Por meio de uma ferramenta de gerenciamento, as informações de políticas são inseridas em um repositório, o qual é tipicamente implementado por um serviço de diretório como o LDAP.

C.1.3 Ponto de decisão de políticas

Funcionalmente, o ponto de decisão de políticas (PDP) manipula políticas que podem ser obtidas em tempo real por meio de um console de gerência ou de um repositório de políticas. Esses dados são processados juntamente com informações de recursos da rede para tomada de decisões e controle dos dispositivos de rede.

O COPS propõe dois modelos para o controle baseado em políticas, o *outsourcing* e o *provisioning*. No modelo *outsourcing* é usado tipicamente em cenários onde o PEP precisa de uma decisão de política instantânea. Neste caso, ele envia uma requisição ao PDP e aguarda a decisão de política. No modelo *Provisioning* não há correlação direta entre os eventos do PEP e decisões do PDP. Nesta situação, o PDP passa a suprir o PEP de forma pró-ativa ou a pedido do próprio PEP com a informação de política.

No modelo *outsourcing* essas atividades incluem, basicamente, o controle de admissão de novas requisições, onde o PDP aceita ou rejeita a requisição baseada em políticas administrativas encontradas no repositório de políticas.

Já no modelo de *provisioning* novas políticas são inseridas por meio do console de gerência. O PDP decide se a ação relativa a uma dessas políticas deve ser instalada com base nos parâmetros de disponibilidade da rede ou nas condições da própria política. O PDP possui os seguintes módulos: interface de comunicação, interface com um repositório de políticas; módulo de funcionalidades do protocolo COPS (*Common Open Policy Service*), módulo de inteligência (*Policy Rule Engine*). Este módulo é responsável pela análise de regras, condições e ações, determinação de prioridades entre regras e entre condições, ordenação de ações, além de detecção de conflitos.

C.1.4 Ponto de instalação de políticas

O PEP é uma entidade abrigada por roteadores, *firewalls* ou *hosts*, e tem a função de fazer valer as decisões tomadas pelo PDP. O PEP recebe decisões de políticas que traduz em parâmetros específicos de cada elemento de rede e deve relatar ao PDP o sucesso ou falha na instalação de decisões.

C.1.5 Repositório de políticas

Existem varias formas de armazenar políticas em repositórios. LDAP é um protocolo concebido pelo IETF com objetivo de ser padrão de acesso aos diretórios da família ITU X.500 [46] e interface entre sistemas de gerência e aplicações que necessitam de acesso interativo aos diretórios. O mapeamento do modelo de políticas orientado a objetos PCIM (*Policy Core Information Mode*) para um diretório, usando o protocolo LDAPv3, é apresentado em [42].

Apêndice D

Arquivos XML utilizados pelo módulo

Linguagem extensível de formatação ou *Extensible Markup Language* (XML) foi o padrão adotado para provê o formato que descreve os dados estruturados de entrada e saída do modulo fuzzy [47]. O XML é um formato textual e usa marcadores para delimitar os dados, facilitando declarações mais precisas do conteúdo e interações com múltiplas plataformas permitindo, assim, a definição de um número infinito de *tags*, representadas abaixo entre os sinais de menor (<) e maior (>).

D.1 entrada01.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<Variaveis>
  <field nome="MODE_INFERENCE"="MIN"/>
  <field nome="MAX_BW_CLASS"="70"/>
</Variaveis>
```

D.2 entrada02.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<Fuzzy>
  <Ocupacao do link>
    <field linguistic_variable="Baixo" B_a="0" B_b="32" B_c="96"/>
    <field linguistic_variable="Medio" M_a="64" M_b="128" M_c="192"/>
  </Ocupacao do link>
</Fuzzy>
```

```

    <field linguistic_variable="Alto" A_a="160" A_b="224" A_c="256"/>
</Ocupacao do link>
<Perda de pacotes>
    <field linguistic_variable="Baixo" B_a="2" B_b="3" B_c="7"/>
    <field linguistic_variable="Medio" M_a="5" M_b="15" M_c="20"/>
    <field linguistic_variable="Alto" A_a="17" A_b="25" A_c="33"/>
</Perda de pacotes>
<Periodo do dia>
    <field linguistic_variable="Madrugada" M_a="00:00" M_b="03:00" M_c="06:00"/>
    <field linguistic_variable="de manha Cedo" C_a="05:00" C_b="09:00" C_c="12:00"/>
    <field linguistic_variable="a Tarde" T_a="11:00" T_b="15:00" T_c="18:00"/>
    <field linguistic_variable="de Noite" N_a="17:00" N_b="21:00" N_c="24:00"/>
</Periodo do dia>
<Atraso máximo>
    <field linguistic_variable="Baixo" Ba="0" Bb="75" Bc="150"/>
    <field linguistic_variable="Médio" Ha="100" Hb="175" Hc="250"/>
    <field linguistic_variable="Alto" Ma="200" Mb="275" Mc="350"/>
</Atraso máximo>
<Bandwidth>
    <field linguistic_variable="Pouco positivo" Pp_a="2" Pp_b="5" Pp_c="8"/>
    <field linguistic_variable="meio-termo positivo" Mtp_a="6" Mtp_b="9" Mtp_c="12"/>
    <field linguistic_variable="Muito positivo" Mp_a="10" Mp_b="13" Mp_c="16"/>
    <field linguistic_variable="Pouco negativo" Pn_a="-2" Pn_b="-5" Pn_c="-8"/>
    <field linguistic_variable="meio-termo negativo" Mt_a="-6" Mt_b="-9" Mt_c="-12"/>
    <field linguistic_variable="Muito negativo" Mn_a="-10" Mn_b="-13" Mn_c="-16"/>
</Bandwidth>
</Fuzzy>

```

D.3 saida01.xml

```

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE Acoes SYSTEM"Acoes.dtd">
<Acoes>

```

```

<Acao nome="Configuracao de escalonador de filas">
  <Atributos>
    <Atributo nome="Classname" tipo="String" mascara="\verb"ALPHA_NUMERIC_UNDERSCORE""
      faixa="*"
      defaultValue="">
    <Valores></Valores>
  </Atributo>
  <Atributo nome="sched_type" tipo="int" mascara="0" faixa="0..2"
    defaultValue="0">
    <Valores>
      <Valor id="0" descricao="PQ"/>
      <Valor id="1" descricao="CB-WFQ"/>
      <Valor id="2" descricao="WFQ"/>
    </Valores>
  </Atributo>
  <Atributo nome="band_perc" tipo="int" mascara="0" faixa="0..100"
    defaultValue="0">
    <Valores></Valores>
  </Atributo>
  <Atributo nome="target_intf" tipo="String" mascara="*" faixa="*"
    defaultValue="">
    <Valores>\textbf{MAX_BW_CLASS}</Valores>
  </Atributo>
</Atributos>
</Acao>
</Acoes>

```