UNIVERSIDADE FEDERAL DE SANTA CATARINA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Enio Rôvere Silveira

Monitoração e Análise de Tráfego de Rede sob o Paradigma da Lógica Difusa

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação.

Orientador: Mário Antônio Ribeiro Dantas

Florianópolis-SC, Novembro de 2004.

MONITORAÇÃO E ANÁLISE DE TRÁFEGO DE REDE SOB O PARADIGMA DA LÓGICA DIFUSA

Enio Rôvere Silveira

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação, área de concentração em Sistemas de Computação, e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Prof. Dr. Raul Sidnei Wazlawick
Coordenador

Banca Examinadora

Prof. Dr. Mário Antônio Ribeiro Dantas

Prof. Dr. Carlos Barros Montez

Prof. Dr. Carlos Becker Westphall

Prof. Dra. Silvia Modesto Nassar

"Aprendemos a ser filhos depois que somos pais. Só aprendemos a ser pais depois que somos avós...". (Affonso Romano de Sant'Anna)

Dedico este trabalho a minha esposa Márcia, que em todos os momentos desta jornada me incentivou e apoiou, e aos meus filhos gêmeos Augusto e Guilherme que chegaram a pouco tempo e preencheram definitivamente nossas vidas.

AGRADECIMENTOS

Ao meu orientador, Professor Mário Dantas, pela oportunidade de estar concluindo uma pós-graduação e pelo apoio e incentivo à pesquisa.

A Vera Lúcia Sodré Texeira, pelas incansáveis orientações sobre procedimentos relacionados com o programa de pós-graduação.

A todos que de uma forma ou outra manifestaram opiniões e críticas para o aprimoramento da pesquisa.

SUMÁRIO

ÍNDICE DE FIGURAS

ÍNDICE DE TABELAS

RESUMO

As ferramentas de monitoramento de tráfego de rede, proprietárias ou não, distribuídas comercialmente ou de acesso livre, apresentam as mais variadas características. Apesar das facilidades de monitoramento, o processo de diagnóstico, verificado empiricamente, ainda é uma tarefa difícil. A cada momento o administrador de rede, que tem suas decisões apoiadas basicamente em dados, depara-se com situações que exigem tratamento de incertezas.

A presente pesquisa tem como objetivo principal aplicar e avaliar o uso das técnicas da lógica difusa na procura de relações, entre os dados presentes no tráfego de rede, que possibilitem apurar diferentes estados de comportamento da rede no momento em que ocorram.

Os processos baseados em lógica difusa permitem representar expressões qualitativas ou descritivas, incorporadas em frases simbólicas, que são mais naturais e intuitivas, facilitando o processo de automatização do monitoramento de tráfego.

A efetiva utilização de um protótipo implantado em um ambiente de produção de rede comprova a eficácia do uso da lógica difusa no suporte à atividade de monitoração de tráfego de rede.

PALAVRAS-CHAVE: Difuso, Fuzzy, Tráfego de Rede.

ABSTRACT

The traffic monitoring tools of the net, owners or else, distributed comercially ou free access, present a myriad of characteristics. Despite the monitoring easiness, the process of diagnosing, empirically verifying, is still a difficult task. The net administrator who has his decision-making based upon data faces challenging situations on a daily-basis because of the uncertainty of the system.

This research aims at applying and assessing the use of techniques of fuzzy logic in relationship searches, in the data found in the net traffic, which allow the assessment of different behaviours in the net when they happen.

The processes based on fuzzy logic allow the expression of qualitative and quantitative data, incorporated in symbolic phases, which are more natural than intuitive, enabling the automatization process of traffic monitoring.

The effective use of a prototype installed in a net production environment prove the efficacy of the use of fuzzy logic in the support of monitoring activity in net traffic.

KEYWORDS: Fuzzy, Network Traffic.

1. INTRODUÇÃO

1.1 Considerações Iniciais

As redes de computadores se tornaram ferramentas imprescindíveis para o dia-a-dia de pessoas e empresas. Apesar dessa importância, o processo de gerência de redes não é automatizado por completo, isto é, necessita da intervenção humana na tomada de decisões. O monitoramento de tráfego de redes, que é um dos inúmeros processos sob responsabilidade da gerência de redes, não é exceção, ou seja, caracteriza-se como um processo passivo que exige a intervenção de um administrador nas ocorrências de falhas. O conhecimento sobre o comportamento da rede ou dos seus segmentos é importante para se ter noção dos protocolos (aplicativos) que mais consomem recursos, bem como se ter subsídios para futuros planejamentos.

Um aspecto importante da tarefa de monitoramento de rede está relacionada, justamente, com a escolha da melhor forma de executá-lo, pois em uma monitoração intrusiva poderão ser criadas condições artificiais que não retratem a realidade do tráfego, em contrapartida, numa monitoração demasiadamente passiva, poderão haver perdas de informações sobre eventos importantes. Uma proposta para se tentar diminuir o esforço de monitoramento e ao mesmo tempo não interferir nas características naturais do tráfego de rede, passa pela utilização de técnicas que façam uso de uma base de conhecimento heurística associada à utilização de técnicas de amostragem.

A complexidade das redes e a diversidade de hardware e software dificultam qualquer tipo de avaliação, pois a todo momento um novo software ou hardware está integrando e utilizando recursos de rede.

A lógica difusa se credencia como uma técnica eficiente para trabalhar com uma base de conhecimento, estruturada através de um conjunto de regras condicionais, em processos altamente imprecisos ou inexatos. Já a técnica de amostragem utilizada para

estudar comportamento de uma população, baseada em um subconjunto representativo dessa população, se credencia como técnica para diminuir a interferência do processo de monitoramento no tráfego de rede.

Segundo BROWNLEE, MILLS e RUTH (1999), a implantação de infra-estruturas de medições tem sido alvo de pesquisa de grupos que vêm desenvolvendo e propondo soluções que sirvam de suporte aos administradores de rede.

A *RFC 2722 (Request for Comments)* define uma arquitetura para mensuração de pacotes, onde se procura métricas comuns para mensurar fluxos de tráfego de rede.

O presente trabalho, a exemplo de vários outros relacionados com monitoramento de tráfego, tem o objetivo de utilizar o próprio tráfego de rede como fonte de dados para tratar uma diversidade de problemas. Em relação aos trabalhos de CHEN e HUANG (1996), que tem como objetivo o gerenciamento de falhas na rede, a proposta do presente trabalho poderia facilitar a identificação dos possíveis problemas dos vários componentes da rede. Já em relação ao trabalho de NDOUSSE (1997), o protótipo proposto no presente trabalho poderia se constituir em um dos agentes a ser integrado ao sistema distribuído sugerido pelo referido autor. A similaridade existente entre o presente trabalho, o trabalho de DICKERSON (2000 e 2001), e a proposta de AICKELIN e HESKETH (2003), está relacionado com a utilização de uma base de regras para descrever o conhecimento especializado.

1.2 Objetivo

O presente trabalho tem como objetivo principal aplicar e avaliar a utilização da lógica difusa, e ao mesmo tempo fazer uso dos conceitos sobre mensuração de tráfego de rede, na perspectiva de encontrar relacionamentos entre as variáveis que representam o tráfego, possibilitando, dessa maneira, evidenciar diferentes estados de comportamento de uma rede ou segmento, no momento em que ocorram. Não é objetivo do trabalho a medição específica de desempenho ou a definição de uma nova ferramenta de monitoração.

A contribuição da presente dissertação se manifesta, justamente, na tentativa de utilizar técnicas, entre as quais as técnicas difusas, técnicas de *clusterização* e as técnicas de medição, com o objetivo de alcançar, através do próprio tráfego de rede, o conhecimento sobre o perfil da rede de um ambiente computacional.

1.3 Organização do Trabalho

O trabalho foi estruturado em seis capítulos, sendo este o primeiro, onde se procura dar um enfoque inicial ao assunto e demonstrar a estruturação do documento.

O segundo capítulo procura relacionar trabalhos correlatos recentes que façam uso das técnicas utilizadas na presente pesquisa.

O terceiro capítulo descreve detalhadamente as técnicas utilizadas no decorrer dos experimentos, focando principalmente a lógica difusa, principal base teórica do trabalho. As técnicas de *clusterização* foram também descritas devido à necessidade de utilização de técnicas que possibilitassem a atualização de parâmetros do modelo, a partir de um conjunto de amostras de tráfego.

O quarto capítulo caracteriza o problema em estudo e descreve a motivação da pesquisa, além de apresentar o modelo proposto, as ferramentas utilizadas no desenvolvimento do protótipo, e o próprio protótipo implementado.

O quinto capítulo descreve o ambiente utilizado nos experimentos e os resultados auferidos com a pesquisa.

O sexto e último capítulo apresenta as conclusões sobre o trabalho e propõem futuras investigações sobre o tema.

2. AMBIENTES INTELIGENTES

2.1 Sistemas Inteligentes

A Inteligência Computacional caracteriza-se como a área da ciência que estuda técnicas e teorias inspiradas nas características humanas e da natureza, sendo a lógica difusa uma dessas teorias que tenta imitar a habilidade da mente humana quando emprega modos de inferência mais aproximados do que exatos.

A lógica difusa foi originalmente concebida para a representação do conhecimento inerentemente vago ou de natureza lingüística. ZADEH (1965) introduziu formalmente a teoria de conjuntos difusos, apresentando-a, também, como uma abordagem alternativa para o tratamento de incertezas.

Segundo JOY (2000), o futuro pertence aos sistemas biodigitais inteligentes e auto-reprodutivos, vastos e complexos, dos quais os seres humanos serão absolutamente dependentes. Já para BITTENCOURT (2003), o desconhecimento dos princípios que fundamentam a inteligência e o desconhecimento dos limites práticos da capacidade de processamento dos computadores, periodicamente, resultam em promessas exageradas e nas correspondentes decepções.

Em uma conceituação geral pode-se mencionar que os sistemas inteligentes possuem a capacidade de compreensão, reação, percepção, comunicação e aprendizado. Para REZENDE (2003), os sistemas inteligentes utilizam a tecnologia da informação para manipular conhecimentos especializados com benefícios qualitativos e quantitativos.

Segundo GAVRILOV (2002), o problema na criação da teoria de sistemas inteligentes estaria na união de conhecimentos e experiências acumuladas em diferentes áreas da ciência, vinculadas para um aprendizado inteligente e um comportamento

adaptativo. SHADBOLT (2003), dentro de uma visão mais abrangente, define um ambiente inteligente como sendo a convergência de várias áreas da computação.

A essência dos sistemas inteligentes, segundo BARBALHO (2001), está na aquisição de uma base de conhecimento heurística, representada através de um conjunto de regras condicionais. Esses sistemas são capazes de ampliar sua base de conhecimento definida inicialmente através de um processo de aprendizado.

2.2 Lógica Difusa e Redes Neurais

A lógica difusa e as redes neurais figuram entre as técnicas que marcaram a evolução dos sistemas inteligentes. Essas áreas de pesquisa procuram imitar a forma como o ser humano raciocina, baseando-se nos aspectos psicológicos e em processos algorítmicos.

MAMDANI (1975) propôs na década de 70 um método de inferência difusa que por muitos anos foi um padrão na utilização dos conceitos da lógica difusa em processamento do conhecimento.

A modelagem de sistemas baseados na lógica difusa teve como marco os trabalhos apresentados por TAKAGI e SUGENO (1985), onde foram descritos conjuntos de regras difusas que tinham como objetivo representar sistemas lineares, ou seja, um sistema com múltiplas entradas e uma saída poderia ser descrito, segundo BARBALHO (2001), como:

$$R_i: \quad \textbf{SE} \quad x_1 \text{ é } A_{1,i} \ \textbf{E} \ x_2 \text{ é } A_{2,i} \ \ldots \ \textbf{E} \ x_m \text{ é } A_{m,i} \qquad \textbf{ENTÃO} \qquad y_i = a_{0,i} + a_{1,i} x_1 + \ldots + a_{m,i} x_m$$

onde, Ri indica a *i-ésima* regra, x_j (j=1,..m) são as variáveis de entrada, y_i é a variável de saída da *i-ésima* regra, e $A_{1,i}$, $A_{2,i}$, ..., $A_{m,i}$, são os termos lingüísticos associados a essas variáveis e definidos por funções de pertinência.

Uma função de pertinência caracteriza-se como uma função que atribui valores de pertinência difusa para os valores discretos de uma variável em seu universo de discurso. É

importante mencionar que o universo de discurso de uma variável representa o intervalo numérico de todos os possíveis valores reais que a variável possa assumir.

O modelo de TAKAGI e SUGENO (1985) aproxima um sistema não linear através da combinação de vários sistemas lineares.

Com o surgimento das técnicas de reconhecimento de padrões, SUGENO e YASUKAWA (1993) propuseram um modelo difuso onde as conseqüentes das regras difusas (conclusões das regras) são descritas por variáveis lingüísticas, evitando, assim, a utilização de equações lineares expressas no modelo de TAKAGI e SUGENO (1985). Este modelo, segundo BARBALHO (2001), tem a seguinte forma:

$$R_i: \quad \textbf{SE} \quad x_1 \text{ é } A_{1,i} \ \textbf{E} \ x_2 \text{ é } A_{2,i} \ \ldots \ \textbf{E} \ x_m \text{ é } A_{m,i} \qquad \textbf{ENTÃO} \quad y_i \text{ é } B_i$$

onde, Ri indica a *i-ésima* regra, x_j (j=1,..m) são as variáveis de entrada, y_i é a variável de saída da *i-ésima* regra, e $A_{1,i}$, $A_{2,i}$, ..., $A_{m,i}$, e B_i, são os termos lingüísticos associados a essas variáveis e definidos por funções de pertinência.

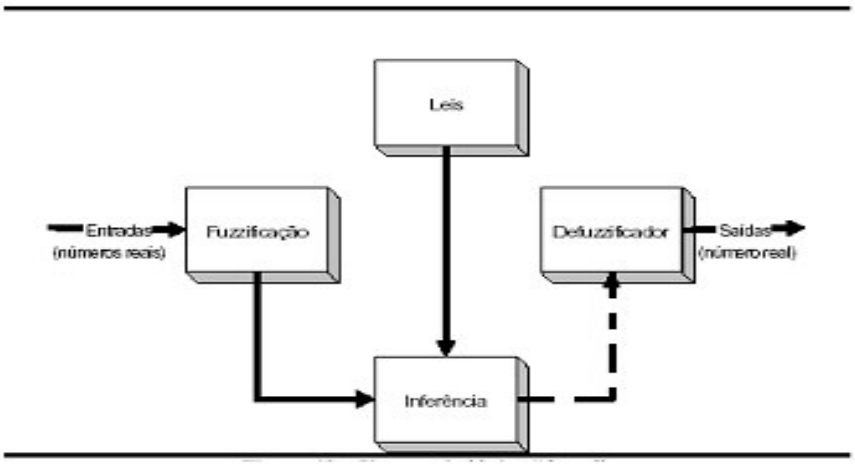


Figura 2.1: Esquema Básico de um Sistema Difuso (HENRIQUES, 1999)

A figura 2.1 ilustra os componentes básicos de um sistema difuso, onde é possível observar os processos responsáveis pela aplicação das técnicas difusas sobre as entradas e saídas do sistema, bem como a base de conhecimento, representada pelo processo chamado de *leis*, que contém os conhecimentos especializados necessários para resolução de

problemas de um domínio específico. Este conhecimento, segundo CHASBEN (2003), consiste de fatos e heurísticas. Os fatos são as informações que estarão disponíveis para serem compartilhadas e atualizadas pelo especialista. As heurísticas são as regras que caracterizam o nível de tomada de decisão do especialista em um domínio. Cabe mencionar que muitas aplicações requerem termos lingüísticos como saída de um processo difuso, tornando o processo de defuzzificação opcional.

De posse dos valores das variáveis de entrada, devidamente mapeadas para domínio difuso, e de sua base de conhecimento, é possível a aplicação do processo de inferência para apuração dos valores de saída.

No contexto da busca por formas de representação do conhecimento, figuram, também, as redes neurais artificiais como modelos matemáticos que, segundo BRAGA, CARVALHO e LUDEMIR (2000), assemelham-se às estruturas neurais biológicas e têm capacidade computacional adquirida por meio de aprendizado e generalização.

As redes neurais artificiais realizam o seu processamento através de estruturas neurais artificiais, chamadas de processadores, que armazenam e processam informação de maneira paralela e distribuída. Cada processador corresponde a um neurônio, descrito muitas vezes como uma estrutura lógico-matemática, que procura simular a forma, o comportamento e as funções de um neurônio biológico (TONSIG, 2000).

Para BRAGA, CARVALHO e LUDEMIR (2000), o aprendizado de uma rede neural artificial está normalmente associado à capacidade de adaptar os seus parâmetros como conseqüência da sua interação com o meio externo.

O processo de aprendizado de uma rede neural tem como característica, basicamente, a ocorrência de estímulos gerados através da apresentação de um conjunto de dados. Os estímulos provocam atualizações nos parâmetros da rede, que em conseqüência levam a gradativas mudanças de comportamento, na procura de uma melhoria de desempenho. Os parâmetros referenciados são justamente os pesos informados através de um vetor chamado de vetor de pesos.

Existem três paradigmas distintos sobre a forma de aprendizado em redes neurais artificiais: aprendizado supervisionado, aprendizado não-supervisionado e aprendizado por reforço.

Segundo entendimento de BRAGA, CARVALHO e LUDEMIR (2000), o aprendizado supervisionado caracteriza-se pela existência de um *supervisor* externo à rede que tem a função de monitorar a resposta para cada vetor de entrada.

O aprendizado não-supervisionado caracterizar-se pela inexistência de saídas esperadas para determinadas entradas.

Por fim, o paradigma por reforço caracteriza-se pela intermediação entre os anteriores, onde existe um *crítico* externo em substituição ao *supervisor*, mas como não existe uma saída esperada, este *crítico,* em vez de retornar um erro de saída da rede, retorna um sinal de reforço ou penalidade, conforme a existência de uma melhoria ou degradação do desempenho dos resultados.

O principal objetivo do processo de aprendizado de redes neurais artificiais, segundo BRAGA, CARVALHO e LUDEMIR (2000), é obter um modelo de boa capacidade de generalização, tendo como alicerce um conjunto de dados.

O aprendizado, que é o resultado do processo de ensinar e aprender, no contexto de uma rede neural artificial, é obtido através de manutenções sucessivas nos parâmetros da rede. Para entender como são realizadas essas manutenções recorre-se ao conceito de treinamento em redes neurais e a utilização do algoritmo de *backpropagation.* O termo *backpropagation* surgiu devido à característica do algoritmo de se basear na retropropagação dos erros para ajuste dos pesos (parâmetros) das camadas de uma rede. O ajuste dos pesos deve modificar a saída da rede de forma que o erro diminua a cada interação. O objetivo do treinamento é encontrar o ajuste ideal dos parâmetros, bem como definir o número de parâmetros e a quantidade de camadas que a rede neural apresentará.

A vantagem de um sistema difuso é estabelecida pela clareza da representação do conhecimento. Já as redes neurais são altamente adequadas para tratar grandes quantidades de dados e classes, mas são *caixas pretas* para seus usuários, pois não é possível extrair conhecimento explícito delas. As redes neurais estão aptas a resolver problemas complexos, no entanto, desconhece-se a forma como o fazem. Sendo assim, os pontos fortes das redes neurais estão caracterizados pela sua capacidade de aprendizagem e estrutura distribuída que permitem implementações de software ou hardware altamente paralelas.

As duas técnicas, mencionadas anteriormente, se fundiram e possibilitaram a criação de uma área de pesquisa chamada de neuro difusa ou *neuro fuzzy*.

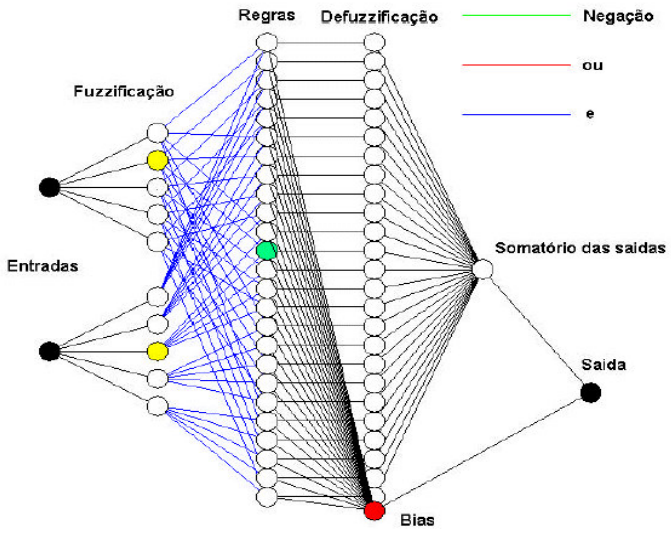


Figura 2.2: Visão de um Sistema Neuro Difuso (HENRIQUES, 1999)

A estrutura da rede criada em um sistema neuro difuso, como ilustrado na figura 2.2, é similar a uma rede neural que mapeia as entradas através de funções de pertinência e de seus parâmetros associados, e daí, através de funções de pertinência da saída e de seus parâmetros associados, obtém a saída (HENRIQUES, 1999).

A contextualização, anteriormente apresentada, tem como objetivo uma concisa descrição das abordagens de pesquisa *difusa*, *neural* e *neuro difusa*. Estas abordagens possuem estreita relação em seus objetivos de representação do raciocínio humano. A partir desse ponto se faz necessário focar a presente revisão bibliográfica em pesquisas que utilizem as técnicas difusas, principal base teórica desse trabalho, como mecanismo responsável pela implementação de *inteligência* na área de redes de computadores.

2.3 Redes de Computadores

Na área de redes constata-se a existência de trabalhos que utilizam a lógica difusa com objetivo de disponibilizar aplicações dotadas de técnicas de aprendizagem, fazendo uso do raciocínio difuso.

Na área de controle de congestionamento de rede, CHRYSOSTOMOU, PITSILLIDES e ROSSIDES (2003) propõem uma atualização do *RED* (*Random Early Discard*), caracterizado como o mais popular algoritmo de controle de congestionamento. Essa proposta está embasada na premente necessidade de criação de estratégias efetivas para controle de congestionamento de tráfego de rede, face aos novos tipos de serviços que estão sendo disponibilizados sob a arquitetura *TCP/IP* (*Transmission Control Protocol/Internet Protocol*). Este algoritmo, originalmente, implementa uma estratégia de descarte aleatório de pacotes sempre que o tamanho da fila de *buffers* suplanta um valor estabelecido. Na proposta, o gerenciamento desses *buffers* é realizado por um controlador difuso que executa o descarte de pacote de uma maneira dinâmica, baseado no estado da rede.

Existem estruturas que estabelecem maneiras de uma aplicação informar a qualidade de serviço, *QoS* (*Quality of Service*), necessária para alocar um recurso qualquer de rede. Essa informação, sob a forma de um valor, deve ser precisa, impedindo, assim, informações subjetivas. No caso das redes ATM, por exemplo, é necessário informar parâmetros precisos para taxa de células e taxa de perdas do circuito. Esse tipo de alocação de *QoS* não permite uma política de especificação do tipo: alocar um serviço se existir um bom descarte de pacotes, ou mesmo, parar um serviço se a taxa de erro é muito alta. Diante

desse contexto, VIEIRA e WESTPHALL (2001) propõem um controlador difuso para indicar os valores de *QoS* através de um conjunto difuso. De acordo com essa proposta, a aplicação indicaria um intervalo de valores aceitáveis, com diferentes níveis de qualidade, para que o gerenciador de *QoS* pudesse alocar as condições necessárias ao atendimento dos requisitos da aplicação.

Uma arquitetura de gerenciamento de sistema para melhorar *QoS* é encontrada em FERNANDEZ, PEDROZA e REZENDE (2003). Nesse trabalho é proposta uma metodologia para viabilizar o mapeamento da política de gerenciamento em parâmetros de um controlador difuso. Esse controlador implementa mecanismos dinâmicos de provisionamento, coordenado pela política de gerenciamento de sistema.

Em CARBONELL, JIANG e PANWAR (2002) encontra-se caracterizada a implementação de um controlador difuso para o controle de fluxo de rede. Nessa proposta o campo *windows size*, que compõem o protocolo *TCP* e faz o controle de transmissão em uma conexão *TCP*, é definido de acordo com o comportamento da rede, inferido através de uma série de variáveis difusas.

O algoritmo *Fuzzy C-Means* é utilizado como ferramenta para reconhecimento de padrões através da utilização da técnica de *clusterização*. Segundo JONHSON e WICHERN (1992), os métodos de aglomeração (*clustering*) podem ser caracterizados como qualquer procedimento estatístico que, utilizando um conjunto finito e multi-dimensional de informações, classifica seus elementos em grupos restritos e homogêneos internamente, permitindo gerar estruturas agregadas significativas e desenvolver tipologias analíticas.

O problema de *clusterização* pode ser tratado segundo diferentes abordagens, entre elas, a abordagem convencional, na qual cada objeto deve ser classificado única e totalmente em uma determinada categoria, e a abordagem alternativa (difusa), mais flexível, na qual um objeto pode ser classificado em várias categorias, com diferentes graus de associação a cada uma delas.

Segundo BEZDEK (1981), o uso da teoria de conjuntos difusos torna-se conveniente uma vez que grande parte das categorias comumente encontradas e empregadas na *clusterização* possui limites vagos.

As próximas ilustrações, adaptadas de BEZDEK e PAL (1992), mostram o problema da *clusterização* graficamente. Na primeira, os objetos do conjunto não estão classificados. Na segunda, estão classificados de acordo com alguma técnica tradicional de *clusterização*. Nesse exemplo, o conjunto universo é um conjunto de objetos X={o1,o2,…,o17,o?} e o objetivo é classificar esses objetos como um dos três tipos de frutas, a saber: *M = maçã*, *L= laranja*, *P = pêra*. Note que os rótulos são atribuídos aos objetos conforme sua similaridade na forma, sendo que o objeto de forma elíptica, rotulado como ***o?*** , representa uma anomalia nos dados.

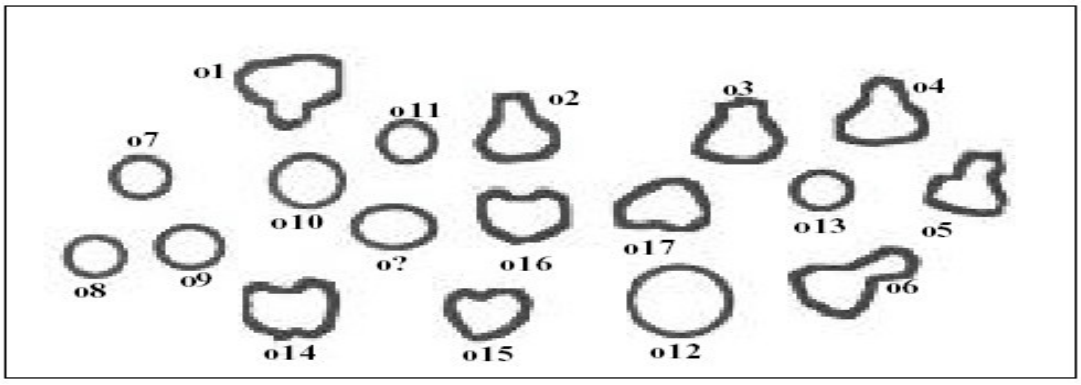


Figura 2.3: Problema de *Clusterização* (BEZDEK & PAL, 1992)

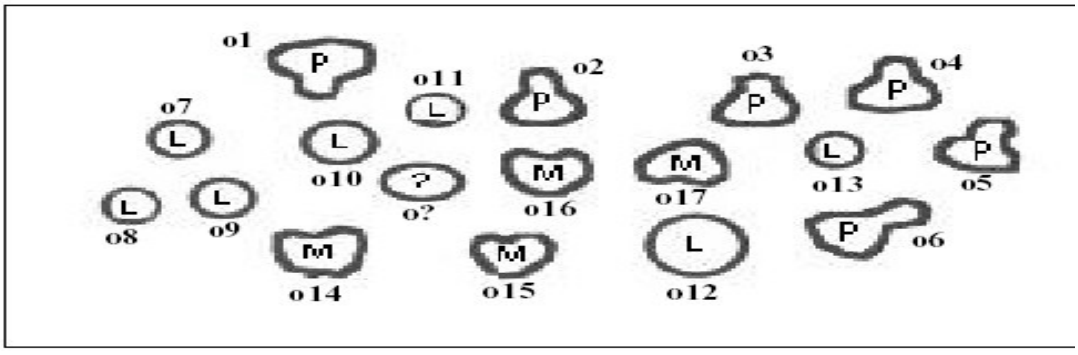


Figura 2.4: *Clusterização* Tradicional (BEZDEK & PAL, 1992)

A classificação dos elementos pode ser realizada de acordo com a abordagem clássica ou convencional, na qual cada elemento pertence totalmente a uma única classe, ou de acordo com abordagens alternativas, como a difusa, onde um elemento pode pertencer a várias classes, com diferentes graus de pertinência.

As ilustrações 2.5 e 2.6 demonstram em forma tabular a utilização da abordagem tradicional e difusa na resolução do problema acima formulado. Note que na primeira tabela, o objeto pertence (células com valor 1) ou não pertence (células com valor 0) a uma das categorias de frutas. Na segunda tabela, tem-se na célula um valor de pertinência do objeto a uma das categorias de frutas, conforme sua similaridade de formato.

Partição																		
	o1	*o2*	*o3*	*o4*	*o5*	*o6*	*o7*	*o8*	*o9*	*o10*	*o11*	*o12*	*o13*	*o14*	*o15*	*o16*	*o17*	*o?*
P	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	1
M	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0
Total	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figura 2.5: Abordagem Clássica de *Clusterização* (BEZDEK & PAL, 1992)

Partição																		
	o1	*o2*	*o3*	*o4*	*o5*	*o6*	*o7*	*o8*	*o9*	*o10*	*o11*	*o12*	*o13*	*o14*	*o15*	*o16*	*o17*	*o?*
P	0.9	1	0.7	0.7	0.9	1	0	0	0.1	0.1	0.1	0.1	0	0.1	0.1	0.2	0.2	0.15
L	0	0	0.1	0	0	0	0.8	0.8	0.8	0.7	0.8	0.6	0.8	0.1	0.1	0.1	0.1	0.60
M	0.1	0	0.2	0.3	0.1	0	0.2	0.2	0.1	0.2	0.1	0.3	0.2	0.8	0.8	0.7	0.7	0.25
Total	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figura 2.6: Abordagem Difusa de *Clusterização* (BEZDEK & PAL, 1992)

Com objetivo de encontrar similaridades entre as aplicações que trafegam nas redes de computadores, LAMINEN, KOIVISTO e HONKANEN (2002) aplicaram o algoritmo *Fuzzy C-Means* (BEZDEK, 1981) e obtiveram resultados efetivos na criação de *clusters* de aplicações. Esses *clusters* combinados com a topologia de rede, objetivam subsidiar as tomadas de decisões em relação à administração e gerenciamento de redes.

Na área de segurança da informação, a maior preocupação está relacionada à identificação de uma tentativa de acesso não autorizado quando a mesma está em andamento.

Devido à complexidade das redes de computadores, suas ligações com softwares e outros hardwares, é muito difícil avaliar um sistema de detecção de invasão.

O rápido crescimento e diversidade de ataques dificultam, segundo LAUREANO (2002), a atualização de ferramentas *IDS (Intrusion Detection System)*. Estas ferramentas acabam por agir somente sobre alguns tipos de ataques mais conhecidos, ou comparam o tráfego de rede buscando padrões conhecidos de ataques. Para LAUREANO (2002), o número de tentativas de invasão seria menor se as ferramentas de *IDS* fossem devidamente configuradas, e se existisse quantidade suficiente de pessoas com o conhecimento técnico adequado para configurar e trabalhar com esse tipo de ferramenta.

O *FIRE* (*Fuzzy Recognition Engine*) é um *IDS,* baseado na lógica difusa, que utiliza o tráfego de rede e técnicas de mineração de dados (*Data Mining*), com objetivo de emitir alarmes para os administradores de ambiente sempre que uma invasão está em andamento (DICKERSON & DICKERSON, 2000 e 2001).

GOMEZ e DASGUPTA (2002) propuseram um método para gerar um classificador difuso, usando algoritmos genéticos, cujo objetivo é evitar a geração de falsos alarmes, tão comuns de ocorrer nas ferramentas *IDS* convencionais.

FLOREZ, BRIDGES e VAUGHN (2002) desenvolveram técnicas utilizando mineração difusa (*fuzzy data mining),* com objetivo de extrair padrões de comportamento das tentativas de invasões.

AICKELIN e HESKETH (2003) propõem uma atualização na base de regras do software *snort.* O software *snort* é utilizado na detecção de invasão em redes. Na proposta dos referidos autores, o conjunto de regras de condição desse software, conhecida

como *snort rules*, sofre um processo de *difusão automática* com objetivo de perceber novos ataques, baseado em seu conjunto de regras.

A gerência de redes é uma atividade que, segundo WESTPHALL (1991), por sua natureza, deve possuir pelo menos as seguintes características:

- Deve produzir o mínimo possível de tráfego nos meios de comunicação pelos quais deve trafegar, caso contrário, a gerência chegaria à conclusão que é ela a causadora de congestionamento;

- Deve ser rápida suficiente para que possa detectar em tempo hábil os problemas das redes.

Um gerenciamento pró-ativo dos recursos de rede é alcançado através de funções de gerenciamento que devem estar contidas em diversos componentes da rede, permitindo o diagnóstico, a prevenção e a reação para problemas (WESTPHALL 1991,WESTPHALL 1996).

Um sistema de gerência pró-ativa pode ser associado com a área de sistemas especialistas. Em NETO (1998) foi proposto um modelo de gerenciamento pró-ativo que utilizava informações sobre as aplicações que trafegavam em segmentos monitorados. Para análise do perfil de funcionamento desses segmentos foi aplicada a técnica de séries temporais (CHATFIELD 1984) e implementada uma série de funções de gerenciamento.

Técnicas de inteligência artificial foram também utilizadas no reconhecimento de problemas de rede, mais especificamente no congestionamento de rede, em ROCHA (1997).

Na atualidade, muitos fabricantes propõem soluções inovadoras, mas desconsideram o parque computacional existente, ou seja, o gerenciamento de rede deve se adequar ao ambiente da empresa (DANTAS, 2002).

No contexto de tráfego de rede, deve-se mencionar a necessidade da criação de técnicas para mensuração e avaliação de tráfego, a partir de coletas periódicas. Porém, o volume de dados coletados é muito grande, dificultando sobremaneira a análise e o gerenciamento. As ferramentas utilizadas para auxiliar neste trabalho são bastante limitadas e não permitem uma análise mais detalhada de diversas variáveis importantes que poderiam ser correlacionadas a fim de disponibilizar um diagnóstico mais preciso da rede.

A implantação de infra-estruturas de medições tem sido alvo de pesquisa de diversos grupos que vem desenvolvendo e propondo soluções que sirvam de suporte aos administradores da rede. O *CAIDA (Cooperative Association for Internet Data Analysis*), por exemplo, tem como uma das suas ferramentas o *Coralreef* que vem com a proposta de unir, em um só pacote, recursos de coleta, classificação e visualização dos dados. Os conceitos sobre as medições por *fluxo de tráfego* são bastante úteis quando se deseja identificar as características e distribuição do tráfego das redes, aliado a uma redução do volume de dados a ser coletado.

Pode-se conhecer melhor uma rede quando recolhe-se dados sobre o que está sendo *enviado/recebido,* e quando utiliza-se essas informações para tomar decisões sobre os controles que precisam ser desenvolvidos.

3. LÓGICA DIFUSA , TRÁFEGO DE REDE E CLUSTERIZAÇÃO

3.1 Lógica Difusa

A lógica difusa caracteriza-se como uma metodologia muito eficiente quando se necessita trabalhar com informações inexatas, imprecisas, incompletas através de uma sistemática rigorosa. A primeira descrição matemática da lógica difusa foi feita por L. A. Zadeh (1965).

Para NASSAR (2001), os dois principais métodos para tratar a incerteza são o método simbólico e o método numérico. O método simbólico trata as incertezas através de regras de inferência, enquanto que o método numérico propaga as incertezas numericamente através das inferências e combinações de evidências.

O aspecto principal da lógica difusa, segundo WEBER e KLEIN (2003), está na sua capacidade de capturar com clareza e concisão as várias nuances dos conceitos psicológicos utilizados pelos seres humanos em seu raciocínio usual.

Na literatura mundial são encontrados os termos *fuzzy* e *crisp* para descrever respectivamente números e conjuntos difusos, e números e conjuntos ordinários.

O atributo da bivalência significa a utilização de dois valores: algo é verdadeiro ou não. A lógica clássica de Aristóteles faz do atributo da bivalência um marco histórico em nossa cultura ocidental. Espera-se sempre que uma determinada afirmação seja verdadeira ou falsa.

Segundo SHAW e SIMÕES (1999) a bivalência, que fundamenta a ciência da computação na utilização da lógica *booleana*, está enraizada no modo de pensar e no

comportamento ético. O mundo real é analógico e não digital, com um infinito espectro de opções.

A linguagem possui uma série de expressões verbais, carregadas de imprecisões, onde comumente se utilizam as mesmas palavras com significados diferentes, ou seja, as palavras não representam uma idéia única.

Através da lógica difusa é possível manipular dados numéricos e lingüísticos simultaneamente, construindo-se, assim, um mapeamento não linear de um conjunto de dados de entrada.

A modelagem difusa descreve o comportamento de um sistema através do uso da teoria dos conjuntos difusos e da linguagem natural baseada na lógica difusa.

Para WEBER e KLEIN (2003), um modelo difuso é caracterizado por um conjunto de regras que expressam a relação entre as variáveis do sistema. Cada regra representa a descrição local da dinâmica do sistema e é composta por uma parte antecedente (condição da regra) e uma parte conseqüente (conclusão da regra).

3.1.1 Teoria dos Conjuntos Difusos

A linguagem natural, apesar de vaga e ambígua, é o veículo de comunicação humano, e parece apropriado o uso da teoria matemática, que trabalha com vaguesa e ambigüidade, para expressar e interpretar o caráter lingüístico da nossa linguagem. A comunicação humana envolve termos naturais que são freqüentemente vagos, imprecisos, ambíguos e incertos. Os conjuntos difusos são empregados na descrição numérica desses termos.

ZADEH (1978) apresenta a noção de conjunto difuso como uma generalização da noção de conjunto ordinário. Tal como os conjuntos ordinários, os conjuntos difusos também são definidos sobre um domínio (universo de discurso), mas diferem daqueles porque não possuem uma fronteira claramente definida. Nesses conjuntos, os elementos

podem ter pertinência parcial. Estas características ficam bem evidenciadas através da figura 3.1.

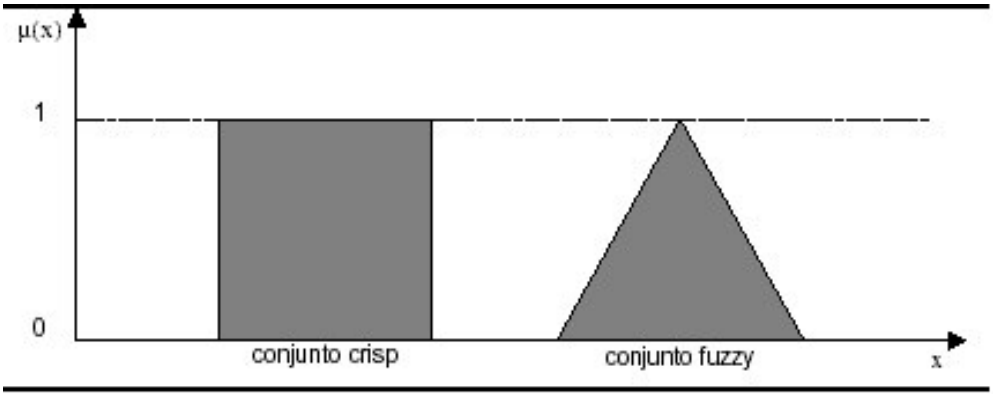


Figura 3.1: Exemplos Conjuntos *Crisp* e *Fuzzy* (Difuso) (HENRIQUES, 1999)

O conceito central na teoria de conjuntos difusos é o da função de pertinência, que representa, numericamente, o grau de certeza com que um elemento pertence a um conjunto.

A função de pertinência é uma função que mapeia cada elemento do universo de discurso em um valor entre 0 e 1, representando o grau de pertinência do elemento ao conjunto. Nos conjuntos ordinários, essa função assume o valor 1 para os elementos pertencentes ao conjunto e 0 para os elementos não-pertencentes, o que faz com que estes conjuntos possam ser considerados casos particulares dos conjuntos difusos.

Um número *crisp* x pode ser representado, segundo HENRIQUES (1999), por exemplo, pela função de pertinência $\mu_A(x)$, sendo A um conjunto *crisp*:

$$A = \{ (\mu_A(x), x) \mid \mu_A(x) = \begin{vmatrix} 0 \text{ se } x \notin A \\ \\ 1 \text{ se } x \in A \end{vmatrix} \}$$

Um número difuso x pode ser representado, segundo HENRIQUES (1999), por exemplo, pela função de pertinência $\mu_A(x)$, sendo A um conjunto difuso:

$A = \{ (\mu_A(x), x) \mid \mu_A(x) \in [0,1] \}$

Se um elemento denominado genericamente por x, do universo de discurso U, pertence a um conjunto difuso A, então este conjunto difuso pode ser definido através da seguinte relação:

$A = \{ (\mu_A(x), x) \mid \mu_A(x) \in [0,1] \}$, onde $\mu_A(x)$ é a chamada função de pertinência.

Conforme demonstrado, os elementos pertencentes a um conjunto difuso são especificados através de um par, constituído do elemento propriamente dito e de seu grau de pertinência ao conjunto.

A definição de conjunto difuso pode ser aceita como uma extensão da definição do conjunto clássico ordinário. Se os valores da função de pertinência $\mu_A(x)$ ficam restritos a 0 e 1, então A representa um conjunto clássico.

Para HENRIQUES (1999), existem outros modos de se representar um conjunto difuso, como por exemplo, a representação ilustrada na seguinte equação:

$$A = \begin{cases} \sum_{x_i \in x} \frac{\mu_A(x_i)}{x_i} & \text{se X é discreto} \\ \int_x \frac{\mu_A(x)}{x} & \text{se X é contínuo} \end{cases}$$

A função de pertinência, na teoria difusa, tem como objetivo *modelar* ou *refletir* o conhecimento que se tem acerca da intensidade com que um elemento pertence a um conjunto difuso.

Na verdade, a função de pertinência é definida com base na intuição ou experiência que se tem sobre o contexto e sobre as variáveis envolvidas.

Segundo AGUIAR e JUNIOR (1999), na síntese de conjuntos difusos, existe extrema flexibilidade na escolha da forma geométrica da função que descreve os vários graus de pertinência dos elementos de um universo de discurso em relação a um dado subconjunto difuso. As funções mais encontradas na prática para representar os graus de pertinência são as triangulares, trapezoidais, gaussianas e sigmoidais.

Segundo WEBER e KLEIN (2003), as triangulares e gaussianas aparecem normalmente em casos onde se deseja exprimir pertinência crescente à esquerda e decrescente à direita. As funções trapezoidais podem ser usadas em situações similares ao uso das formas triangulares e gaussianas, sendo que no caso das trapezoidais deve existir a necessidade de expansão da faixa de pertinência máxima. As funções sigmoidais são usualmente aplicadas em casos onde se busca delimitar pontos extremos, a partir dos quais a pertinência se mostra constante.

Para AGUIAR e JUNIOR (1999), os modelos difusos são tolerantes a aproximações relativamente grosseiras, tanto estruturais quanto no tocante a síntese de relações de pertinência. Isto significa ter bom desempenho onde não se tenha conseguido descrever com excelente precisão o comportamento dos objetos sob análise.

Na representação gráfica dos conjuntos difusos temos o eixo x, que descreve o universo de discurso do sistema, e o eixo y, que mapeia uma faixa de 0 a 1 para representar o grau de participação de um elemento ao conjunto difuso do contexto em estudo (ROSS, 1995). Como todas as informações contidas em um conjunto difuso estão descritas pela função de pertinência, torna-se importante descrever as características gerais dessas funções.

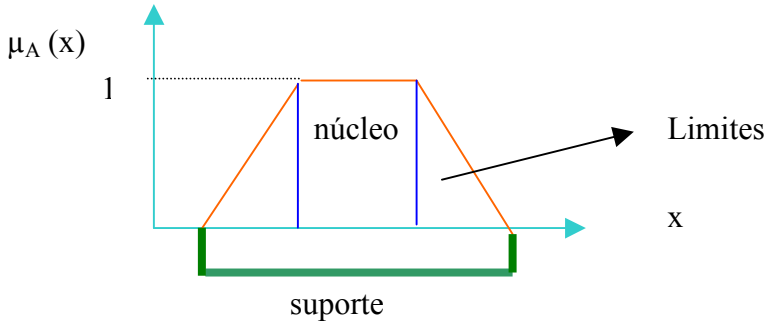


Figura 3.2: Características da Função de Pertinência

Conforme ilustrado na figura 3.2, o núcleo de uma função de pertinência para um conjunto difuso é a região do universo de discurso que é caracterizada pela pertinência completa do elemento, ou seja , o núcleo descreve os elementos do universo que possuem $\mu_A(x) = 1$. O suporte de uma função de pertinência para um conjunto difuso caracteriza a região do universo que possui pertinência diferente de zero, ou seja, descreve os elementos do universo onde $\mu_A(x) > 0$.

A teoria difusa possui uma série de conceitos importantes a serem aplicados nos modelos baseados nessa técnica, entre os quais destacam-se :

- **Conceito de α - cut e strong α - cut :**

Dado um conjunto difuso A definido em X e um número $\alpha \in [0,1]$, os conjuntos *α - cut* e *strong α - cut* caracterizam-se como conjuntos *crisp* definidos respectivamente por :

$$^{\alpha}A = \{ x | \mu_A(x) \geq \alpha \} \quad \text{e} \quad ^{\alpha+}A = \{ x | \mu_A(x) > \alpha \}$$

Dado, por exemplo, um conjunto difuso A com os elementos x_1, x_2, x_3, x_4, x_5 e seus respectivos graus de pertinência 0,5; 0,4; 0,7; 0,8; 1, e o valor 0,4 como α - *cut* e *strong* α - *cut* (corte) , temos:

$$A = \{ 0,5/ x_1 + 0,4/ x_2 + 0,7/ x_3 + 0,8/ x_4 + 1/ x_5 \}$$

$$^{0,4}A = \{1/ x_1 + 1/ x_2 + 1/ x_3 + 1/ x_4 + 1/ x_5 \} => \{ x_1, x_2, x_3, x_4, x_5\}$$

$$^{0,4+}A = \{1/ x_1 + 0/ x_2 + 1/ x_3 + 1/ x_4 + 1/ x_5 \} => \{ x_1, x_3, x_4, x_5\}$$

Os conjuntos α - *cut* e *strong* α - *cut* tem como função, na teoria difusa, o retorno para um conjunto *crisp,* a partir de um conjunto difuso, sendo que o *strong* α - *cut* é mais restritivo nesse retorno.

- **Conceito de cardinalidade |A| de um conjunto difuso finito A:**

$$|\mathbf{A}| = \sum \mu_A (x), \text{ onde } x \in X.$$

- **Conceito de conjunto difuso convexo:**

Segundo SHAW e SIMÕES (1999), a convexidade carrega informações sobre a conectividade interior e o formato do número difuso, como forma de prevenir *buracos* entre os limites de uma função de pertinência. Um conjunto difuso convexo é todo conjunto em que para dois pontos quaisquer de sua função de pertinência, o segmento de reta que os une também pertença ao conjunto. Na figura 3.3 são ilustradas duas funções de pertinência A e B que representam, respectivamente, a convexidade e não-convexidade de um conjunto difuso, informando, por exemplo, um valor de corte de 0,8 (α-*cut* = 0,8).

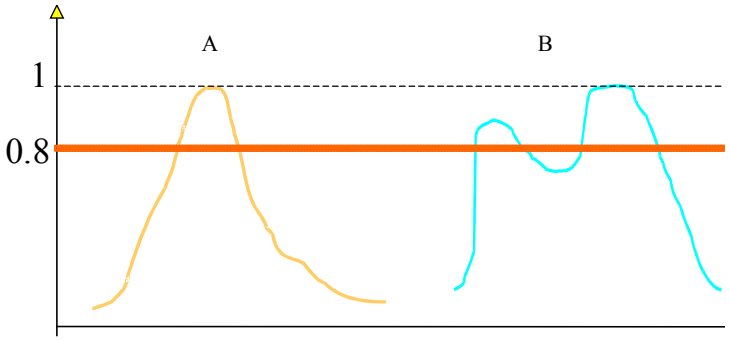


Figura 3.3: Características de Conjuntos Convexos (NASSAR, 2002)

- **Conceito de conjunto difuso normalizado**:

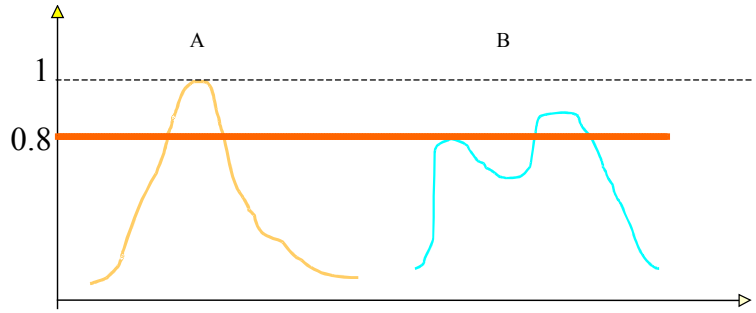


Figura 3.4: Características de Conjuntos Normalizados (NASSAR, 2002)

Um conjunto difuso é normal, segundo NASSAR (2002), se existe pelo menos um elemento que tenha pertinência igual a 1. Na figura 3.4 são ilustradas duas funções A e B, caracterizando, respectivamente, a normalidade e subnormalidade.

3.1.2 Operações com Conjuntos Difusos

As operações realizadas com os conjuntos difusos são semelhantes às operações dos conjuntos clássicos, com o diferencial de serem realizadas em termos de funções de pertinência. A seguir estão descritas as operações difusas padrão.

Considerando-se dois conjuntos difusos A e B, definidos em X, com o uso das funções de pertinência μ_A e μ_B, a função de pertinência da união ou disjunção de A com B, segundo SILVA (2001), é definida ponto a ponto para todos os elementos $x \in X$, tal que:

$$\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}$$

De modo análogo, a função de pertinência da intersecção ou conjunção de A com B, segundo SILVA (2001), é definida ponto a ponto para todos os elementos $x \in X$, tal que:

$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}$$

A função de pertinência do complemento de A, segundo SILVA (2001), é definida ponto a ponto para todos os elementos $x \in X$, tal que:

$$\mu_{\overline{A}}(x) = 1 - \mu_A(x)$$

Os resultados dessas operações entre os conjuntos difusos A e B são demonstrados graficamente através da figura 3.5.

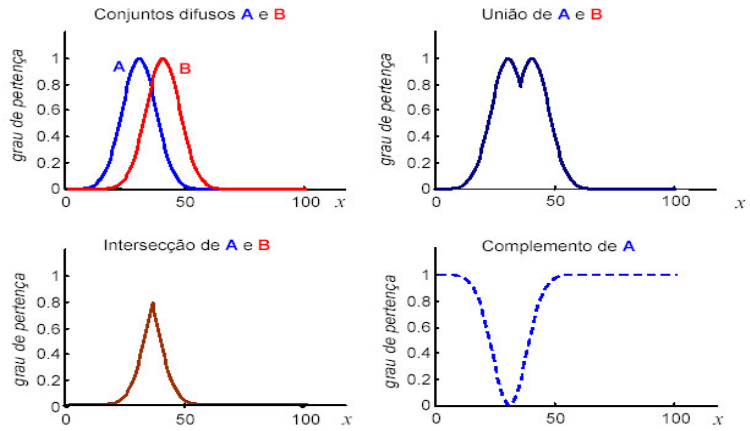


Figura 3.5: Representação Gráfica de Operações Difusas (SILVA, 2001)

Aos operadores lógicos união e intersecção, uma vez que estabelecem uma ligação entre conjuntos difusos, dá-se o nome genérico de *conectivas*.

O produto cartesiano de n conjuntos difusos A_1, A_2, … A_n, definidos respectivamente em X_1, X_2, …, X_n, é o conjunto difuso definido no espaço cartesiano $X_1 \times X_2 \times \ldots \times X_n$, que, segundo SILVA (2001), tem a seguinte função de pertinência:

$$\mu_{A_1 \times \cdots \times A_n}(x) = \min\{\mu_{A_1}(x), \cdots, \mu_{A_n}(x)\}$$

Na lógica tradicional as operações com conjuntos são essencialmente as operações *booleana* possibilitadas pelos conectivos *E, OU e NÃO*. Na lógica difusa existem diversos operadores para realizar as operações lógicas, podendo-se dividi-los, segundo AGUIAR e JUNIOR (1999), em duas classes: as normas triangulares, chamadas de *normas-t*, que são uma extensão da intersecção, e as normas duais, chamadas de *normas-s* ou *co-normas*, que são uma extensão da união.

Segundo SHAW e SIMÕES (1999), as *normas-t* podem ser representadas através das operações: de intersecção, do produto algébrico e de produto drástico.

Alguns exemplos de *normas-t* para $\mu_A, \mu_B \in [0,1]$:

- Intersecção → $\mu_A(x)$ *normas-t* $\mu_B(x) = \min(\mu_A(x), \mu_B(x))$

- Produto Algébrico → $\mu_A(x)$ *normas-t* $\mu_B(x) = \mu_A(x).\mu_B(x)$

- Produto Drástico → $\mu_A(x)$ *normas-t* $\mu_B(x) =$

$$\begin{cases} \mu_A(x) \text{ quando } \mu_B(x) = 1 \\ \mu_A(x) \text{ quando } \mu_A(x) = 1 \\ 0 \quad \text{quando } \mu_A(x), \mu_B(x) < 1 \end{cases}$$

Segundo SHAW e SIMÕES (1999), as *co-normas* podem ser representadas através das operações: de união e da soma algébrica.

Alguns exemplos de *co-normas* para $\mu_A, \mu_B \in [0,1]$:

- União → $\mu_A(x)$ *co-normas* $\mu_B(x) = \max(\mu_A(x), \mu_B(x))$

- Soma Algébrica → $\mu_A(x)$ *co-normas* $\mu_B(x) = \mu_A(x) + \mu_B(x) - \mu_A(x).\mu_B(x)$

Para AGUIAR e JUNIOR (1999), o modo como são combinados os conjuntos difusos determina a qualidade e a abrangência dos processos de inferência. Os vários tipos de operadores possibilitam formas alternativas de *ponderar*, *compensar* ou *suavizar* o comportamento, por vezes agudo, verificados na utilização dos operadores lógicos de união e intersecção.

3.1.3 Variáveis Lingüísticas

As variáveis lingüísticas, segundo BARBALHO (2001), cumprem, na lógica difusa, o mesmo papel que as variáveis numéricas nos modelos matemáticos convencionais, sendo que na lógica difusa os valores podem assumir conceitos expressos em linguagem natural.

Uma variável difusa é definida com um certo número de funções de pertinência, cada qual representando um espectro de valores que a variável possa assumir.

Uma variável lingüística tem como incumbência fazer a transição entre o domínio de valores das variáveis convencionais para o domínio de valores lingüísticos das variáveis difusas.

Para GANOULIS (1994), uma variável lingüística é, portanto, o instrumento da lógica difusa que permite quantificar e manipular conceitos qualitativos, sendo especialmente úteis para caracterizar incertezas em problemas onde as variáveis ou as relações funcionais não são bem definidas.

Os conjuntos difusos podem ser utilizados para quantificar o significado da linguagem natural. Essas variáveis lingüísticas são representadas nos conjuntos difusos para possibilitar uma aproximação com o mundo real. As variáveis lingüísticas não possuem valores precisos, representam um espectro de valores.

Segundo BASTOS (1994), as variáveis lingüísticas são variáveis cujos valores não são números, mas palavras ou frases na linguagem natural, conforme demonstra a figura 3.6.

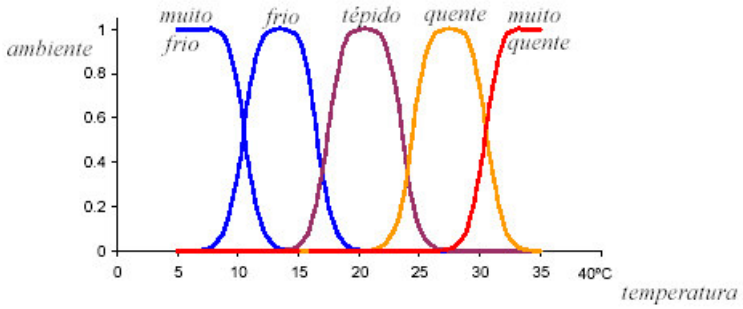


Figura 3.6: Representação da Variável Lingüística Temperatura Ambiente (SILVA, 2001)

3.1.4 Raciocínio Difuso

Pode-se dizer que raciocínio difuso é uma metodologia de inferência que utiliza ferramentas e conceitos da lógica difusa para atingir objetivos e conclusões. Esta metodologia é a base da construção dos sistemas difusos que, segundo LEE (1990), diferem dos sistemas convencionais por se aproximarem do pensamento humano e da linguagem natural.

O modelo de um sistema difuso é estruturalmente similar ao utilizado por sistemas especialistas que também mantém, fundamentalmente, uma base de conhecimento e a lógica para tomada de decisões. Para AGUIAR e JUNIOR (1999), no modelo difuso as regras que descrevem o comportamento de um especialista na área em estudo, atuam em paralelo e segundo sua compatibilidade com as entradas fornecidas ao sistema.

A figura 3.7 ilustra os componentes principais de um sistema difuso que possui a mesma estrutura de um sistema especialista, onde o motor de inferência e a base de conhecimento compõem a inteligência do sistema.

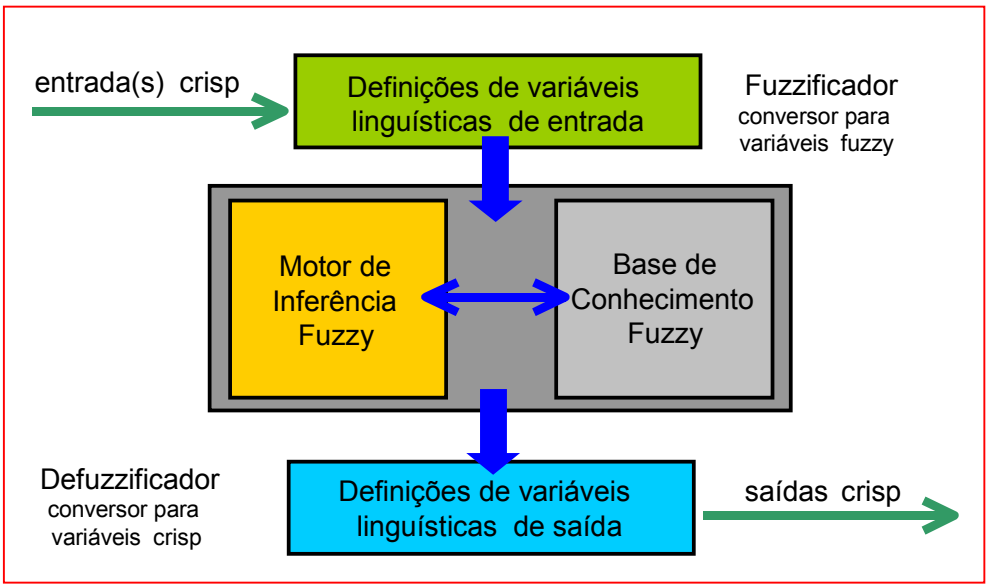


Figura 3.7: Visão de um Típico Sistema Difuso (XEREZ, 1999)

Uma das principais componentes de um sistema difuso são as regras. Essas regras são representadas através de implicações lógicas *SE-ENTÃO,* assumindo a forma:

Se *x é K então y é Z*, onde *K* e *Z* são valores lingüísticos definidos por conjuntos difusos sobre um universo de discurso. Nomeia-se *x é K* de antecedente e *y é Z* de conseqüente. Essas regras condicionais difusas foram definidas por MAMDANI (1975), que as utilizou para capturar modos imprecisos de raciocínio em sistemas de controle.

TAKAGI e SUGENO (1985) propuseram um tipo de regra que utiliza igualmente proposições difusas para descrever o antecedente (condições), mas os conseqüentes (conclusão da regra) eram descritos com expressões não difusas. Tipicamente, estas regras utilizam expressões que são funções lineares das variáveis lingüísticas antecedentes, descritas, segundo BARBALHO (2001), como:

Se x é K e y é Z então $z = p*K + q*Z + r$, onde x e y são variáveis antecedentes, K e Z são termos lingüísticos associados a estas variáveis, e p, q e r são constantes.

Ambos os tipos de regras são normalmente utilizados em modelagem de sistemas de controle. As regras do tipo TAKAGI e SUGENO (1985), embora mais eficientes computacionalmente, são mais complexas e menos intuitivas.

Os sistemas de inferência difusa servem para representar a dependência existente entre as variáveis independentes (entrada) e dependentes (saída) em um modelo. A base desses sistemas é um conjunto de regras condicionais difusas. Um sistema de regras difusas pode ser formalmente descrito, segundo BARDOSSY e DUCKSTEIN (1995), como:

$$IF \quad x_1 \text{ é } A_{i,1} \otimes x_2 \text{ é } A_{i,2} \otimes \ldots \otimes x_k \text{ é } A_{i,k}$$
$$THEN \quad y_1 \text{ é } B_{i,1} \otimes \ldots \otimes y_m \text{ é } B_{i,m} \;,\; i = 1,\ldots,n \;,$$

onde, $x_1, x_2, \ldots, x_k$ e $y_1, \ldots, y_m$ são, respectivamente, variáveis representando as entradas e saídas; $A_{i,k}$ e $B_{i,m}$ são os termos lingüísticos associados a estas variáveis e definidos pelas funções de pertinência $\mu A_{i,k}$ e $\mu B_{i,m}$; e $\otimes$ representam os operadores lógicos *E* ou *OU*.

Para JANG (1993), o conjunto de regras e o conjunto das funções de pertinência, definidas para cada um dos termos lingüísticos, compõem a base de conhecimento do sistema. Dependendo do tipo de regra utilizada, o sistema será referido como sistema do tipo MAMDANI ou SUGENO.

De posse dos valores das variáveis de entrada, o sistema de regras difusas pode ser avaliado ou inferido, e os valores das variáveis de saída conhecidos. Nesse processo, as regras são inferidas paralelamente. A interpretação ou inferência de cada regra consiste na avaliação das proposições antecedentes (premissas), seguida da aplicação das conseqüentes.

As seguintes etapas, adaptadas de BARBALHO (2001), são aplicadas em um raciocínio difuso:

1. O conteúdo (valores numéricos) de cada variável de entrada é submetido à função de pertinência correspondente, resultando o grau de pertinência de cada valor nos termos lingüísticos correspondentes (*fuzzificação*);

2. Uma função é aplicada aos graus de pertinência obtidos para cada proposição antecedente, produzindo um valor numérico, entre 0 e 1, que representa o grau em que a expressão condicional da regra é satisfeita (grau de aplicabilidade da regra (g)). As funções utilizadas nesse processo dependem do operador lógico usado nas proposições, sendo mais comumente adotadas as funções :

   - $g = \min (\mu_{A1}(x_1) , \mu_{A2}(x_2))$ ou $g = \mu_{A1}(x_1).\ \mu_{A2}(x_2)$;

   - $g = \max (\mu_{A1}(x_1) , \mu_{A2}(x_2))$ ou $g = \max (\mu_{A1}(x_1) + \mu_{A2}(x_2) - \mu_{A1}(x_1) .\ \mu_{A2}(x_2))$

3. É a etapa chamada de implicação, em que as conseqüentes das regras, cujas condições são satisfeitas,ou seja, cujo grau de aplicabilidade é maior que zero, são calculadas com base nos seus respectivos graus. Nos casos em que as regras possuam mais de um conseqüente, todas as conseqüências são igualmente afetadas pelo grau de aplicabilidade. O processo de implicação consiste na modificação dos conjuntos difusos associados com as conseqüentes da regra. Nos sistemas de regras do tipo MAMDANI (1975), onde a resposta do processo é um conjunto difuso para cada regra, dois métodos são utilizados nessa etapa, sendo que no primeiro método o conjunto difuso é truncado em um nível correspondente ao grau de aplicabilidade da regra, e no segundo, o conjunto difuso é reduzido proporcionalmente a este grau. Nos sistemas de regras do tipo TAKAGI e SUGENO (1985) o processo de implicação se restringe à avaliação das

equações não difusas, sendo que nesse processo os graus de aplicabilidade obtidos nas avaliações dos antecedentes são utilizados somente no cálculo da resposta final do sistema;

4. Quando um sistema de regras é avaliado para um conjunto de valores informado nas variáveis de entrada, encontram-se, em geral, mais de uma regra aplicável. Nesse caso, as conseqüências obtidas pela inferência destas regras devem ser combinadas para produzir uma resposta única do sistema para cada variável de saída. Entre os vários métodos de combinação de conseqüências disponíveis menciona-se:

    - A função máxima que corresponde à operação de união dos conjuntos difusos.

    - A função soma, onde a resposta é obtida pela soma das funções de pertinência que representam os conjuntos difusos.

    Para situações onde os conjuntos difusos obtidos pela combinação dos conseqüentes não são suficientes como respostas para sistema, os esquemas a seguir, que compõem o processo chamado de defuzzificação, podem ser utilizados, onde resposta é :

    - O elemento correspondente ao centro de área ou centróide da área definida pelo conjunto difuso;

    - O primeiro elemento ou a média entre aqueles elementos com máximo grau de pertinência;

    - O elemento de maior grau de pertinência.

A ilustração 3.8, retirada do manual do software MATLAB®, procura retratar graficamente as etapas mencionadas anteriormente.

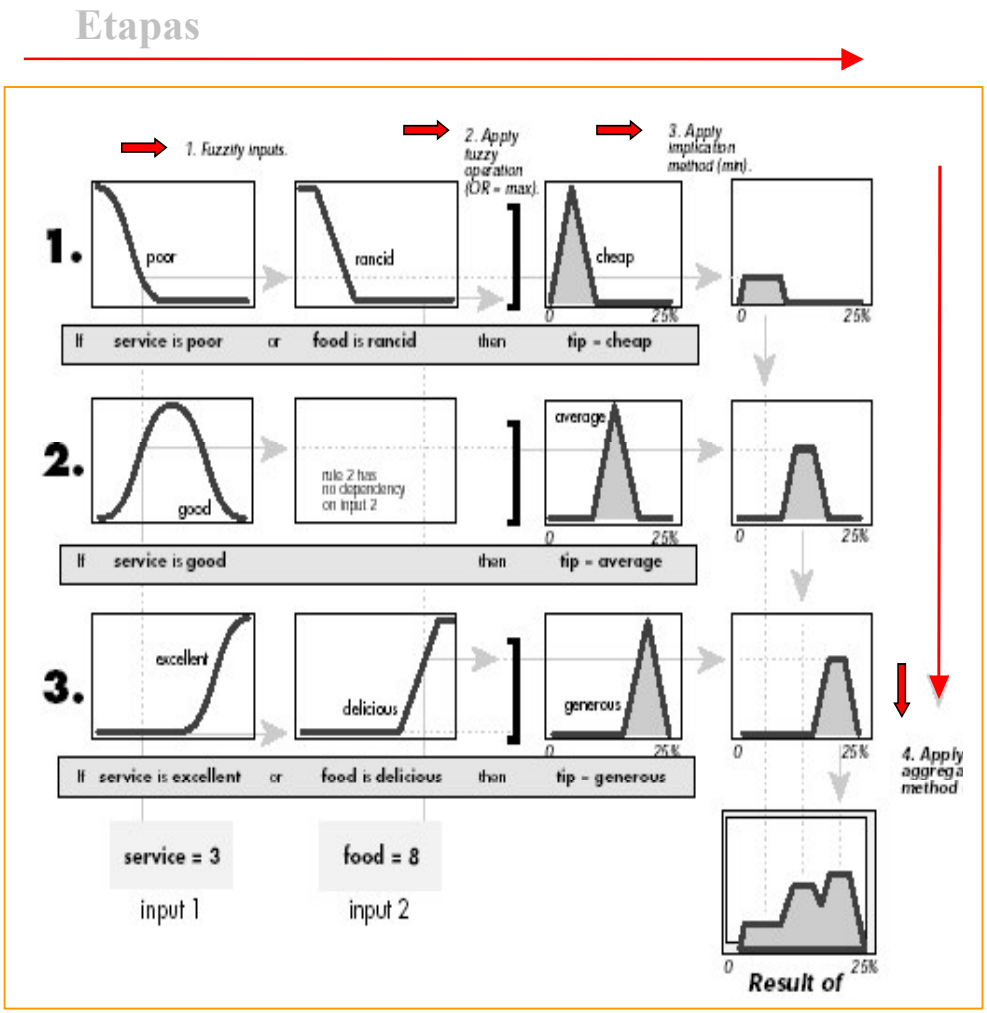


Figura 3.8: Representação do Raciocínio Difuso (Manual do software MATLAB®)

A vantagem dos sistemas difusos fica, então, estabelecida pela adequada representação do conhecimento, na forma de regras *SE-ENTÃO* dos antecedentes e conseqüentes difusos.

3.2 Modelos Difusos

O aspecto principal na modelagem difusa está relacionado com a construção da sua base de conhecimento.

No processo de construção das regras, o conhecimento e os dados são estruturados sobre a forma de regras. Essas regras, quando conhecidas por um especialista, podem ser transcritas diretamente no modelo. Nos casos da inexistência de um especialista, deve-se construir uma estratégia de modelagem que permita uma definição inicial e, posteriormente, um ajuste com base no conjunto de dados das variáveis de entrada e de saída, buscando representar, da melhor forma possível, segundo algum critério especificado, a relação entrada/saída desejada.

O processo de construção do conjunto de regras em um modelo difuso é, muitas vezes, subjetivo, pois se estará tentando encontrar um conjunto de regras que, quando avaliadas, produzam as respostas esperadas para o sistema.

O processo de modelagem difusa envolve as seguintes etapas:

3.2.1 Definição das Variáveis do Modelo

Não existe método para identificação das variáveis relevantes, sendo geralmente utilizada uma análise dos dados disponíveis. O objetivo dessa etapa é determinar as variáveis de entradas e as variáveis de saída que descrevam o comportamento do objeto em estudo. Caso exista um especialista no domínio da aplicação, segundo BARBALHO (2001), o processo de definição é rapidamente concluído. Nos casos de exigência de uma análise de dados, o processo se torna complexo, motivado pelo fato de se identificar, eventualmente, a necessidade de determinadas transformações nas variáveis presentes no conjunto de dados para fornecer melhor explicação do comportamento das variáveis.

3.2.2 Particionamento do Universo de Discurso das Variáveis

O particionamento do universo de discurso das variáveis tem como objetivo a representação das variáveis numéricas como variáveis lingüísticas.

A definição do número de partições influencia diretamente na qualidade do modelo proposto, pois muitas partições resultarão em grande quantidade de parâmetros a serem ajustados, enquanto que um número muito pequeno de partições pode resultar em modelos que não consigam representar a relação presente no conjunto de dados. Normalmente, é realizada uma divisão do domínio em intervalos de tamanhos iguais. É importante mencionar a existência de métodos de reconhecimento de padrões, entre os quais os algoritmos de *clusterização*, que podem ser utilizados na identificação automática das partições.

Ao final dessa etapa deve-se possuir um nome ou termo lingüístico adequado a cada uma das diversas partições de cada variável de entrada e de saída do modelo.

3.2.3 Definição das Funções de Pertinência e Termos Lingüísticos

Os modelos difusos são tolerantes a aproximações, ou seja, a atribuição de funções de pertinência para cada partição de domínio das variáveis é também um processo intuitivo.

Várias formas de funções tem sido sugeridas, mas, até o momento, ainda não se comprovou maior eficácia de uma forma em relação às demais. WEBER e KLEIN (2003) sugerem as formas triangulares e gaussianas em casos onde se deseja exprimir pertinência crescente à esquerda e decrescente à direita. Já as funções trapezoidais podem ser utilizadas em situações similares as formas anteriormente descritas, com o diferencial da exigência de alargamento da faixa de pertinência máxima. Por fim, esses autores sugerem a utilização das formas sigmoidais e semi-trapezoidais nos casos em que se busca delimitar pontos extremos, onde, a partir dos quais, a pertinência se torna constante.

As funções de pertinência, segundo SHAW e SIMÕES (1999), não precisam ser simétricas ou igualmente espaçadas e ainda, pode-se ter, para cada variável, um conjunto de funções com formatos e distribuições diferentes.

Uma vez especificadas as formas das funções de pertinência associadas a cada partição de domínio das variáveis de entrada e saída, inicia-se a fase de escolha dos parâmetros de cada função. Quando a forma triangular é utilizada, esses parâmetros correspondem, em geral, aos valores extremos mínimo e máximo da partição, para quais são atribuídos graus de pertinência zero, e para média dos valores encontrados na partição é atribuído grau de pertinência 1. No caso de funções trapezoidais, também são atribuídos o grau zero para os valores mínimo e máximo da partição, com outros dois parâmetros definidos, em geral, subjetivamente.

Os parâmetros de todas as funções de pertinência geralmente são ajustados durante o processo de teste do sistema de regras. O ajuste desses parâmetros é realizado, em geral, num processo de tentativa e erro, sendo que atualmente existe uma tendência de utilização da técnica neuro difusa, onde as redes neurais artificiais em combinação com sistemas de regras difusas têm permitido ajustes automáticos. Para EVSUKOFF e ALMEIDA (2003), a otimização dos parâmetros que definem as funções de pertinência pode prejudicar o entendimento dos termos associados aos conjuntos difusos.

3.2.4 Construção das Regras

Nessa etapa as proposições antecedentes e conseqüentes de cada regra são descritas considerando as possíveis interações entre as variáveis selecionadas. Um sistema de regras estará completo, segundo SHAW e SIMÕES (1999), se ele puder responder satisfatoriamente a todas as ocorrências possíveis envolvendo o fenômeno modelado. O número de regras deve ser determinado cuidadosamente a fim de evitar o excesso de parâmetros, fazendo com que o modelo perca a capacidade de generalização (*overfiting)*.

Para alcançar um equilíbrio entre a eficiência das interações e os números de parâmetros, é recomendável iniciar um modelo com um número pequeno de regras. Por

outro lado, existem algoritmos com o objetivo de minimizar o conjunto de regras, mas é sempre importante avaliar o sistema com e sem a regra, comparando as respostas obtidas. Somente nos casos de respostas idênticas haverá possibilidade de extinção da regra.

3.2.5 Técnicas Difusas para Ajuste do Modelo

A tarefa que consome mais tempo no processo de desenvolvimento de um sistema difuso é o ajuste dos parâmetros das funções de pertinência. Este ajuste envolve um processo de tentativa e erro, em que o conjunto de regras é sistematicamente inferido com base nos valores informados para as variáveis de entrada e os resultados, dessa inferência, avaliados através de comparações com os valores obtidos para as variáveis de saída.

A inexistência de um método bem definido que indique as modificações a serem feitas nas funções de pertinência, para que as repostas produzidas pelo sistema se aproximem das saídas desejadas, dificulta e retarda essa etapa do processo.

Segundo BARBALHO (2001), em geral, há muitos graus de liberdade na escolha de parâmetros, pois uma mesma alteração na resposta do sistema pode ser obtida com a alteração dos parâmetros das funções de pertinência do antecedente ou da conseqüente das regras envolvidas, ou ainda, com a simples alteração da forma dessas funções.

O modelo neuro difuso, que é uma combinação de redes neurais com as técnicas difusas, surge como alternativa, segundo HENRIQUES (1999), para o ajuste automático dos parâmetros das funções de pertinência, através da utilização dos algoritmos de aprendizagem das redes neurais. Isto significa que a principal intenção de uma abordagem neuro difusa, nesse contexto, é criar ou aperfeiçoar um sistema difuso, automaticamente, por meio dos métodos existentes nas redes neurais. Apesar da utilização das técnicas neurais, ressalta o autor que, o modelo deve ser interpretável em termos de regras difusas, porque ele é baseado nas técnicas difusas que procuram refletir conhecimento vago.

Em continuidade ao objetivo de obter ajuste automático de parâmetros, o processo de treinamento, quando baseado no paradigma de aprendizado supervisionado neural,

consiste num processo iterativo, em que, a cada iteração o conjunto de valores de entrada é processado pela rede, produzindo saídas que são, então, comparadas às saídas desejadas. Sempre que as saídas produzidas pelo modelo não corresponderem, dentro de uma precisão especificada, às saídas desejadas, o conjunto de parâmetros da rede é atualizado, segundo critérios que depende do algoritmo de aprendizado utilizado. O algoritmo *backpropagation*, comumente empregado nos modelos neuro difusos, utiliza o método do gradiente descendente para atualização dos parâmetros.

O modelo neuro difuso mais difundido na literatura chama-se *ANFIS* (*Adaptive Neuro Fuzzy Inference System*). O *ANFIS* está disponibilizado no utilitário de lógica difusa *Fuzzy Logic Toolbox*. Esse utilitário constitui-se num conjunto de funções desenvolvidas no ambiente de computação numérica *MATLAB®*, que permite a implementação de sistemas de regras difusas. O utilitário possui interface gráfica, conforme a ilustração da figura 3.9, que o torna de fácil utilização, embora todas as funções possam ser chamadas através de linha de comando.

A implementação do módulo *ANFIS* tem como objetivo, segundo HENRIQUES (1999), atender a situações de inexistência de métodos para transformar conhecimento ou experiência em base de regra, ou ainda, nas situações de ajuste dos parâmetros das funções de pertinência. A estrutura da rede criada é similar a uma rede neural, que mapeia as entradas, através de funções de pertinência e de seus parâmetros associados, e daí, através de funções de pertinência da saída e de seus parâmetros associados, para a saída.

Através de um algoritmo de aprendizado pode-se construir automaticamente, segundo HENRIQUES (1999), um sistema de regras difusas, a partir de um conjunto de observações. O modelo proposto se baseia numa arquitetura, também intitulada de *ANFIS*, que consiste em uma classe de redes neurais funcionalmente equivalentes aos sistemas de inferência difusa. Os sistemas de regras resultantes utilizam regras difusas do tipo proposto por TAKAGI e SUGENO (1985). O modelo ajusta automaticamente os parâmetros das funções de pertinência de forma a maximizar o desempenho do modelo na representação da relação entrada/saída desejada. O ajuste dos parâmetros, conforme descrito anteriormente, é

realizado num processo de treinamento que utiliza o algoritmo de propagação *backpropagation*, ou uma combinação desse com métodos de mínimos quadrados (algoritmo de aprendizagem híbrido).

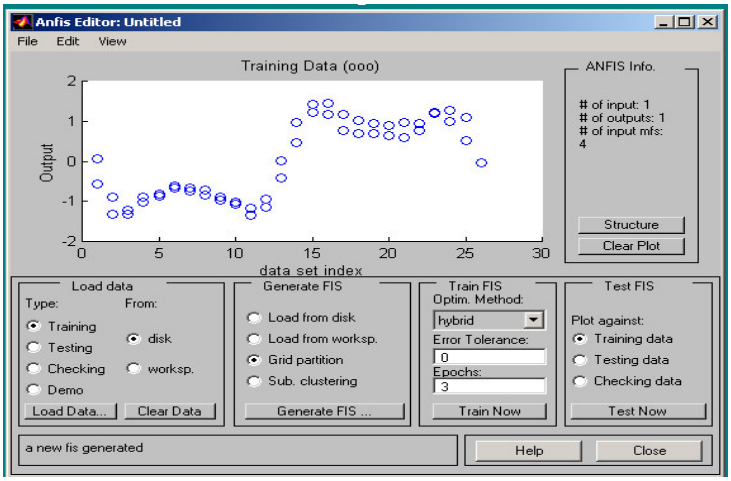


Figura 3.9: Editor ANFIS do Software MATLAB®

O aproximador usado pelo *ANFIS* utiliza etapas para a construção de um modelo difuso, onde primeiramente são definidas quais e quantas serão as funções de pertinência, e em seguida são informados os dados de treinamento de entrada e saída para realização do treinamento e teste do sistema.

O sistema será bem modelado, se o conjunto de treinamento for suficientemente representativo, ou seja, se existir uma razoável distribuição de valores que torne possível interpolar todos os valores necessários para a operação do sistema.

O *ANFIS* agrupa os dados de treinamento de entrada e saída em *clusters* com objetivo de identificar grupos naturais de dados, de maneira a produzir uma representação concisa e significativa do comportamento do sistema.

As duas técnicas implementadas MATLAB® são: (1) *Fuzzy C-Means (FCM)*, onde é necessário indicar o número de *clusters* a serem utilizados no agrupamento dos dados. Nessa técnica os centros dos *clusters* são buscados de maneira iterativa, e essa iteração baseada na minimização da função objetivo, que representa a distância entre qualquer dado

ao centro do *cluster* . Se não existir idéia de quantos *clusters* deva ser utilizado, faz-se uso do (2*) Subtractive Clustering* , que estima o número de *clusters* usando o próprio *FCM* (Software MATLAB®).

Para SHAW e SIMOES (1999), o *ANFIS*, apesar de ser uma ferramenta eficiente, possui como limitações a necessidade de utilização do modelo TAKAGI e SUGENO (1985), a necessidade de se construir modelos de saída única, e a impossibilidade de se extrair conhecimento sobre a forma como o modelo foi gerado, devido à utilização das redes neurais.

O próximo assunto, intitulado como tráfego de redes, tem como objetivo dar continuidade a apresentação da base teórica do presente trabalho.

3.3 Tráfego de Redes

Em PEARSON (1996) encontram-se estudos sobre o comportamento das redes, particularmente, sobre o tráfego em *ATM (Asynchronous Transfer Mode).* Nesse trabalho há demonstrações de que o tráfego *Ethernet* não segue uma distribuição de *Poisson* no processo de chegada de pacotes. Uma hipótese aparentemente mais aceitável, mas não comprovada, caracteriza o tráfego *Ethernet* como um modelo fractal com auto-similaridade, aplicando-se essa hipótese, inclusive, para o tráfego entre as redes *Ethernet* locais e a Internet.

A medição do tráfego de rede é uma prática usual na caracterização de uma rede. Para ANGELIS (2003), hoje, se verifica um esforço no sentido de padronizar ou estabelecer uma linguagem comum para medição de tráfego de redes.

O uso do conceito de fluxos, como base para medições e para distribuição de tráfego em uma rede, é previsto em LAI (2001). Nesse trabalho o autor sugere a confecção dos fluxos a partir das informações sobre os endereços *IP,* número das portas fonte e destino, protocolos, tipos de serviço, marcação de tempo de início e fim do fluxo, contadores de pacotes e octetos.

3.3.1 Mecanismos de Controle de Tráfego de Redes

Os mecanismos de controle de tráfego podem ser convencionalmente classificados em mecanismos por conversação e por agregação.

O controle de tráfego por conversação utiliza mecanismos que controlam cada conversação como um fluxo em separado. A conversação caracteriza-se por conter todo o tráfego gerado por aplicações pares operacionalizadas em equipamentos diferentes. Esta conversação fica evidenciada na arquitetura *TCP/IP*, por exemplo, através do endereço *IP* fonte e destino, porta e protocolo utilizado. Para este tipo de controle, os recursos são alocados, também, por conversação, sob a perspectiva da aplicação, existindo a garantia dos recursos alocados, independentemente de efeitos provocados por outros tipos de tráfegos.

Para MELO (2001), os controles de tráfego por conversação tendem a melhorar a qualidade de serviço, mas impõem maior carga ao equipamento de rede, que deve manter um estado de independência e aplicar processamento independente para cada conversação.

No controle de tráfego agregado, um conjunto de tráfego de múltiplas conversações é classificado como um mesmo fluxo e controlado de forma agregada. Esse tipo de controle caracteriza-se pela redução da carga de processamento e manutenção dos dispositivos de rede.

Segundo MELO (2001), a qualidade de serviço percebida por uma aplicação é influenciada pelo efeito do tráfego de outras conversações que foram agregadas no mesmo fluxo. Como resultado, a qualidade do serviço percebida por uma aplicação pode não ser muito consistente.

Os Serviços Integrados (*Integrated Services* ou *IntServ*) foram projetados para prover extensões ao modelo de entrega de tráfego de melhor esforço, utilizado na arquitetura *TCP/IP*. Esses serviços foram planejados para dar tratamento especial a certos tipos de tráfego e prover um mecanismo de escolha entre múltiplos níveis de serviços de entrega. O *IntServ*, por exemplo, é baseado na reserva de recursos.

Embora estas abordagens tenham reduzido os impactos da atividade de gerência sobre o tráfego da rede, existe, ainda, a necessidade de se possuir mecanismos que permitam identificar o perfil do tráfego gerado.

Segundo SPECIALSKI (2002**),** é interessante conhecer quais são os protocolos (aplicações) que mais consomem recursos. Hoje, comenta a autora, é comum acreditar que o maior tráfego seja gerado pelos serviços de informação disponíveis via *Web*. Entretanto, outros serviços essenciais como o terminal virtual (*telnet*), transferência de arquivos (*ftp*) e o correio eletrônico (*e-mail*) estão sendo utilizados a todo o momento.

3.3.2 Medições de Fluxo de Tráfego

A necessidade de uma garantia mínima para que as aplicações possam ser executadas com um desempenho satisfatório em um ambiente de rede, está desencadeando pesquisas em várias áreas de conhecimento. A medição de tráfego de rede, exemplificada pela figura 3.10, se caracteriza como uma dessas áreas que vem ganhando importância à medida que as aplicações se tornam mais exigentes com relação a recursos e desempenho de redes.

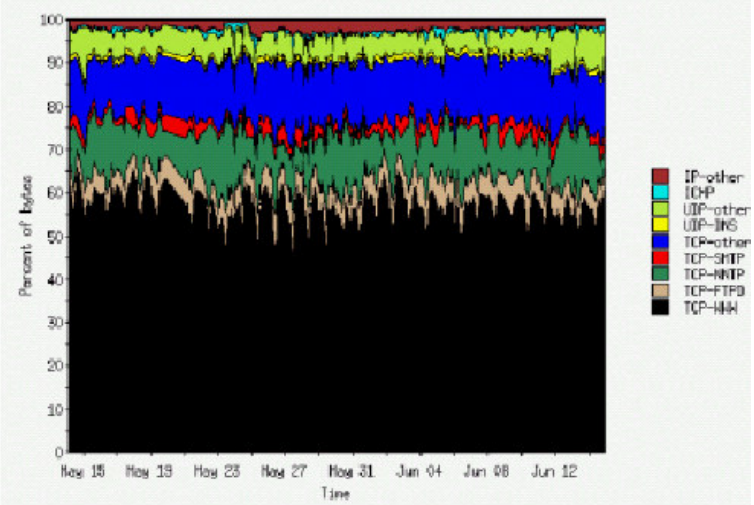


Figura 3.10: Representação da Medição de Tráfego de Aplicação (RICARDO & CARRAPATOSO, 2003)

Os altos custos dos meios de comunicação de dados e a influência dos sistemas abertos fazem com que o núcleo da rede não seja de solução única, mas caracterizado por um conjunto de redes e pela diversidade de equipamentos e tecnologias. Desta forma, cada rede ou grupo de interesse monta uma infra-estrutura de medições de acordo com a sua conveniência de custos, equipamentos e necessidades específicas, resultando em diversas ilhas de informação.

Diversos esforços estão sendo realizados em torno de uma melhor caracterização do tráfego de rede. Segundo BROWNLEE, MILLS e RUTH (1999), a implantação de infra-estruturas de medições tem sido alvo de pesquisa de grupos que vêm desenvolvendo e propondo soluções que sirvam de suporte aos administradores de rede.

O *CAIDA (Cooperative Association for Internet Data Analysis)* vem com a proposta de unir em um só pacote os recursos de coleta, classificação e visualização dos dados. Através dos vários resultados experimentais obtidos e de intensas pesquisas, o *CAIDA* alerta para o fato de que nenhuma análise sobre tráfego de rede pode ser realizada com o uso de apenas um parâmetro, pois, é importante uma avaliação de um conjunto de variáveis e dos seus efeitos.

O grupo de trabalho *IPPM (IP Performance Metrics)* do *IETF* (*Internet Task Force*) tem como objetivo desenvolver um conjunto de métricas padrão, que possam ser usadas para medir o desempenho e a qualidade dos serviços de dados.

Na *RFC 2722 (Request for Comments)* está definida uma arquitetura para mensuração de pacotes, onde se procura métricas comuns para mensurar fluxos de tráfego de rede com objetivo de disponibilizar :

- Informações sobre o comportamento da rede;
- Informações sobre dimensionamento e expansão da rede;
- Informações para quantificar desempenho da rede.

Existem basicamente dois tipos de medições:

1) passivas – são coletadas informações sobre todos os pacotes que trafegam na rede sem provocar nenhuma interferência no tráfego e;

2) ativas – são gerados pacotes de teste e monitorado seu desempenho através da rede.

As medições passivas são caracterizadas por não interferir com o tráfego da rede. Para esse tipo de medição utiliza-se um dispositivo que observa todo o tráfego gerado e armazena as informações em arquivos.

Nas medições passivas, o monitoramento por *fluxo de tráfego* propõe a redução do volume de dados coletados através da utilização do conceito de fluxos de tráfego, em lugar da utilização dos pacotes de rede. Isto só é possível porque quando um fluxo é criado, esse se torna apenas uma entidade, caracterizada e individualizada através de parâmetros retirados, na maioria das vezes, dos cabeçalhos dos pacotes e, a partir daí, um contador tem seu valor incrementado no aparecimento de pacotes com as mesmas características, conforme ilustra a figura 3.11.

IP Fonte	**IP Destino**	**Protocolo**	**Porta Fonte**	**Porta Destino**	**Qtd. Pacotes**
10.176.8.30	192.168.10.	TCP	127	80	878
10.176.8.7	10.176.125.4	TCP	1722	110	89

Figura 3.11: Exemplo de Monitoramento por Fluxo de Tráfego**.**

Enquanto as medições passivas são mais adequadas para monitoramento do tipo de tráfego, as medições ativas tornam-se necessárias no monitoramento de outras métricas como, por exemplo, no atraso e perda de pacotes. Estas medições realizam o monitoramento influenciando no tráfego da rede através do envio de pacotes de testes. A ferramenta mais utilizada para estas medições é o comando *ping* que, através das

estatísticas do *RTT (round-trip-time)* dos pacotes, proporciona diversas conclusões sobre algumas métricas, como por exemplo, a latência e perda de pacotes.

Os esforços são direcionados para a caracterização do tráfego com objetivo de determinar:

- Quanto do tráfego está sendo destinado para cada serviço;
- A origem e destino de um tráfego;
- Qual o tempo médio de utilização da banda de cada serviço.

Para determinar se um recurso de rede está sendo muito ou pouco utilizado, ou se está em sua capacidade e desempenho máximos, precisa-se, primeiramente, saber o que é usual em um ambiente. Cada ambiente é único, com diferentes fatores afetando o modo como os recursos são utilizados. Coletar e salvar dados em períodos considerados de acesso normal ajuda a conhecer as demandas do ambiente. Para alcançar este objetivo deve-se utilizar sempre uma mesma metodologia, que através de coletas continuadas, constituirá um banco de dados das atividades de medição.

Na etapa de construção do modelo difuso sentiu-se necessidade de utilizar-se uma técnica de reconhecimento de padrões em arquivos de dados, justificando, dessa maneira, a apresentação do próximo assunto.

3.4 Clusterização e Clusterização C-Means

Segundo KAGEYAMA e LEONE (1999), o objetivo dos métodos de classificação é dividir em subconjuntos (classes), os mais semelhantes possíveis, um conjunto de elementos (indicadores) a partir de distâncias dois a dois.

Segundo SIMÕES (2003), os métodos de *clusterização* (*clustering*) podem ser caracterizados como qualquer procedimento *estatístico* que, utilizando um conjunto finito e multi-dimensional de informações, classifica seus elementos em grupos restritos e homogêneos internamente, permitindo gerar estruturas agregadas significativas .

Os métodos de classificação podem ser descritos como não-hierárquicos, quando produzem uma partição em número fixo de classes, e métodos hierárquicos, quando produzem seqüências de partições em classes cada vez mais vastas (BAROUCHE & SAPORTA, 1982).

Na classificação não-hierárquica agrupam-se n indivíduos em k classes, de maneira que indivíduos de uma mesma classe sejam os mais semelhantes possíveis e que as classes estejam bem esparsas.

As classificações hierárquicas podem ser aglomerativas ou divisivas. Segundo MIYAMOTO (1990), os métodos aglomerativos são mais utilizados em classificações hierárquicas aplicadas às ciências sociais, iniciando-se com cada elemento do conjunto referencial de dados (universo) formando seu próprio *cluster*. Sucessivamente, o número de classes irá decrescendo em uma unidade, reunindo-se as duas classes mais homogêneas, ou seja, as que possuam maior similaridade. Tal processo continua até que todos os elementos façam parte de um só agrupamento.

Os métodos divisivos iniciam de forma inversa, ou seja, consideram todo o conjunto de elementos como um só agrupamento e, utilizando-se de alguma métrica de dissimilaridade, vão procedendo a separações deste grupo inicial até que cada indivíduo seja uma classe em si. (KAGEYAMA & LEONE, 1999).

Para o objetivo do presente trabalho e considerando a grande quantidade de dados a serem analisados, optou-se pela utilização de um método não-hierárquico, considerando que esses métodos são mais indicados em situações que envolvam grande amostras.

Para JONHSON e WICHERN (1992), a definição da métrica de dissimilaridade acompanha as peculiaridades do conjunto de dados, as características estatísticas das variáveis e os objetivos da classificação, podendo ser baseada em um ou múltiplos atributos dos indivíduos. Os métodos usuais de classificação, presentes nos softwares estatísticos atuais, permitem a utilização de várias métricas, sendo a distância euclidiana a mais utilizada.

Para HAN e KAMBER (2001), as métricas de similaridade ou dissimilaridade podem ser obtidas através da apuração das distâncias entre pares de objetos. Essa mensuração pode ser obtida através da aplicação de distância euclidiana, distância Manhattan ou mesmo distância Minkowski. Os algoritmos *k-means*, e por extensão os algoritmos fuzzy c-means, são implementados, utilizando, na sua maioria, a distância euclidiana na apuração das similaridades.

A classificação de indivíduos em grupos homogêneos, nos quais os valores médios de cada classe representariam os indivíduos nela alocados, com a variabilidade intraclasse mínima e variabilidade interclasse máxima, permite criar taxonomias, tipologias, reduzindo a quantidade de dimensões a serem analisadas e possibilitando um entendimento mais direto das características inerentes das informações (JONHSON & WICHERN,1992).

Para SIMÕES (2003), a teoria dos conjuntos traz consigo uma noção dicotômica fundamental: pertencer ou não pertencer. A definição de um conjunto clássico implica na tomada de uma decisão binária quanto à pertinência de determinado indivíduo (objeto, elemento) numa dada classe (grupo, categoria): aceitar (1) ou rejeitar (0) tal proposição.

Para KAUFMAN e ROUSSEEUW (1990), se o conjunto de informações, seja pelas peculiaridades do objeto a que representam, seja pela ambigüidade da própria estrutura de dados, possui uma fonte de imprecisão, que não a aleatoriedade derivada de processos estocásticos, e sim derivada da ausência de fronteiras abruptamente definidas entre as classes, deve-se voltar a atenção para a utilização da Teoria dos Conjuntos Nebulosos *(Theory of Fuzzy Sets).*

De acordo com ZADEH (1965), um subconjunto difuso de um conjunto X qualquer é definido como uma função $\mu : X [0,1]$; para cada $x \in X$ o valor de μ (X) é o grau de pertinência de x a um subconjunto. Assim, se em vez de assumir valores no intervalo discreto $\{0,1\}$ a função de pertinência assumir valores no intervalo contínuo $[0,1]$ então o conjunto A denomina-se conjunto difuso, com cada indivíduo podendo vir a pertencer parcialmente a múltiplos conjuntos. O valor de μ (X) é usualmente utilizado para

representar o grau, ou a extensão na qual X se associa com a descrição semântica de μ, sendo que μ (X) não pode ser interpretado como a probabilidade de X pertencer à classe μ, e sim o quanto pertence.

O processo de *clusterização* difusa tem como objetivo identificar, em conjunto de dados coletados, agrupamentos que demonstrem regiões com contornos delineados. Esses agrupamentos são chamados de *clusters*. Faz parte do processo de *clusterização* calcular o vetor V de centros de cada *cluster* e uma matriz de partição difusa U, conforme ilustra a figura 3.12.

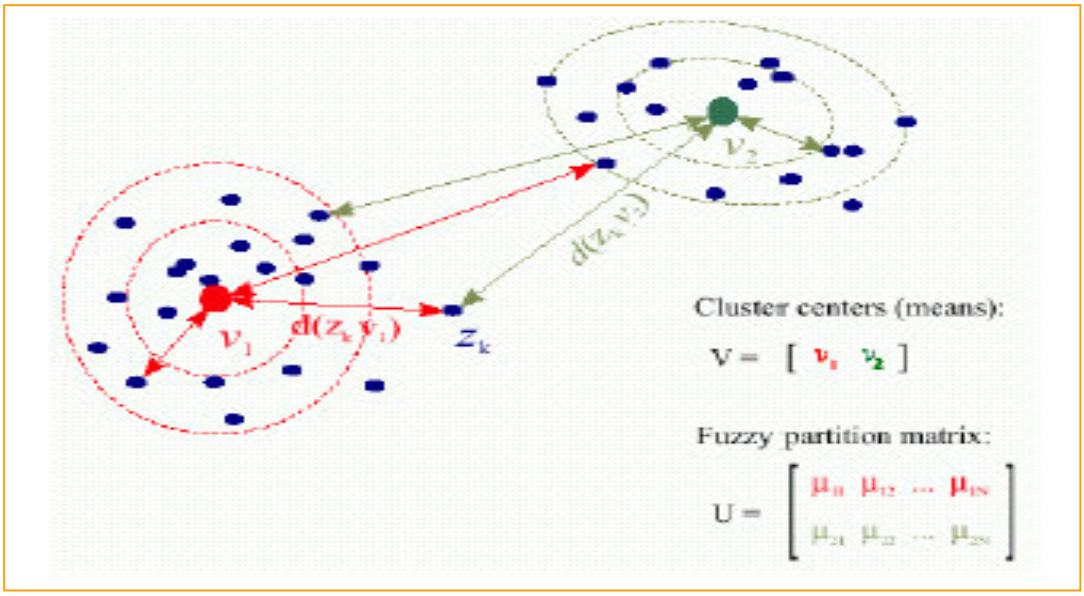


Figura 3.12 Processo de *Clusterização* Difusa (SIMÕES,2003)

A matriz de partição difusa U apresenta os graus de pertinência de todos os pontos a todos os *clusters*. Para todo o novo ponto coletado Z_k são calculadas as distâncias desse ponto a cada *cluster*, definindo assim uma determinada coluna U.

Um dos algoritmos de *clusterização* mais utilizados é o *Fuzzy C-Means* (FCM). O objetivo do FCM, para SIMÕES (2003), é minimizar uma função do tipo:

$$J = \sum_{i=1}^{c} \sum_{j=1}^{N} \mu_{ij}^{m} d^2(z_j, v_i)$$

onde *m* é um fator que controla a difusividade dos *clusters*. *m>=1*. Quanto maior seu valor, mais difusas ficam as regiões de transição entre os *clusters*. Um valor típico é *m=2*. $d^2(z_j,v_i)=(z_j-v_i)^T(z_j-v_i)$ é a norma euclidiana que representa a distância entre o ponto z_j e o centro v_i i-ésimo *cluster*. v_i é a variável de livre escolha no algoritmo.

O FCM não leva em conta a direção do ponto em relação a um determinado *cluster*, por isso, segundo SIMÕES (2003), os agrupamentos são esféricos. No caso do algoritmo de Gustafson-Kessel, a pertinência de um ponto a um *cluster* tem sua variação dependente tanto da distância em si, como da direção.

Para o FCM, as seguintes regras devem ser obedecidas:

$$0 \leq \mu_{ij} \leq 1 \quad , \quad i = 1 \ldots c \quad , \quad j = 1 \ldots N$$ Faixa do grau de pertinência.

$$0 < \sum \mu_{ij} < N \quad , \quad i = 1 \ldots c$$ Nenhum *cluster* vazio.

Para BERNI (2004), a seqüência do algoritmo FCM inicia com a geração de uma matriz de partição nebulosa U, de forma randômica, respeitando as restrições acima mencionadas.

Nos cálculos dos centros dos *clusters*, segundo BERNI (2004), utiliza-se:

$$v_i = \frac{\sum_{k=1}^{N} \mu_{i,k}^m z_k}{\sum_{k=1}^{N} \mu_{i,k}^m}$$

Nos cálculos das distâncias, segundo BERNI (2004), utiliza-se:

$$d_{ik}^2(z_k, v_i) = (z_k - v_i)^T (z_k - v_i)$$

Na determinação dos graus de pertinência que definem a nova matriz de partição nebulosa, segundo BERNI (2004), utiliza-se:

$$\mu_{ik} = \frac{1}{\sum_{j=1}^{c} (d_{ik}/d_{jk})^{2/(m-1)}}$$

Por fim, calcula-se o erro entre a matriz U atual e a anterior. Caso $\|\Delta U\| < \epsilon$, finalizar o algoritmo. Caso contrário, voltar a executar os cálculos dos centros de *clusters* e repetir toda a seqüência.

4. TRÁFEGO DE REDE SOB O PARADIGMA DA LÓGICA DIFUSA

4.1 Caracterização do Problema

Os computadores são interconectados como conseqüência dos avanços das tecnologias e dos benefícios advindos pelo uso das redes de computadores. Simultaneamente, verifica-se a diminuição dos custos dos equipamentos, permitindo agregar à rede cada vez mais dispositivos de diferentes fornecedores, tornando essas redes cada vez maiores e mais complexas.

As redes de computadores são extremamente importantes para muitas empresas, porque, normalmente, vinculado a sua utilização, está a competitividade e a sobrevivência de muitas corporações. As empresas têm se tornado altamente dependente das tecnologias de redes, sentindo, imediatamente, o impacto quando os seus recursos não estão disponíveis. Torna-se evidente a necessidade de estabelecer monitoramento e controle sobre o comportamento do tráfego de rede, como forma de garantir que os problemas sejam identificados e solucionados rapidamente.

Para atender a esta necessidade foram desenvolvidos os protocolos de gerenciamento. A preocupação de um protocolo de gerenciamento é realizar tarefas que permitam a obtenção de dados sobre desempenho e tráfego da rede, o diagnóstico de problemas de comunicação, e a reconfiguração da rede para atender as mudanças de necessidades dos usuários e do ambiente.

Uma das principais funções de um sistema pró-ativo é a notificação sobre a ocorrência de situações que possam levar à degradação da rede e sugerir ações pró-ativas. Para que seja feita uma boa notificação de previsões, o processo de diagnóstico é adaptado à monitoração constante da rede.

Segundo BOWERMAN e OÇONNEL (1987), a escolha da técnica de previsão deve levar em consideração os seguintes fatores:

- **intervalo de tempo**: identifica os períodos em que se deseja monitorar, pois as previsões são feitas para intervalos de tempo;
- **padrão dos dados**: identifica padrões ou tendências existentes nos dados;
- **custo de previsão**: abrange o custo de desenvolvimento do modelo, o custo do armazenamento das informações necessárias à previsão e o custo do cálculo da previsão;
- **erro amostral**: grau de confiança da previsão;
- **disponibilidade de dados;**
- **facilidade de operação e compreensão**.

Segundo FRANCESCHI et al. (1997), o processo de diagnóstico pode ser realizado através da utilização de um sistema especialista, onde se tem o conhecimento representado através de fatos e regras. Os fatos são obtidos através dos dados disponíveis, e as regras determinadas por um especialista.

4.2 Motivação da Pesquisa

As ferramentas de monitoramento de tráfego de rede, proprietárias ou não, distribuídas comercialmente ou de acesso livre, apresentam as mais variadas características. Apesar das facilidades de monitoramento de variáveis e visualizações gráficas de tráfego, o processo de diagnóstico, verificado empiricamente nas ferramentas ilustradas na tabela 4.1, ainda é uma tarefa difícil. A cada momento o administrador de rede, que tem suas decisões apoiadas basicamente em dados, depara-se com situações que exigem tratamento de incerteza. As ferramentas ainda não atingiram um grau de automação

que lhes possibilitem identificar problemas e sugerir ações corretivas, deixando para o administrador o encargo da interpretação de gráficos e de valores de variáveis para tomadas de decisões.

Produto	Fornecedor
Scotty	Utwente, TU Braunschweig
Ntop	Luca Deri
NetTraMet	Nevil Brownlee
The Multi Router Traffic Grapher (MRTG)	T. Oetiker and D. Rand
Ethereal	Open Source
Network Management Software	3Com
Net-snmp	Univ of California, Davis
PATROL® Enterprise Manager	BMC Software
Hp OpenView	Hewlett Packard Company

Tabela 4.1: Ferramentas de Monitoramento de Rede Analisadas Empiricamente (SIMPLEWEB, 2004)

A presente pesquisa tem como meta aplicar e avaliar o uso das técnicas da lógica difusa na procura de relações, entre os dados presentes no tráfego de rede, que possibilitem apurar diferentes estados de comportamento, reportando esses eventos à administração da rede, no momento em que ocorram.

Aliado a aplicação das técnicas da lógica difusa, procurou-se utilizar os conceitos de monitoração por fluxos de tráfego, descrito no tópico sobre tráfego de redes, como forma de diminuir a massa de dados a ser coletada para utilização na geração de contadores e somadores utilizados como variáveis de entrada do protótipo.

A idéia básica é, justamente, conseguir um perfil de comportamento de determinado segmento de rede e, a partir dessa constatação, gerar informações sobre possíveis desvios. Entende-se como segmento de rede, uma rede local com uma infra-estrutura que justifique o monitoramento.

A complexidade das redes e a heterogenidade de hardware e software dificultam qualquer tipo avaliação. Não é possível estabelecer limites ou escopos fixos de avaliação de desempenho, se, a todo momento, um novo software ou hardware está compondo ou utilizando recursos de rede, ficando, assim, caracterizado a incerteza e a imprecisão no gerenciamento do tráfego.

Por outro lado, quando se trabalha com mensuração de tráfego de rede, deve-se ter em mente que qualquer estratégia não poderá interferir ou contribuir para modificar o comportamento natural do tráfego. As medições passivas, já descritas nesse trabalho, são caracterizadas por não interferir com o tráfego da rede, utilizando um dispositivo que observa todo o tráfego gerado e armazena informações em arquivos. Nas medições passivas, o monitoramento por fluxo de tráfego propõe a redução do volume de dados coletados através da utilização do conceito de fluxos.

O tempo de monitoramento é, também, um fator importante quando se trabalha com medições passivas, pois existirá um custo de processamento e armazenamento, que exigirá uma capacidade computacional adequada para um pós-processamento e geração de informação em tempo hábil à tomada de decisão. O tempo de monitoramento, adequado para protótipo, foi estimado através de diversos ensaios experimentais.

4.3 Técnicas Utilizadas

Para alcançar o objetivo de caracterizar o comportamento de um segmento rede, utiliza-se :

Técnica	Motivo
Lógica difusa	Possibilitar a modelagem de um raciocínio, por inferência lógica, nas situações de incertezas encontradas na subjetividade de se modelar o comportamento de uma rede;
Medições passivas de tráfego	Evitar ruídos no tráfego de rede durante o monitoramento;
Cabeçalhos dos protocolos de rede e o conceito de fluxo de tráfego.	É a fonte de informação para geração dos contadores/somadores que compõem as variáveis de entrada do modelo difuso;
Consultoria de profissional de administração de redes	Necessidade de apoio na construção da base de regras difusa e na avaliação dos resultados;
Utilização do método *fuzzy c-means*	Possibilitar o ajuste de parâmetros das funções de pertinência;
Conceitos de administração e modelagem de dados	Possibilitar a modelagem de uma base de dados para armazenamento de informações de tráfego de rede, servidores e de serviços de rede.

Tabela 4.2 : Técnicas Utilizadas no Modelo

4.4 Modelo Experimental Proposto

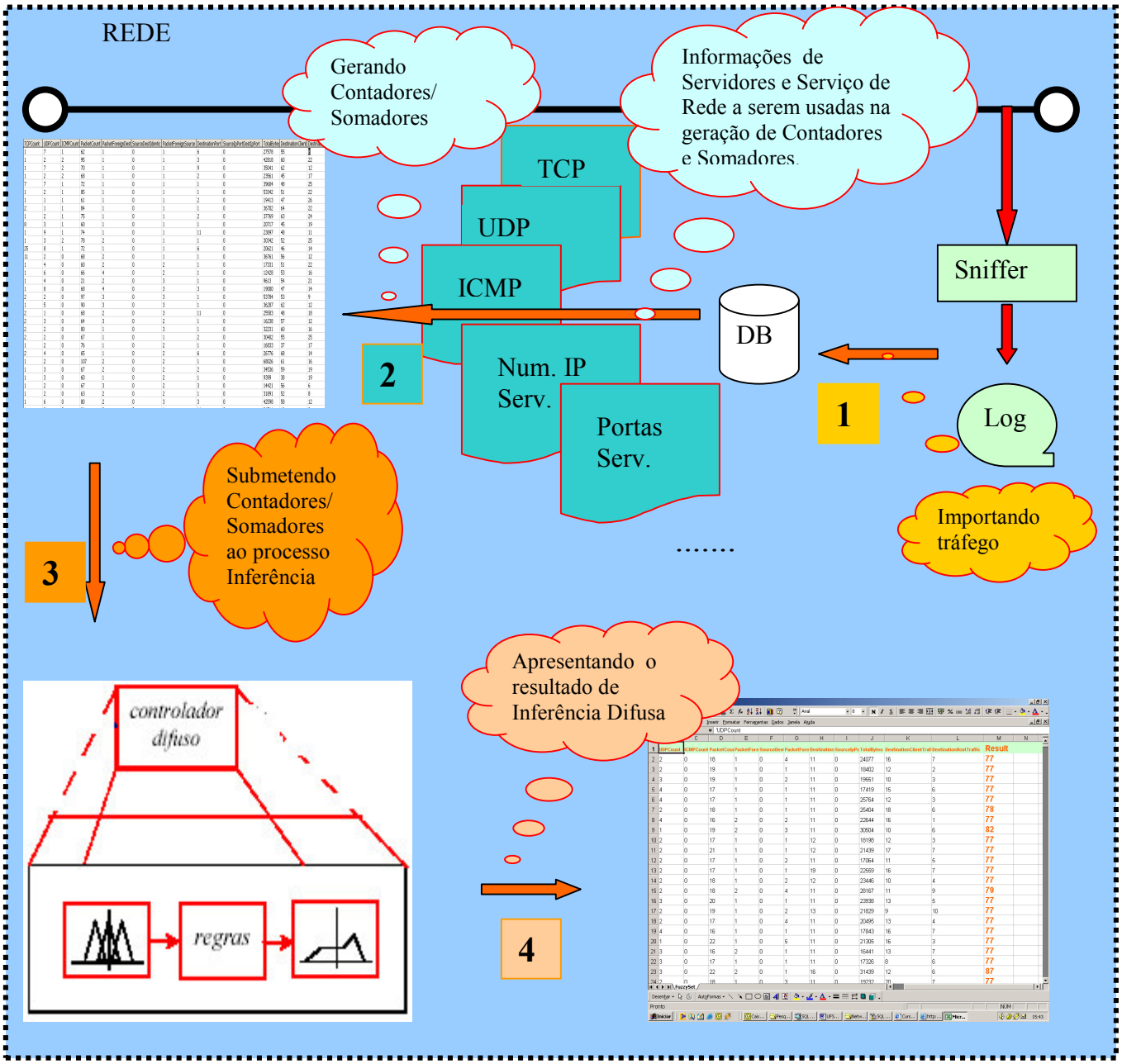


Figura 4.1: Esquema do Modelo Difuso Proposto

A ilustração da figura 4.1 retrata o funcionamento do modelo difuso proposto para monitorar o comportamento de um segmento de rede. Conforme é possível constatar, se faz uso de um processo do tipo *sniffer* que observa todo o tráfego de rede e armazena as

informações dos cabeçalhos dos protocolos, que compõem o tráfego, em *logs,* durante um horário pré-estabelecido.

Em determinados intervalos de tempo é executado um *package (pacote),* detalhado posteriormente nesse capítulo*,* que seqüencializa a execução dos procedimentos identificados na figura 4.1, da seguinte maneira:

1. O primeiro procedimento da série executa a importação de todo tráfego de rede armazenado no *log* do *sniffer* e atualiza tabelas de uma base de dados;

2. De posse das informações do *log*, e em seqüência, o processo de geração de contadores e somadores entra em execução e gera, em forma tabular, as variáveis de entrada para o controlador difuso. Para geração dos contadores e somadores são selecionados os pacotes dos protocolos *TCP(Trasnmission Control Protocol), UDP (User Datagram Protocol) e ICMP (Internet Control Message Protocol)*, identificados os servidores existentes, as portas disponíveis por servidor, máscara da rede e números de IP's origem e destino;

3. O controlador difuso, baseado no modelo MAMDANI (1975), executa o processo de inferência que terá como resultado a avaliação do comportamento de um segmento de rede monitorado. Uma notificação é gerada, para a administração de rede, nos casos de mudança desse comportamento;

4. A notificação, caracterizada na figura 4.1, é realizada através de um *e-mail* endereçado à administração da rede com uma planilha anexa contendo os valores das variáveis de entrada e o resultado do processo difuso.

Não é objetivo do presente trabalho estruturar o modelo difuso para que gere sugestões de *ações corretivas*, mas sim, conforme citado anteriormente, avaliar a

possibilidade de uso das técnicas difusas na busca de relações entre os dados presentes no tráfego de rede, que possam caracterizar o comportamento das redes. As ações corretivas surgirão em conseqüência da evolução do uso das técnicas difusas em ferramentas desenvolvidas com objetivo de monitoramento e gestão de redes.

A utilização da lógica difusa e dos conceitos de medição por fluxo de tráfego objetivam, nesse trabalho, direcionar os procedimentos a serem adotados pela administração da rede, inclusive, no que se refere à utilização de softwares específicos para atender situações específicas.

4.5 Etapas de Construção do Modelo Difuso

O aspecto fundamental de uma modelagem difusa é o sistema de inferência que está baseado em seu conjunto de regras. No processo de construção dos modelos difusos, o conhecimento especializado, os dados disponíveis, ou ambos são básicos para o sucesso do modelo.

4.5.1 Definição das Variáveis do Modelo

Não existe método para identificação das variáveis relevantes, sendo, geralmente, utilizada uma análise dos dados disponíveis (AGUIAR & JUNIOR,1999).

O objetivo dessa etapa é determinar as variáveis de entradas (variáveis independentes) e as variáveis de saída (variáveis dependentes) que descrevam o comportamento do sistema.

As variáveis de entrada utilizadas no modelo proposto são contadores e somadores gerados a partir dos cabeçalhos dos protocolos de redes. Num primeiro momento procurou-se identificar quais protocolos tem maior presença e importância na arquitetura *TCP/IP*.

Para DICKERSON e DICKERSON (2001), os três principais protocolos da internet são os protocolos *TCP(Transmission Control Protocol)*, *UDP (User Datagram Protocol)* e *ICMP (Internet Control Message Protocol)* .

Com base na pesquisa realizada, foram escolhidos os protocolos *TCP* e *UDP* da camada de transporte e o protocolo *ICMP* da camada de rede, todos pertencentes à arquitetura de rede *TCP/IP*. Os dois primeiros protocolos mencionados são responsáveis pela geração da maior parte do fluxo de tráfego de rede, devido ao transporte das informações da camada de aplicações. O protocolo *ICMP*, conforme menciona DANTAS (2002), tem por objetivo prover mensagens de controle na comunicação entre nós em um ambiente.

Os contadores e somadores, utilizados como variáveis de entrada do modelo difuso, são gerados a partir dos campos que compõem o cabeçalho dos protocolos. Os campos escolhidos estão relacionados na tabela 4.3.

Endereço IP Origem
Endereço IP Destino
Porta Origem
Porta Destino
Tamanho do Pacote (cabeçalho + dados)

Tabela 4.3 : Campos dos Cabeçalhos dos Protocolos Coletados

Após a coleta dos *logs,* gerados pelo *sniffer,* e sua transferência para as tabelas da base de dados, inicia-se o processo de geração dos contadores e somadores, a partir das informações dos campos descritos na tabela 4.3. Esses contadores e somadores se transformarão nas variáveis de entrada do modelo difuso. A próxima tabela sintetiza a relação de contadores e somadores escolhidos e o motivo de sua escolha.

Variáveis	Motivo
Contador de tráfego *TCP,* por unidade de tempo	O TCP é um protocolo desenvolvido para oferecer um serviço confiável entre aplicações. Para efetuar suas tarefas com sucesso, o protocolo identifica os pacotes recebidos fazendo uma correlação de cada pacote com suas respectivas conexões (DANTAS, 2002). Por esse motivo, o contador é gerado considerando a conexão, ou seja, considerando todos os pacotes com mesma origem, destino e portas, por unidade de tempo. O objetivo desse contador é identificar

	possíveis anormalidades para grandes números de conexões, o que poderia identificar uma sobrecarga na rede, ou, no caso de um pequeno número de conexões, significar um problema com a rede. Várias aplicações, entre as quais o *SMTP – Simple Mail Transfer Protocol, FTP – File Transfer Protocol e TELNET*, fazem uso desse protocolo.
Contador de tráfego *UDP*, por unidade de tempo	Esse protocolo é conhecido como otimista ou leve, justamente, por efetuar o envio de seus pacotes acreditando que eles irão chegar sem problemas e em seqüência no destino (DANTAS, 2002). O objetivo desse contador é identificar possíveis anormalidades para grandes números de pacotes, o que poderia identificar uma sobrecarga na rede, ou, no caso de um pequeno número de conexões, significar um problema com a rede. Várias aplicações, entre as quais o *SNMP – Simple Network Management Protocol, TFTP – Trivial File Transfer Protocol, RPC – Remote Procedure Call*, fazem uso desse protocolo.
Contador de tráfego *ICMP,* por unidade de tempo	O protocolo ICMP tem por objetivo prover mensagens de controle na comunicação (DANTAS, 2002). Um elevado valor para esse contador, possivelmente, indicará problemas físicos de rede.
Contador de tráfego por unidade de tempo	Essa contagem abrange todo o tráfego coletado em determinado período. Tem como objetivo indicar a vazão da rede (*throughput*).
Somador para o total de bytes do tráfego, por unidade de tempo.	Essa soma abrange todo o tráfego coletado em determinado período. Tem como objetivo indicar a vazão da rede (*throughput*), em termos de quantidade de bytes.
Contador de tráfego com destinos externos a rede monitorada, por unidade de tempo.	Esse contador tem como objetivo verificar o comportamento dos usuários em torno do uso da internet.
Contador de tráfego com origens externas a rede monitorada, por unidade de tempo	Esse contador procura verificar o acesso externo ao segmento de rede monitorado.
Contador de tráfego com mesmo endereço origem e destino.	Esse contador tem como objetivo verificar a existência de conexões TCP anormais no segmento monitorado.
Contador de tráfego por porta destino de servidores	Em geral as portas destino representam serviços de rede, como por exemplo, a porta 23 que é utilizada para a aplicação *telnet* e a porta 80 para *http*. Em situações normais, os usuários tendem a utilizar os mesmos serviços, demonstrando um padrão de comportamento (DICKERSON & DICKERSON, 2001). Isso significa que esse contador deverá ter poucas variações após análise do perfil de comportamento dos usuários do segmento de rede monitorado.
Contador para tráfego destinado às estações de trabalho, por unidade de tempo.	O objetivo desse contador é avaliar o tráfego restrito ao segmento monitorado e que não faz uso dos serviços de servidores disponibilizados na rede.
Contador para tráfego destinado a servidores de trabalho, por unidade de tempo.	O objetivo desse contador é avaliar o tráfego restrito ao segmento monitorado e que faz uso dos serviços de servidores disponibilizados na rede.

Contador de tráfego com mesmo endereço IP origem, porta origem, endereço IP destino e porta destino, por unidade de tempo	Existe uma tendência dos mesmos usuários utilizarem sempre os mesmos serviços dos mesmos servidores, sendo assim, esse contador poderá informar possíveis problemas de desempenho de rede quando os usuários mudarem sua forma de trabalho ou exigirem mais de determinado serviço de rede.

Tabela 4.4 : Variáveis de Entrada do Modelo Difuso Proposto

Estes contadores e somadores foram estruturados com o objetivo de expressar um perfil de rede, e a partir dessa constatação, ser possível, através de um mecanismo inteligente, avaliar e informar mudanças de comportamento. É importante mencionar que estes contadores e somadores foram especificados juntamente com a administração da rede e, a qualquer momento, podem ser atualizados, modificados, ou mesmo extintos. A utilização prática do protótipo poderá ajudar nessa customização.

A figura 4.2 demonstra os contadores e somadores já devidamente contabilizados e formatados para execução do processo de inferência difusa. A unidade de tempo utilizada para montagem dos contadores e somadores é de um segundo.

TCPCount	UDPCount	ICMPCount	PacketCount	PacketForeignDest	SourceDestIdentic	PacketForeignSource	DestinationPort	SourceIpPortDestIpPort	TotalBytes	DestinationClient	Destinatior
1	7	1	62	1	0	1	6	0	27578	55	3
1	2	2	95	1	0	1	3	0	42818	60	22
1	7	2	70	1	0	1	9	0	35041	62	12
1	2	2	68	1	0	1	2	0	23561	45	17
7	7	1	72	1	0	1	1	0	39604	48	25
1	2	1	85	1	0	1	1	0	53342	51	22
1	1	1	61	1	0	1	2	0	19413	47	26
2	1	1	84	1	0	1	1	0	36702	64	22
1	2	1	75	1	0	1	2	0	37769	63	24
8	3	1	60	1	0	1	1	0	20717	45	19
1	9	1	74	1	0	1	11	0	23897	48	11
1	3	2	78	2	0	1	1	0	30342	52	25
25	8	1	72	1	0	1	6	0	20621	46	14
11	2	0	68	2	0	1	1	0	36761	56	12
1	4	0	60	2	0	2	1	0	17331	51	22
1	6	0	66	4	0	2	1	0	12428	53	16
1	4	0	21	2	0	3	1	0	9613	54	21
1	8	0	68	4	0	3	3	0	19000	47	14
2	2	0	97	3	0	3	1	0	53784	53	9
1	5	0	90	3	0	3	1	0	36287	62	12
2	1	0	68	2	0	3	11	0	25583	48	18
2	3	0	64	3	0	2	1	0	16238	57	12
2	2	0	80	1	0	3	1	0	32231	60	16
2	2	0	67	1	0	1	2	0	30402	55	25
1	2	0	76	1	0	2	1	0	16833	37	17
2	4	0	65	1	0	2	6	0	26776	68	14
1	2	0	107	2	0	2	1	0	68026	61	16
1	3	0	67	2	0	2	2	0	34536	59	19
1	3	0	60	1	0	2	1	0	9399	38	19
1	2	0	67	3	0	2	3	0	14421	56	6
1	2	0	63	2	0	2	1	0	31891	52	8
1	6	0	80	2	0	3	3	0	42598	58	12

Figura 4.2: Variáveis de Entrada (Contadores e Somadores) do Modelo Difuso (Protótipo Difuso)

4.5.2 Particionamento do Universo de Discurso das Variáveis

O particionamento do universo de discurso das variáveis tem como objetivo a representação das variáveis numéricas como variáveis lingüísticas.

Para uma modelagem inicial optou-se em executar, para cada contador e somador, uma função matemática que informasse os valores mínimos e máximos auferido, através de uma amostra coletada durante uma semana. Estes valores foram utilizados como universo de discurso para cada variável do modelo.

4.5.3 Definição das Funções de Pertinência e Termos Lingüísticos

A atribuição de funções de pertinência, para cada partição de domínio das variáveis, é, ainda, uma questão em estudo na modelagem de sistemas difusos. Por esse motivo as formas mais simples são as mais utilizadas.

Para WEBER e KLEIN (2003), a maioria das aplicações difusas utiliza funções de pertinência padrão, ou seja, as formas triangulares e trapezoidais, porque são simples, porém suficientes para o uso em modelos difusos, e são eficientes em termos computacionais na maioria das plataformas. Essas funções de pertinência, segundo AGUIAR e JUNIOR (1999), aparecem normalmente em casos onde se deseja exprimir pertinência crescente à esquerda e decrescente à direita.

A quantidade de funções em um universo de discurso e seu formato são escolhidos com base na experiência de um especialista ou na natureza do processo. Algumas sugestões são mencionadas em SHAW e SIMÕES (1999) :

- Um número prático de funções de pertinência é algo entre 2 e 7.
- Os formatos mais freqüentes são a triangulares e trapezoidais, sendo que em casos onde um desempenho suave é de importância crítica, as funções gaussiana e sigmoidais podem ser utilizadas. É importante notar que as funções de pertinência não precisam ser simétricas ou igualmente espaçadas, e que cada variável pode ter um conjunto de funções de pertinência diferente, com diversos formatos e distribuições.
- Um fator que afeta a precisão é o grau de superposição entre as funções de pertinência, sendo que um mínimo de 25% e um máximo de 75% foram

> determinados experimentalmente como adequados, mas 50% têm uma aceitabilidade razoável nos primeiros testes do modelo.

De posse das considerações bibliográficas, optou-se pelas formas triangulares para referenciar os termos lingüísticos de normalidade (*normal*) e aceitabilidade (*acceptable)*, e funções de formato Z e S para descrever, respectivamente, os termos com valores baixos (*lower*) e anormais (*anormal)*, representando, assim, o mapeamento dos números reais dos contadores e somadores em números difusos.

Os termos lingüísticos *lower* (pequeno), *normal* (normal), *acceptable* (aceitável) e *anormal* (anormal), foram, então, os escolhidos para descrever o comportamento de cada variável.

Esses termos foram igualmente espaçados em cada universo de discurso, de cada variável de entrada, nos primeiros testes do modelo, mas esta sugestão, mencionada em bibliografias consultas, mostrou-se ineficaz na especificação do presente modelo. Posteriormente, então, utilizou-se o algoritmo *Fuzzy C-Means* e a ferramenta de *clusterização*, ambos disponíveis no software MATLAB®, para o ajuste das funções de pertinência de cada termo lingüístico.

Para execução dos ajustes considerou-se a existência de quatro *clusters* e um *log* de tráfego de rede gerado e armazenado durante *três meses* de coleta.

O modelo MAMDANI (1975) foi escolhido para o desenvolvimento do protótipo, pois, segundo SHAW e SIMÕES (1999), matematicamente não há diferenças entre a abordagem MAMDANI (1975) e TAKAGI e SUGENO (1985). O diferenciador das duas abordagens está caracterizado nas relações difusas existentes nos conseqüentes das regras.

A saída de cada regra é representada, no protótipo, por funções de pertinência triangulares com três termos lingüísticos, assim caracterizados: *normal*, *acceptable* e *anormal*.

O ajuste dos parâmetros das funções de pertinência é uma tarefa que consome tempo por envolver um processo de tentativa e erro, em que o conjunto de regras é inferido, com base nos valores informados para as variáveis de entrada, e os resultados dessa inferência avaliados através de comparações com os valores obtidos para as variáveis de saída.

Em geral, há muitos graus de liberdade na modelagem difusa, pois uma mesma alteração na resposta do modelo pode ser obtida com a alteração dos parâmetros das funções de pertinência dos antecedentes ou das conseqüentes das regras envolvidas.

4.5.4 Construção das Regras

Nessa etapa as proposições antecedentes e conseqüentes de cada regra são descritas considerando as possíveis interações entre as variáveis selecionadas.

Um sistema de regras estará completo quando puder responder satisfatoriamente a todas as ocorrências possíveis envolvendo o fenômeno modelado.

O processo de inferência de um sistema que utiliza lógica difusa está caracterizado na construção de sua base de regras, sendo que o processo nada mais faz do que avaliar os níveis de compatibilidade das entradas com os antecedentes das várias regras, ativando os conseqüentes com intensidade proporcional aos mesmos.

O resultado de todo o processo é um conjunto difuso que será convertido em escalar (valor condensado ou defuzzificação) para o fornecimento da saída do sistema (AGUIAR & JUNIOR, 1999).

Na agregação dos vários conjuntos difusos de entrada, dentro de uma mesma regra, as *normas-t min* e *normas-t produto* são mais comuns, enquanto que na combinação das saídas difusas de cada regra, a *co-normas max* prevalece. A *normas-t min* e *produto* representam a intersecção difusa e implicam em um conectivo *E*. O conjunto difuso conseqüente da inferência das regras será uma versão truncada do menor valor difuso

encontrado entre os antecedentes de cada regra, quando utilizada as *normas-t min,* e uma versão escalonada (multiplicada) no caso de utilização das *normas-t produto*.

Na composição, os conjuntos difusos conseqüentes de cada regra são combinados usando o operador *co-normas max,* que corresponde ao conectivo *OU.* De acordo com as regras de união difusa*,* as *co-normas max* geram um contorno ou envelope, comum aos conjuntos difusos resultantes de cada regra. Para o presente trabalho optou-se por utilizar a *normas-t min* na agregação da regras e a *co-normas max* na composição.

Uma observação pertinente sobre esta etapa, apesar de não evidenciada no presente trabalho, é encontrada em SHAW e SIMÕES (1999), onde é descrito o fato da *co-normas max* tenderem a produzir uma distribuição uniforme na composição final das regras, à medida que o número de conjuntos difusos combinados aumenta, ou seja, quando o número de regras aumenta, a sensibilidade do sistema diminui. O operador *soma* apresenta-se como uma alternativa para modelos com uma grande quantidade de regras, pois a utilização desse operador, implicará em uma soma dos números difusos, representando cada um dos conjuntos conseqüentes de cada regra.

4.5.5 Defuzzificação

O processo de construção das regras finaliza com a chamada defuzzificação, onde o valor da variável lingüística de saída, inferida pelas regras difusas, será traduzida para um valor numérico. A defuzzificação é a transformação de uma saída do domínio difuso para o domínio discreto. São três os métodos de defuzzificação mais utilizados para alcançar este objetivo:

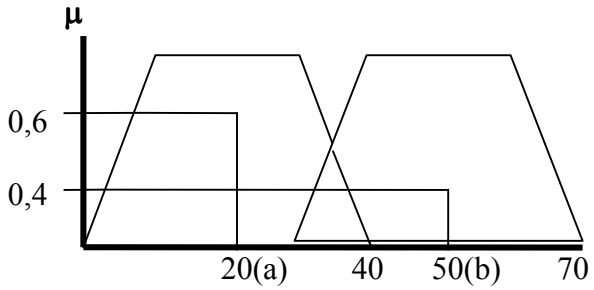
Método do Centro da Área: este método calcula o centróide da área composta pelas saídas difusas formadas pela união de todas as contribuições de regras. O centróide é um ponto que divide a área difusa de saída em duas partes iguais. Esse método apresenta alguns problemas. Um dos problemas ocorre quando as funções de pertinência não possuem sobreposição, acarretando, assim, a inexistência de um centro geométrico. Outro problema

ocorre quando mais de uma regra resulta em uma mesma saída difusa, havendo, nesse caso, uma sobreposição de áreas que não será devidamente contabilizada.

$$\textbf{centróide} = \sum_{i}^{n} u_i \, \mu_{out}(u_i) \; / \; \sum_{i}^{n} \mu_{out}(u_i) \text{ , onde}$$

$\mu_{out}(u_i)$ é a área de uma função de pertinência modificada pelo resultado da inferência difusa; e u_i a posição do centróide de cada função de pertinência individualmente (Adaptado de AGUIAR e JUNIOR (1999).

Para ilustrar o método acima descrito, demonstra-se abaixo um exemplo:



- No eixo x a centróide do ponto (a) é 20 e do ponto (b) é 50.
- O grau de pertinência do ponto (a) é 0,6 e do ponto (b) é 0,4.
- Área do 1º trapézio modificada = 0,6 (40+28) /2 = 20,4.
- Área do 2º trapézio modificada = 0,4 (40 +32) /2 = 14,4.
- Média ponderada = 20,4 (20) + 14,4 (50) / 20,4 + 14,4 = 32,4.

Método da Média dos Máximos: este método utiliza os máximos das funções de pertinência representados no universo de discurso da variável de saída. A saída discreta é obtida como uma média ponderada dos máximos.

Método do Centro do Máximo: esta abordagem utiliza-se da saída, cujo valor tenha maior grau de pertinência. Em casos onde a função de pertinência tenha mais de um máximo, esse método não tem aplicabilidade.

Optou-se pela utilização do método do centro da área por ser considerado um método contínuo, onde pequenas mudanças em uma variável de entrada não causarão mudanças abruptas nas variáveis de saída.

4.6 Implementação do Protótipo

4.6.1 Escopo da Implementação

O objetivo da implementação é testar a viabilidade e analisar o resultado da utilização da lógica difusa e das técnicas de mensuração de tráfego de rede na caracterização do comportamento de um segmento de rede.

Não existe a intenção de desenvolver uma ferramenta para gerenciamento de tráfego de rede, mas avaliar a possibilidade de integração dessas técnicas às ferramentas já existentes.

A construção do protótipo se baseou na plataforma *Microsoft* pelo fato de ser o ambiente computacional da empresa utilizada como laboratório para o projeto.

4.6.2 Ambiente de Desenvolvimento

As ferramentas utilizadas no desenvolvimento do protótipo foram escolhidas com base em dois fatores:

1. Apresentarem-se como ferramentas de Rápido Desenvolvimento de Aplicações (*Rapid Application Development – RAD*);

2. Estarem entre as ferramentas de desenvolvimento homologadas e padronizadas na empresa onde está sendo realizado todo trabalho de implementação e testes.

4.6.3 Plataforma Desenvolvimento

O desenvolvimento do projeto teve como suporte os seguintes equipamentos e configurações:

Marca/Modelo	Número Processadores	Tipo	Clock	Mem. RAM	Disco	Sistema Operacional
Itautec/InfoWay	01	Pentium III	250 Mhz	390 KB	28 GB	Windows 2000 Professional
NovaData/P500	02	Pentium III	800 Mhz	523 MB	100 GB	Windows NT 4.0

Tabela 4.5 – Equipamentos e Configurações de Apoio ao Desenvolvimento.

4.6.4 Ferramentas Utilizadas

4.6.4.1 Software Microsoft SQLServer 7.0

O *SQLServer* é um *SGBD* (Sistema Gerenciador de Base de Dados) relacional , cliente/servidor (*RDBMS*), projetado para oferecer alto desempenho e suporte a processamento de alto volume de dados. Sua plataforma é baseada em sistemas operacionais *Microsoft Windows*. Este sistema tem como linguagem nativa o *SQL (Sequence Query Language),* e o *Transact-SQL* que incorpora vários recursos como otimização de consultas, construção de programação e procedimentos armazenados.

Na sua versão *Microsoft SQLServer 7.0,* o gerenciador de base de dados da *Microsoft* possui um ambiente integrado de desenvolvimento, conhecido como *Enterprise Manager*, que permite administrar todos os recursos de qualquer servidor de banco de dados *SQL* acessível. Trata-se de uma interface que apresenta uma estrutura de árvore, onde

é possível visualizar todos os detalhes de um servidor de banco de dados, incluindo, além dos bancos de dados, todas as ferramentas de gerência necessárias.

Outra ferramenta importante é o *Query Analyzer* que permite executar instruções *SQL* e visualizar os resultados através de uma interface simples. A principal vantagem dessa ferramenta é admitir que várias janelas sejam abertas ao mesmo tempo, possibilitando realizar conexões simultâneas a bancos de dados diferentes.

O *Microsoft SQLServer 7.0,* instalado em um servidor dedicado à administração de dados, é o repositório das informações do protótipo difuso proposto nesse trabalho. A modelagem de dados desenvolvida para o projeto contempla tabelas para armazenagem de informações de servidores, de serviços de redes, e tabelas para armazenagem de informações do tráfego de rede coletado.

O objetivo da modelagem e implementação de uma base de dados para o protótipo é, justamente, facilitar o trabalho de consulta e manipulação de uma grande quantidade de informação, em tempo hábil à geração de alertas.

4.6.4.2 Data Transformation Server – *DTS*

O *Microsoft DTS* é um sofisticado mecanismo de busca e conversão de dados armazenados nos mais diferentes formatos, permitindo a junção e transformação das informações dispersas em um determinado ambiente. O *DTS*, na verdade, é o componente da solução *Microsoft* que possibilita a extração e transformação dos dados operacionais, de forma a prepará-los para o armazenamento adequado ao processamento orientado à análise.

O *DTS* é uma ferramenta de transformação flexível pelas seguintes razões:

- Apresenta as funcionalidades de um *workflow*;
- Permite especificar fontes e destinos de dados *ODBC*, incluindo *SGBDR*, arquivos texto ou planilhas;
- Possibilita chamada a programas externos;

- Proporciona serviço de *log* de erros de execução;
- Pode ser executado automaticamente, como um *job;*
- Funciona como uma plataforma para desenvolvimento de aplicação.

As características do produto *DTS*, acima relacionadas, foram determinantes na escolha dessa ferramenta para o desenvolvimento do protótipo. Como o principal objetivo da pesquisa é verificar a possibilidade de utilização das técnicas difusas, procurou-se utilizar uma ferramenta que permitisse um rápido desenvolvimento, com a devida garantia de tempo de execução. De posse dos resultados experimentais, e comprovado a eficácia das técnicas difusas no tratamento de tráfego de rede, poder-se-á projetar uma ferramenta que complete todas as fases do protótipo.

4.6.4.3 Sniffers

Os *Sniffers (*farejadores) são programas ou dispositivos que monitoram o fluxo de dados numa rede. Esses softwares são usados pelos gerentes de sistemas como ferramentas de análise. Os computadores em uma rede local, compartilham um meio físico, onde, normalmente, uma placa de rede *lê* os pacotes destinados a ela e descarta os demais. Um programa *Sniffer* coloca a placa de rede em modo promiscuo, possibilitando que um computador receba todos os pacotes que circulam no segmento de rede ao qual ele pertence.

A figura 4.3 demonstra a execução de um *sniffer*, onde é possível constatar detalhes da coleta do tráfego de rede.

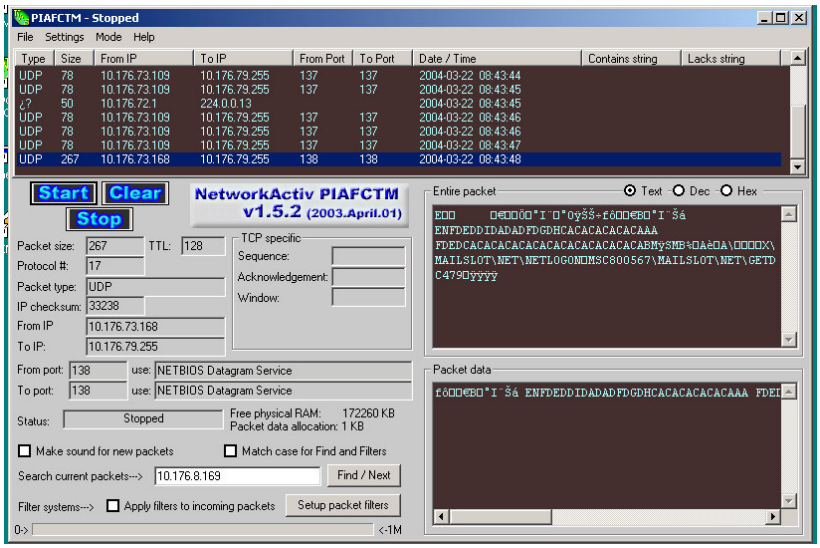


Figura 4.3: Software *NetworkActiv PIAFCTM V1.5.2*

O software *NetworkActiv PIAFCTM V1.5.2* foi escolhido para execução do processo de coleta de tráfego de rede. Os parâmetros para escolha do software foram, basicamente, a sua compatibilidade com o sistema operacional Windows, e a necessidade de utilização de uma versão de software aberta, testada e aceita pela área de segurança da empresa. O software possui as características padrão de um *sniffer*, gerando, durante sua execução, um *log* que é utilizado pelo protótipo como fonte de dados.

É importante mencionar que os softwares *sniffers* apresentam perdas de pacotes durante o processo de coleta, não retratando fielmente o tráfego que flui pelo segmento de rede.

Apesar dessa constatação, para o objetivo desse trabalho, pequenas variações na coleta do tráfego não trarão distorções que possam influir no processo, justamente, pelo fato de se estar utilizando um modelo difuso que, segundo AGUIAR e JUNIOR (1999), e BARBALHO(2001), são tolerantes a aproximações tanto em termos estruturais quanto na síntese das relações de pertinência.

4.7 Protótipo Implementado

O protótipo implementado caracteriza-se como um pacote (*package*), termo utilizado no ambiente *Microsoft DTS,* que pode ter sua execução agendada através dos serviços de agendamento do sistema operacional Windows, ou através dos serviços de agendamento de tarefas (*job´s*) do *SQLServer 7.0*. Esse pacote coordena e sequencializa todos os processos descritos na figura 4.1. Essa sequencialização tem início no processo de importação do *log* gerado por um *sniffer*, passa pela construção dos contadores e somadores, segue com o processo de inferência difusa, e finaliza com ou sem uma notificação sobre possíveis alterações no comportamento do segmento de rede monitorado.

É importante mencionar que o protótipo foi desenvolvido com a possibilidade de importação simultânea de vários *logs,* correspondentes aos possíveis segmentos de redes monitorados.

Algumas facilidades disponíveis na plataforma Windows foram utilizadas para proporcionar um ganho de produtividade e a conseqüente redução do tempo de desenvolvimento.

A utilização dos serviços de agendamento da plataforma *Microsoft* evitou a construção de temporizadores para controle de execução do pacote, e o serviço de correios (*Exchange)* da *Microsoft* possibilitou geração das notificações.

Cabe mencionar que todas as facilidades, acima mencionadas, são passíveis de serem implementadas em uma ferramenta que contemple todas as funcionalidades do protótipo, ou mesmo, migradas para as ferramentas correspondentes de outros sistemas operacionais.

A figura 4.5 ilustra o *package* que executa todo o processo de monitoramento. O *package* é composto por uma série de objetos (processos) que tem funções específicas dentro do ambiente de execução. Esses objetos são ilustrados graficamente na ferramenta

DTS, possibilitando, dessa forma, uma documentação e um melhor entendimento de sua seqüencialização.

Esses *packages* são considerados procedimentos armazenados, a exemplo de *procedures (procedimentos)* ou *views (visões),* fazendo parte do catálogo de objetos de um gerenciador de banco de dados.

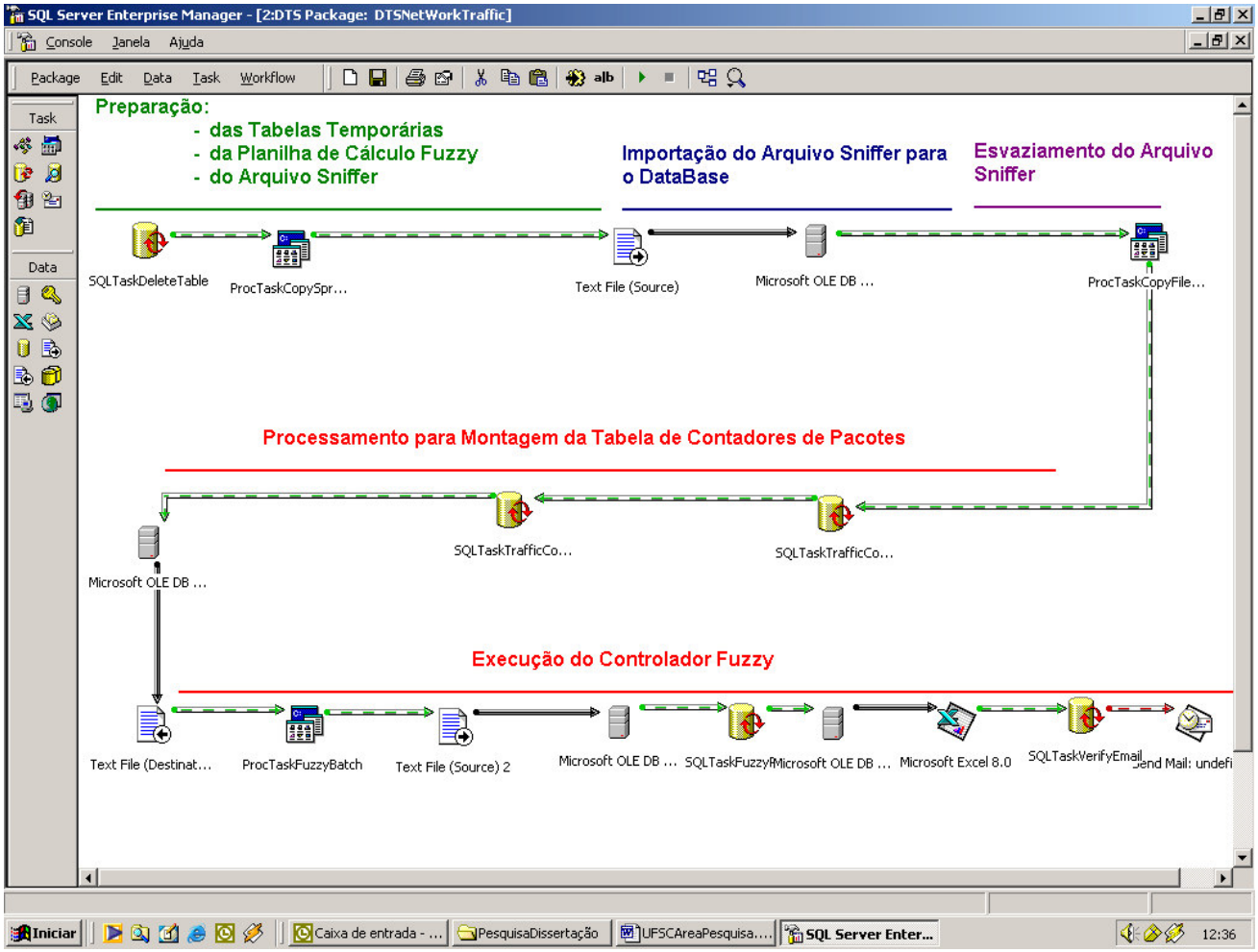


Figura 4.4: *Package* de Execução do Protótipo (Implementação)
(Protótipo Difuso)

5. AMBIENTE E RESULTADOS EXPERIMENTAIS

5.1 Ambiente de Produção

Os experimentos estão sendo realizados no ambiente de produção de rede da Empresa Brasileira de Correios e Telégrafos (ECT). A ECT possui uma série de segmentos de redes (redes locais) que atendem os diversos Estados e Municípios da Federação. Para esse projeto acadêmico, foi autorizada a monitoração do maior segmento de rede da Empresa no Estado de Santa Catarina. Esse segmento está localizado em Florianópolis, sede Regional da Empresa. O segmento possui uma infra-estrutura de cabeamento estruturado, categoria 5e, cujo *backbone* (espinha dorsal) tem uma vazão 100 MB/segundo. Esse *backbone* é composto de um *Switch Master*, com 10 portas para fibra ótica e 10 portas 10/100 MB, e vários outros *switches* instalados na sala de servidores e nos 12 andares do prédio que abriga o segmento.

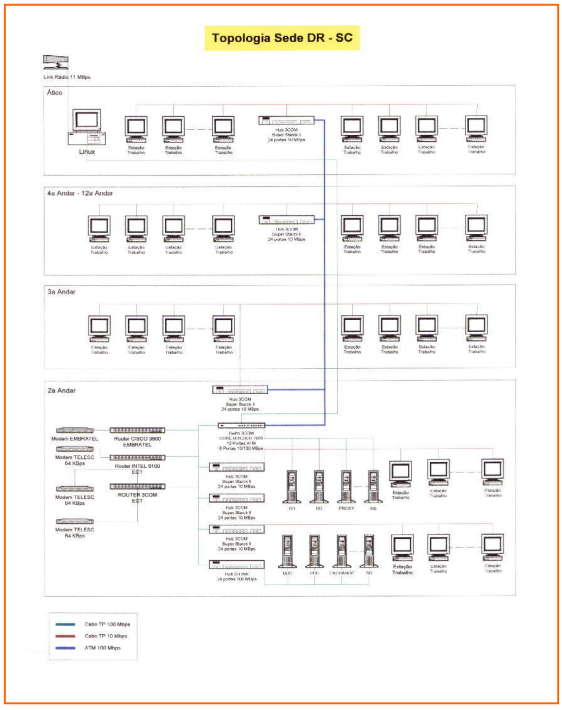


Figura 5.1: Topologia do Segmento de Rede Monitorado (2001)

Fazendo-se um inventário do segmento, constatou-se a existência de uma grande quantidade de dispositivos, entre os quais, menciona-se, no quadro abaixo, os mais significativos em termos de função e/ou quantidade.

Descrição	Sist. Operacional	Qt.
Hubs/ Switches		16
Servidores de Correio Eletrônico Microsoft Exchange	Windows NT 4.0	01
Servidores Microsoft Intranet Information Service	Windows NT 4.0	01
Servidores Proxies	Windows NT 4.0	01
Servidores de FTP	Windows NT 4.0	01
Servidores Autenticadores	Windows NT 4.0	02
Servidores Gerenciamento de Rede (Open View)	Windows NT 4.0	01
Servidores de Arquivo	Windows NT 4.0	02
Servidores de Banco de Dados SQLServer 6.5	Windows NT 4.0	02
Servidores de Banco de Dados SQLServer 2000	Windows 2003 Server	04
Servidor *SMS* (*Microsoft Systems Management Server)*	Windows NT 4.0	01
Servidores de Banco de Dados SQLServer 7.0	Windows NT 4.0	08
Servidores de Backup	Windows 2003 Server	02
Estações de Trabalho	Windows 2000 e Windows 98	600

Tabela 5.1: Parque Computacional do Segmento de Rede Monitorado

Para o gerenciamento desse significativo parque computacional, que se constitui em apenas uma das redes locais (segmentos de rede) que atende a Regional, a equipe de administração de rede utiliza softwares homologados, descrito a seguir na tabela 5.2, que tem como função principal dar suporte ao gerenciamento dos *link´s* responsáveis pela interconexão dos vários segmentos de rede.

Software	Descrição
NetCool/ISM	Verifica latência, falhas nos links (indisponibilidade e latência acima do contrato) por unidade, através de gráficos.
NetCool/OmniBus	Mostra os alarmes quantificados de falhas nos links (indisponibilidade)
Sismon/MRTG	Mostra a utilização dos links. O Sismon fornece informações consolidadas sobre a utilização de tráfego, além de mostrar informações como: largura de banda, porcentagem do tempo que o circuito ficou ocupando acima de 70%. Quando uma das localidades é escolhida, o gráfico correspondente do MRTG é apresentado com a utilização da banda dessa localidade.

Tabela 5.2: Softwares de Gerenciamento de Rede

Todas as informações sobre o tráfego de rede, restrito a um determinado segmento de rede, é obtido através da utilização de *sniffers,* que só são instalados quando existe um problema reportado à equipe de administração da rede. A partir da instalação dos *sniffers*, a equipe começa a análise dos *logs,* em seu formato bruto, a procura de padrões ou informações que possam levar a descoberta dos problemas reportados, conforme demonstrado na figura 5.2.

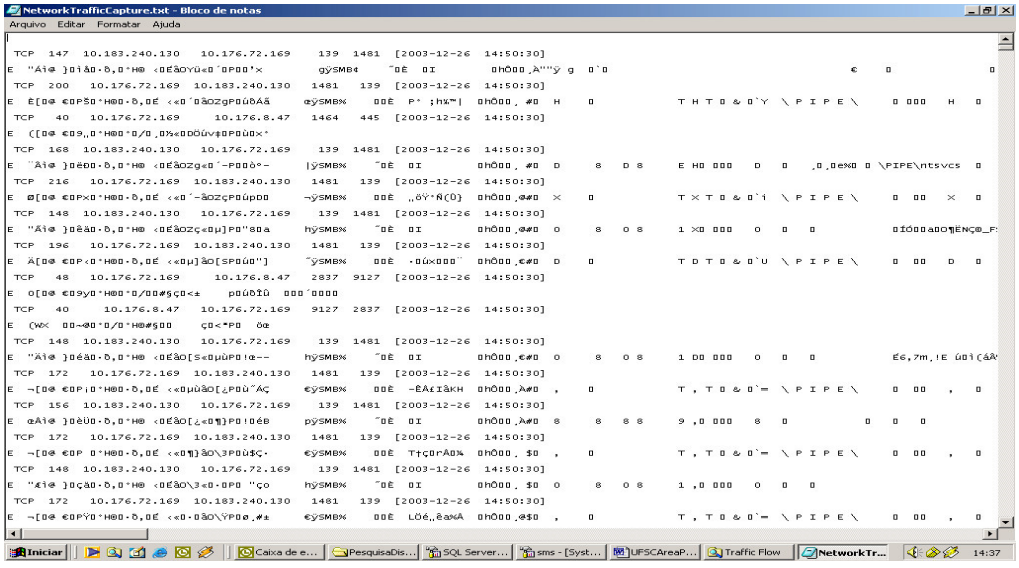


Figura 5.2: *Log* de Tráfego de Rede do *Sniffer NetworkActiv PIAFCTM V1.5.2*

5.1.1 Coleta do Tráfego (Sniffing)

É importante mencionar que muitos dos problemas de tráfego ficam restritos aos segmentos de rede, como por exemplo, as cópias de imensos arquivos entre estações de trabalho pertencentes a um mesmo segmento, ou ainda, a geração intensa de tráfego originado por problemas de softwares de servidores ou hardwares específicos.

As ferramentas homologadas e utilizadas para o monitoramento dos vários segmentos de rede fazem uso das informações de tráfego de entrada e saída dos *roteadores.* Os *roteadores* são equipamentos responsáveis pela interconexão de um segmento específico aos demais.

Para os casos acima exemplificados não existe tráfego direcionado para esses equipamentos, fazendo com que as ferramentas fiquem impossibilitadas de gerar alarmes ou notificações sobre problemas. Na prática estes problemas são reportados à equipe de administração de rede através de reclamações, formuladas por vários usuários, informando, geralmente, perdas de desempenho.

As escolhas do local de monitoramento e da fonte de coleta estão diretamente relacionadas com os objetivos das medições. Conforme mencionado na *RFC (1272)*, quando o objetivo de uma medição está restrita a um domínio (intra-domínio), a utilização de roteadores ou *hosts,* como fonte de informações de tráfego, é desaconselhada.

Dentro desse contexto, é proposta a utilização do protótipo difuso que se utiliza de um processo de *sniffing,* instalado no segmento de rede ou rede local, como fonte de informação sobre o tráfego de rede.

Após a apresentação do ambiente onde estão sendo realizados os experimentos, e antes de apresentar a forma de execução e os resultados experimentais, é fundamental demonstrar a forma de ajuste das funções de pertinência que estão associadas a cada variável de entrada do protótipo, bem como, demonstrar as regras que fazem parte do módulo de inferência do protótipo.

5.2. Resultados Experimentais

5.2.1 Funções de Pertinência

Para este trabalho experimental foram especificados 12 variáveis, entre contadores e somadores, que servem como entrada de dados do protótipo. Com objetivo de demonstrar o processo de ajuste dos parâmetros das funções de pertinência, escolheu-se uma das variáveis para demonstração.

Para melhor entendimento do processo de ajuste e motivado pelo fato do módulo de inferência ter sido desenvolvido em linguagem "C", cujo código fonte foi adaptado de JANG (1994) e incorporado ao *package DTS*, se utilizará ferramentas gráficas do software *MATLAB®* para possibilitar ilustrações.

O utilitário de lógica difusa *Fuzzy Logic Toolbox* demonstra a distribuição dos termos lingüísticos e funções de pertinência para a variável chamada de *Somador de Bytes por Segundo*.

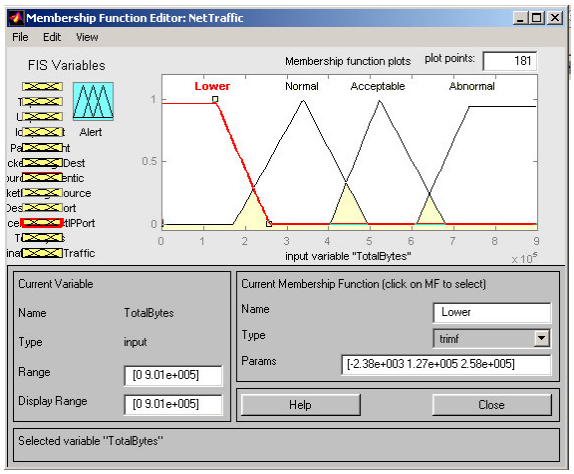


Figura 5.3: Somador de Bytes por Segundo
Utilitário *Fuzzy Logic Toolbox* do Software *MATLAB®*

A figura 5.3 retrata a distribuição simétrica das funções de pertinência que compõem o universo de discurso de uma das variáveis.

Conforme já mencionado, por desconhecimento sobre o perfil do tráfego de rede dos segmentos existentes, utilizou-se funções para apuração de valores máximos e mínimos de cada um dos contadores e somadores, com base em uma amostra de tráfego coletada durante uma semana.

Esses valores foram utilizados como universo de discurso de cada variável e, sobre eles, distribuídas, simetricamente, as quatro funções de pertinência associadas aos quatro termos lingüísticos que objetivam representar o comportamento do segmento de rede monitorado.

Para o ajuste dos parâmetros das funções de pertinência, que melhor retratasse o comportamento de cada termo lingüístico, recorreu-se ao algoritmo *Fuzzy C-Means (FCM),* implementado no software *MATLAB®.*

Esse algoritmo implementa um método de agrupamento difuso, muito utilizado como técnica de reconhecimento de padrões, onde um dado pode ser classificado em várias categorias (*clusters*) com diferentes graus de associação a cada uma delas.

Para melhor entendimento do funcionamento do algoritmo, procurou-se ilustrar o processo de ajustes das funções de pertinência através da figura 5.4.

Para essa ilustração foi utilizada uma amostra de 20 unidades de dados e informado a criação de três *clusters*. A figura demonstra a associação de cada uma das 20 unidades de dados com cada um dos três *clusters*.

A cor do dado define a sua proximidade com o *cluster* baseado na função de pertinência. A primeira ilustração retrata o resultado da inicialização do algoritmo. A segunda ilustração retrata o resultado do algoritmo após um processo iterativo de 8 passos, e por último o resultado após um processo de 37 passos.

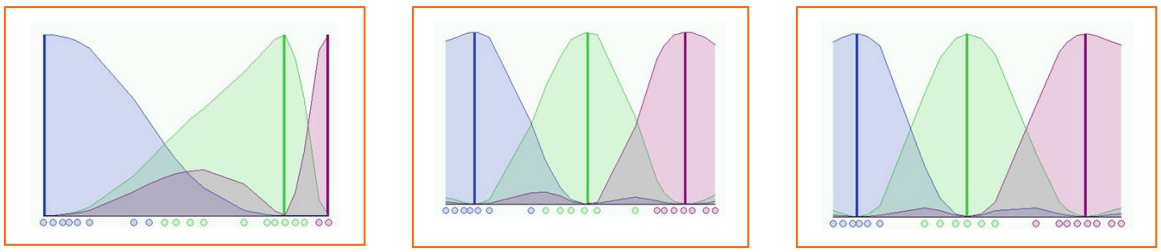


Figura 5.4: Ilustração do Funcionamento do Algoritmo *FCM*.
Adaptado de MATTEUCCI (2003)

Para o ajuste preciso dos parâmetros das funções de pertinência é imprescindível que a amostra utilizada no processo seja representativa, retratando com a maior fidelidade possível o comportamento do segmento em estudo. Com esse propósito, foram coletadas amostras de tráfego durante um período de três meses. Durante o processo de coleta foram constatados eventos de irregularidades que proporcionaram uma maior representatividade para a amostra.

É importante mencionar que mesmo nos processo neuro difusos, onde se procura automatizar a construção dos modelos difusos, as amostras utilizadas no processo de aprendizagem, teste e checagem têm que ser representativas, sob pena do modelo não descrever o comportamento do processo em estudo.

O utilitário de *clusterização* do software *MATLAB®*, ilustrado na figura 5.5, demonstra o resultado do agrupamento dos dados, composto pela amostra mencionada acima, que determinou a estruturação da variável *Somador Bytes por Segundo.*

Os eixos x e y retratam a localização dos quatro *clusters* relacionados com os termos lingüísticos dessa variável.

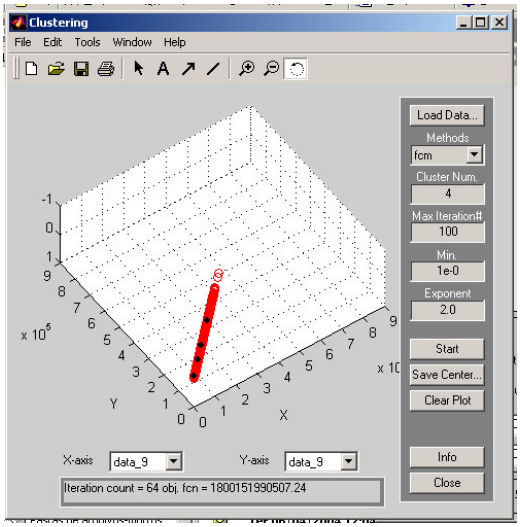


Figura 5.5: *Clusters* do Somador de Bytes por Segundo
Utilitário *Clusterização* do Software *MATLAB®*

A próxima ilustração, retratada na figura 5.6, demonstra, também, a localização dos quatro *clusters,* acima mencionados, em relação à massa de dados da amostra. O eixo x retrata os *clusters,* em número de quatro, e o eixo y a localização dos mesmos em relação à massa de dados. Para essa ilustração utilizou-se o algoritmo *FCM* para o processamento da amostra e o aplicativo de plotagem do software *MATLAB®* para ilustração.

Para o ajuste de todas as funções de pertinência, de cada uma das variáveis de entrada, utilizou-se os valores padrões sugerido pelo algoritmo. Esses valores são : 100 para o número de iterações; 2 para distância máxima entre o centro de cada *clusters* e um dado, e $1e^{-005}$ para o valor de erro.

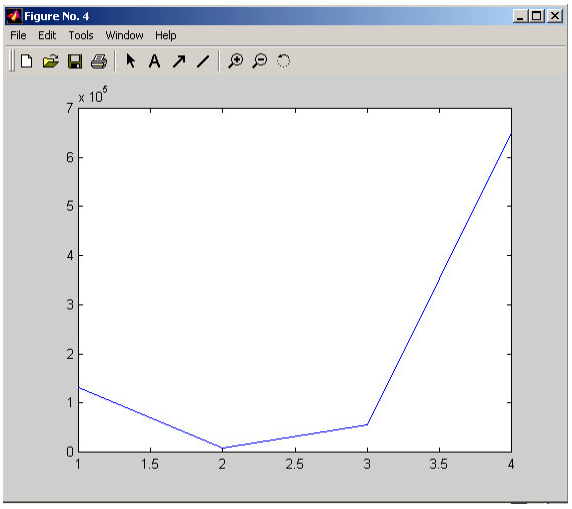


Figura 5.6: *Clusters* e a Amostra do Somador de Bytes por Segundo
Utilitário de Plotagem do Software *MATLAB®*

De posse dos valores dos *clusters*, que estabelecem um agrupamento padrão de cada termo lingüístico em cada variável de entrada, e da amostra utilizada para o ajuste dos parâmetros das funções de pertinência, recorreu-se, novamente, ao utilitário de plotagem do software *MATLAB®*, utilizando-se, agora, o eixo x para retratar a massa de dados da amostra, e o eixo y para retratar o grau de pertinência de cada dado amostral a cada um dos quatro *clusters.*

Através desse processo foi possível uma representação dos valores a serem ajustados para cada função de pertinência, conforme demonstra a figura 5.7.

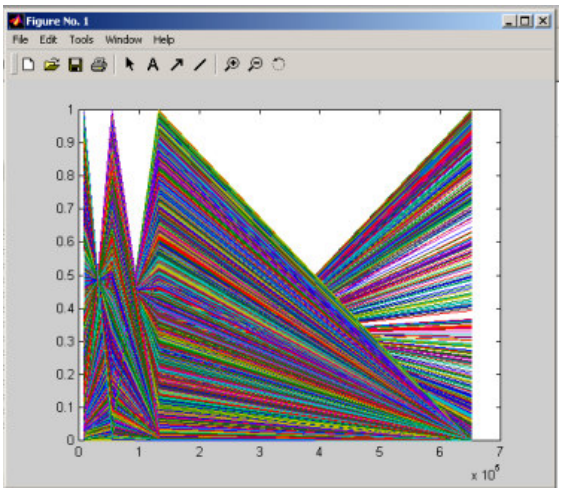


Figura 5.7: Distribuição Amostral do Somador de Bytes por Segundo
Utilitário de Plotagem do Software *MATLAB®*

Utilizou-se o mesmo procedimento para o ajuste de todas as funções de pertinência que compõem o protótipo. Uma vez determinado os ajustes, procedeu-se a atualização das funções implementadas no protótipo.

Pode-se verificar o resultado prático da adoção do algoritmo *FCM*, fazendo-se uma comparação entre as ilustrações das figuras 5.3 e 5.8, representadas graficamente através do utilitário de lógica difusa *Fuzzy Logic Toolbox* do software *MATLAB®* .

É importante mencionar que a utilização do algoritmo *FCM* surgiu da necessidade de se encontrar uma técnica de ajuste para as funções de pertinência. Como relatado, os especialistas da área de rede não tinham conhecimento que os permitisse definir os limites para cada termo lingüístico, mesmo porque o esforço de monitoramento da equipe está voltado à conectividade dos vários segmentos de rede. Logo, não é objetivo do trabalho um aprofundamento sobre o funcionamento e as pesquisas que envolvem a utilização do referido algoritmo. Utilizou-se o *FCM* dentro de uma perspectiva usuária, ou seja, o interesse era o resultado prático obtido pela execução do algoritmo sob a uma amostra de tráfego de rede.

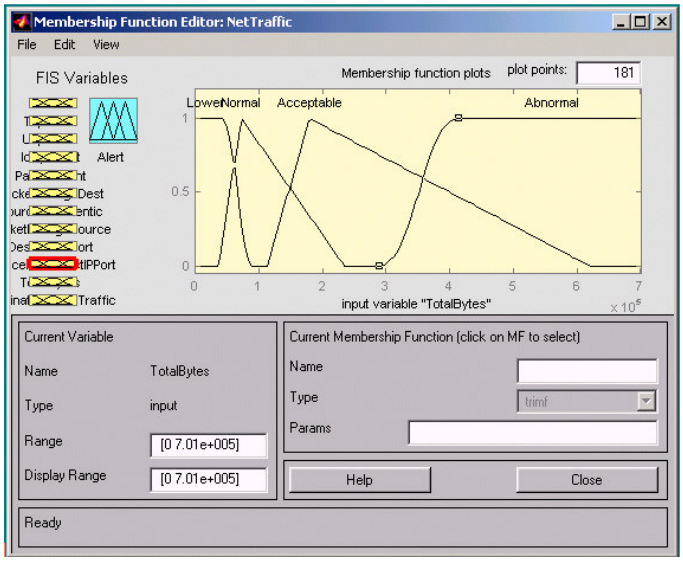


Figura 5.8: Ajuste das Funções de Pertinência do Somador de Bytes por Segundo. Utilitário Fuzzy Logic Toolbox do Software MATLAB®

5.2.2 Base de Regras

O processo de inferência de um sistema difuso está caracterizado na construção de sua base de regras.

A base de regras, no presente trabalho, é estruturada com objetivo de encontrar relações que possam evidenciar e caracterizar o comportamento de um segmento, através dos dados presentes no tráfego da rede. O sistema de regras estará completo quando puder responder satisfatoriamente a todas as ocorrências que caracterizem as mudanças de comportamento.

Com base revisão bibliográfica e se valendo da experiência dos profissionais que fazem a administração de rede, estruturou-se, inicialmente, 18 regras que objetivam descrever, dentro da perspectiva usuária, o comportamento do segmento de rede. Estas regras estão sendo validadas através de comparações entre comportamento dos segmentos e os resultados obtidos através do protótipo.

É importante mencionar que as variáveis, quando analisadas separadamente, não geram informações úteis, como por exemplo, o fato de, em determinado momento, a contagem dos fluxos de tráfego *TCP* está com valores baixos. Isto poderia não caracterizar qualquer anormalidade, mas esta informação acrescida da informação de que a contagem dos fluxos de tráfego *UDP* está com baixos valores, e a contagem dos fluxos de tráfego *ICMP* com valores aceitáveis, poderia remeter a uma informação sobre possíveis problemas físicos no segmento de rede.

Na figura 5.9, abaixo, está descrita a Base de Regras utilizada no modelo, sendo que, durante a confecção da dissertação e com a instalação do protótipo no ambiente de produção, poderá haver atualizações.

```
If      TcpCount  is Normal                and UdpCount is Normal                and
        IcmpCount is Normal                and PacketCount is Normal             and
        PacketForeignDest is Normal        and SourceDestIdentic is Normal       and
        PacketForeignSource is Normal      and DestinationPort is Normal         and
        SourceIPPortDestIPPort is Normal   and TotalBytes is Normal              and
        DestinationClientTraffic is Normal and DestinationHostTraffic is Normal
Then
        Alert is Normal

If      TcpCount  is Lower                 and UdpCount is Lower                 and
        IcmpCount is Lower                 and PacketCount is Lower              and
        PacketForeignDest is Lower         and SourceDestIdentic is Lower        and
        PacketForeignSource is Lower       and DestinationPort is Lower          and
        SourceIPPortDestIPPort is Lower    and TotalBytes is Lower               and
        DestinationClientTraffic is Lower  and DestinationHostTraffic is Lower
Then
        Alert is Abnormal

If      TcpCount  is Abnormal              or UdpCount is Abnormal               or
        IcmpCount is Abnormal              or PacketCount is Abnormal            or
        PacketForeignDest is Abnormal      or SourceDestIdentic is Abnormal      or
        PacketForeignSource is Abnormal    or DestinationPort is Abnormal        or
        SourceIPPortDestIPPort is Abnormal or TotalBytes is Abnormal             or
        DestinationClientTraffic is Abnormal or DestinationHostTraffic is Abnormal
Then
        Alert is Abnormal
```

If TcpCount is not Abnormal or UdpCount is not Abnormal or
IcmpCount is not Abnormal or PacketCount is not Abnormal or
PacketForeignDest is not Abnormal or SourceDestIdentic is not Abnormal or
PacketForeignSource is not Abnormal or DestinationPort is not Abnormal or
SourceIPPortDestIPPort is not Abnormal or TotalBytes is not Abnormal or
DestinationClientTraffic is not Abnormal or DestinationHostTraffic is not Abnormal
Then
Alert is Acceptable

If DestinationClientTraffic is Acceptable and TotalBytes is Acceptable
Then
Alert is Acceptable

If DestinationClientTraffic is Acceptable and DestinationHostTraffic is Lower
Then
Alert is Abnormal

If DestinationClientTraffic is Lower and DestinationHostTraffic is Acceptable
Then
Alert is Acceptable

If PacketForeignSource is Acceptable and DestinationHostTraffic is Acceptable and
SourceIPPortDestIPPort is Acceptable
Then
Alert is Abnormal

If PacketForeignSource is Acceptable and DestinationHostTraffic is Acceptable and
DestinationPort is Acceptable
Then
Alert is Abnormal

If PacketForeignSource is Acceptable and DestinationClientTraffic is Acceptable and
SourceIPPortDestIPPort is Acceptable
Then
Alert is Abnormal

If PacketForeignSource is Acceptable and TotalBytes is Acceptable and
SourceIPPortDestIPPort is Acceptable
Then
Alert is Abnormal

If SourceDestIdentic is Acceptable
Then
Alert is Abnormal

If PacketForeignDest is Acceptable and SourceIPPortDestIPPort is Acceptable
Then
Alert is Abnormal

If PacketForeignDest is Acceptable and DestinationPort is Acceptable
Then
Alert is Abnormal

```
If      PacketForeignDest is Acceptable and SourceIPPortDestIPPort is Acceptable and
        TotalBytes is Acceptable
Then
        Alert is Abnormal

If      TcpCount  is Lower      and  UdpCount is Lower  and  IcmpCount is Acceptable
Then
        Alert is Abnormal

If      TcpCount  is Lower      and  UdpCount is Lower  and  IcmpCount is Lower
Then
        Alert is Abnormal

If      IcmpCount is Acceptable
Then
        Alert is Abnormal

If      TotalBytes is Acceptable  or PacketCount is Acceptable  or
        TcpCount  is Acceptable   or   UdpCount   is Acceptable
Then
        Alert is Acceptable
```

Figura 5.9 : Base de Regras do Protótipo

O motor de inferência do modelo difuso está definido da seguinte forma:

- Para o conectivo *OU* utilizou-se a *normas-t min.*
- Para o conectivo *AND* utilizou-se a *co-normas max.*
- No processo de implicação utilizou-se a *normas-t min.*
- No processo de agregação utilizou-se a *co-normas max*.
- No processo de defuzzificação utilizou-se o método do *centróide*.

Os resultados experimentais estão sendo alcançados através da efetiva utilização do protótipo no ambiente de produção da empresa.

Como o trabalho está sendo realizado em um ambiente de produção, todo e qualquer teste tem que ser cuidadosamente analisado para evitar interferências que tragam danos para a área de negócios ou problemas de logística.

Após o término do período de testes e implantação do protótipo no ambiente de produção, vários eventos foram comprovadamente notificados pelo protótipo durante o seu monitoramento.

Antes de descrever os eventos, é importante que se tenha conhecimento dos termos lingüísticos e respectivos limites definidos para geração dos alertas. Estes valores ainda estão sendo ajustados conforme os resultados obtidos.

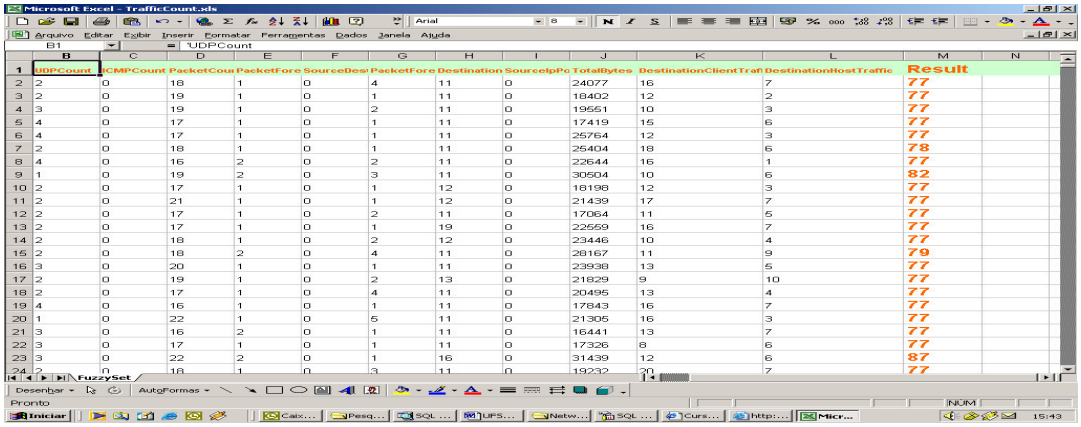
Intervalo	Termos lingüístico
00 a 40	Normal
30 a 70	Acceptable
60 a 100	Anormal

Tabela 5.3: Variável Difusa de Saída

5.2.3 Resultados – Caso 01

Um dos eventos rastreados notifica uma situação anormal resultante da execução de um backup de banco de dados que utilizou o segmento de rede para gravação do arquivo de backup. Este tipo procedimento não é indicado por motivos de segurança e pela conseqüente degradação do tempo de resposta da rede. Este procedimento foi executado, conscientemente, pela equipe de administração de dados, que trabalhava em uma migração de versão de software. O protótipo gerou a notificação sobre uma mudança de comportamento do segmento, informando altos valores escalares para descrever a criticidade do evento.

A figura 5.10 retrata uma das planilhas que foram anexadas aos *e-mail´s* e enviadas para a administração da rede, por ocasião do episódio.



	B	C	D	E	F	G	H	I	J	K	L	M	N
1	UDPCount	ICMPCount	PacketCou	PacketFore	SourceDes	PacketFore	Destination	SourceIpPo	TotalBytes	DestinationClientTraf	DestinationHostTraffic	Result	
2	2	0	18	1	0	4	11	0	24077	16	7	77	
3	2	0	19	1	0	1	11	0	18402	12	2	77	
4	3	0	19	1	0	2	11	0	19651	10	3	77	
5	4	0	17	1	0	1	11	0	17419	15	6	77	
6	4	0	17	1	0	1	11	0	25764	12	3	77	
7	2	0	18	1	0	1	11	0	25404	18	6	78	
8	4	0	16	2	0	2	11	0	22644	16	1	77	
9	1	0	19	2	0	3	11	0	30504	10	6	82	
10	2	0	17	1	0	1	12	0	18198	12	3	77	
11	2	0	21	1	0	1	12	0	21439	17	7	77	
12	2	0	17	1	0	2	11	0	17064	11	5	77	
13	2	0	17	1	0	1	19	0	22559	16	7	77	
14	2	0	18	1	0	2	12	0	23446	10	4	77	
15	2	0	18	2	0	4	11	0	28167	11	9	79	
16	3	0	20	1	0	1	11	0	23938	13	5	77	
17	2	0	19	1	0	2	13	0	21829	9	10	77	
18	2	0	17	1	0	4	11	0	20495	13	4	77	
19	4	0	16	1	0	1	11	0	17843	16	7	77	
20	1	0	22	1	0	6	11	0	21305	16	3	77	
21	3	0	16	2	0	1	11	0	16441	13	7	77	
22	3	0	17	1	0	1	11	0	17326	8	6	77	
23	3	0	22	2	0	1	16	0	31439	12	6	87	
24	2	0	18	1	0	3	11	0	19232	20	7	77	

Figura 5.10: Planilha de Notificação do Protótipo **(**Protótipo Difuso)

A planilha, acima ilustrada, detalha a situação de cada variável de entrada, através dos seus respectivos valores e informa o valor escalar obtido após o processo de defuzzificação. A notificação só é realizada quando a variável difusa *alerta* resulta em termos lingüísticos *Acceptable* ou *Anormal.*

5.2.4 Resultados – Caso 02

Outro evento interessante, capturado pelo protótipo, ocorreu por ocasião de um problema técnico em um dos *switches* que atendem os diversos andares da sede da Regional. O referido *switch* desencadeou um processo de *embaralhamento* de tráfego que provocou a geração de um grande tráfego *TCP*, *UDP,* e um aumento significativo no tráfego *ICMP*. Através da notificação do protótipo foi possível constatar, rapidamente, a origem do problema de degradação de rede.

5.2.5 Resultados – Caso 03

Com a utilização do protótipo constataram-se situações que já eram de conhecimento da equipe técnica, como por exemplo, a degradação do tempo de resposta do segmento de rede nos intervalos de horários entre 12:00 horas e 14:00 horas, e horários

próximos às 18:00 horas. Nessas ocasiões nota-se um aumento significativo, apesar de aceitável, na quantidade de bytes transmitidos/recebidos e na quantidade de tráfego com destino externo ao segmento.

5.2.6 Resultados – Caso 04

O comportamento do protótipo é alterado, exigindo um ajuste nos parâmetros das funções de pertinência, a cada inclusão de um novo segmento de rede. O mesmo não se verifica com a inclusão de novos dispositivos no segmento já monitorado. Este fato se deve a representatividade da amostra utilizada no ajuste inicial dos parâmetros do protótipo, que retrata a movimentação de dispositivos. Deve-se considerar que o segmento de rede monitorado possui um inventário significativo de dispositivos. Quando nos referimos a dispositivos, deve-se desconsiderar a inclusão de servidores, pois para os mesmos, obrigatoriamente, deve-se proceder atualizações nos parâmetros do protótipo, que possui variáveis dependentes da informação de números de *IP´s*, portas e protocolos.

Durante os primeiros ensaios experimentais, devido aos ajustes iniciais das funções de pertinência e a construção gradativa da base de regras, constatou-se a geração de uma grande quantidade de falsos alarmes. Uma vez concluída a etapa de ajuste de parâmetros, observou-se a inexistência de notificações por parte do protótipo, caracterizando assim o padrão de utilização do segmento e o sucesso na caracterização do comportamento desse segmento.

É importante mencionar que apesar da utilização do algoritmo *FCM,* para o ajuste dos parâmetros das funções de pertinência, estão sendo necessários mínimos ajustes em algumas variáveis, cujo particionamento dos termos lingüísticos geraram sobreposição das funções.

6. CONCLUSÃO E PROPOSTAS PARA TRABALHOS FUTUROS

Este trabalho teve como objetivo principal avaliar o uso das técnicas difusas na apuração de diferentes estados de comportamento de um segmento de rede, utilizando-se das possíveis relações existentes entre os dados que compõem o tráfego de rede. Com base na implementação de um protótipo e sua efetiva utilização em um ambiente de produção, pode-se constatar a eficácia do uso da técnica, superando, inclusive, as expectativas de tempo para obtenção de resultados.

O desenvolvimento do projeto trouxe a oportunidade de se buscar técnicas que permitissem, por exemplo, tratar um volume gigantesco de informações coletadas no tráfego de rede, e mesmo, utilizar técnicas de reconhecimento de padrões que possibilitassem ajustar os parâmetros do modelo difuso proposto.

A investigação empreendida neste trabalho mostrou, na prática, o potencial e as dificuldades relacionadas ao uso das técnicas difusas. A definição das variáveis de entrada e os ajustes das funções de pertinência são determinantes para o êxito do uso dessa técnica.

As principais dificuldades estão relacionadas a:

- Definição da camada de rede (arquitetura *TCP/IP)* a utilizar no monitoramento.
- Definição dos protocolos a serem contabilizados.
- Definição dos atributos dos protocolos a serem considerados.
- Definição das variáveis a serem correlacionadas de maneira a caracterizar o tipo de informação a ser disponibilizada.
- Definição das formas das funções de pertinência que melhor caracterizassem a variável considerada, e o ajuste de seus parâmetros.

- Procura de técnicas que permitissem ajustar os parâmetros das funções de pertinência com base em amostras de dados.

A eficácia na utilização de um modelo difuso em um ambiente de produção foi constatado através do protótipo desenvolvido. Entre os resultados obtidos pode-se citar:

- A possibilidade de minimizar as atividades de monitoramento das equipes de administração de rede, que passam a atuar no momento em que o comportamento usual da rede tenha sido afetado.

- A possibilidade de conhecer, melhor, o perfil dos usuários através do tráfego gerado.

- A possibilidade de descrever o conhecimento especializado através de uma base de regras.

- A oportunidade de conhecer o tráfego restrito a segmentos de redes específicos.

A bibliografia pesquisada retratou fielmente o processo de construção de softwares baseados na lógica difusa, mas uma dificuldade, não muito explorada na literatura, ficou bastante evidenciada. Essa dificuldade está relacionada com a parametrização dos componentes de um sistema difuso. Seria importante possibilitar atualizações no universo de discurso das variáveis, na forma e no espectro de valores das funções de pertinência e, ainda, atualizações na base de regras. No entanto, se percebe dificuldades de assimilação no uso prático da técnica por parte dos usuários. Esse fato é mencionado como uma das limitações dos sistemas inteligentes em GÜRER, KHAN e OGIER (1999). Talvez o grande interesse na utilização dos conceitos neuro difusos venham ao encontro dessas dificuldades. Não é intenção promover uma discussão sobre a utilização ou não de conceitos neuro difusos na concepção de sistemas difusos, apenas evidenciar uma dificuldade.

A adoção de diferentes tipos e formas de funções de pertinência, ou ainda, uma análise na base de regras do protótipo, poderia ser objeto de estudo para investigações futuras.

A utilização de simuladores para geração de vários tipos de tráfego também poderia ser objeto de interesse, onde o objetivo seria avaliar o comportamento do protótipo sob diferentes cenários.

Um detalhamento maior do tráfego pode ser obtida com a redefinição das variáveis do modelo, fazendo uso das informações do tipo de protocolo e portas, possibilitando um estudo sobre comportamento da camada de aplicações, ou seja, o comportamento das aplicações *FTP*, *TELNET*, *SMTP*, entre outras, bem como verificar o comportamento das conexões *TCP*. Esse detalhamento poderia, ainda, considerar faixas de horários de uso da rede, possibilitando uma avaliação mais específica sobre o perfil dos usuários da rede.

Apesar de inúmeras discussões a respeito da utilização ou não de técnicas neuro difusas, seria interessante, no nosso entendimento, submeter a um controlador neuro difuso uma amostra dos resultados evidenciados pelo protótipo com objetivo de comparar todo o processo desenvolvido nesta dissertação.

A evolução natural do uso das técnicas difusas poderá possibilitar o desenvolvimento de ferramentas de monitoração de tráfego que identifiquem problemas e sugiram ações corretivas.

REFERÊNCIAS

AGUIAR, H; JUNIOR, O. **Lógica Difusa – Aspectos Práticos e Aplicações**. Rio de Janeiro: Interciência, 1999.

AICKELIN, U; HESKETH, T. **Fuzzy Rule Learning in Intrusion Detection Systems**. Submitted & Under Review Paper. Computer Science - ASAP group, 2003. www.cs.matt.ac.uk, acesso em 20 de agosto de 2003.

ANGELIS, A. **Um Modelo de Tráfego de Rede para Aplicação de Técnicas de Controle Estatístico de Processos**. Tese de Doutorado. São Paulo: Instituto de Física de São Carlos - USP, 2003.

BARBALHO, S. M. V. **Sistema Baseados em Conhecimento e Lógica Difusa para Simulação do Processo de Chuva-Vazão**. Rio de Janeiro: Tese de Doutorado. UFRJ, 2001.

BARDOSSY, A.; DUCKSTEIN, L. **Fuzzy Rule-Based Modeling with Applications to Geophysical, Biological and Engineering Systems**. New York: CRC Press, Boca Raton 1995.

BAROUCHE, J. M.; SAPOTA, A. G. **Análise de Dados**. Rio de Janeiro: Zahar, 1982.

BASTOS, R .C. **Avaliação de Desempenho de Sistemas Educacionais: Um Abordagem Utilizando Conjuntos Difusos.** Florianópolis: Tese de Doutorado. Engenharia de Produção e Sistemas, UFSC, 1994.

BERNI, C. C. **Implementação de Hardware/Firmware de um Sensor Virtual Utilizando Algoritmo de Identificação Nebulosa**. Dissertação de Mestrado. São Paulo: Politécnica da Universidade de São Paulo, USP, 2004.

BEZDEK, C. J. **Fuzzy C-means.Pattern Recognition with Fuzzy Objective Function Algorithms.** USA: Plenum Press, 1981.

BEZDEK, C. J; PAL, K. S. **Fuzzy Models for Pattern Recognition: Methods That Search for Structures in Data**. Hardcover, 1992.

BITTENCOURT, G. **Breve História da Inteligência Artificial**. Notas de Aula. UFSC, 2003. http://www.lcmi.ufsc.br/gia/history/node1.html , acesso em 08 de agosto de 2003.

BOWERMAN, B.L.; OÇONNEL, R.T. **Time Series Forecasting – Unified Concepts and Computer Implementation.** PWS Publishers, 1987.

BRAGA, A. P; CARVALHO, A. P. F.; LUDEMIR, T. B. **Redes Neurais Artificiais: Teoria e Aplicações**. Rio de Janeiro: Livros Técnicos e Científicos, 2000.

BROWNLEE, N.; MILLS, C.; RUTH, G., **Traffic Flow Measurement: Architecture**. RFC 2722, IETF. October, 1999.

CAIDA. **Cooperative Association for Internet Data Analysis**. http://www.caida.org . acesso em 25 de agosto de 2003

CARBONELL, P; JIANG, Z. P.; PANWAR, S. S. **Fuzzy TCP: A Preliminary Study**. Proceedings Of the 15th IFAC World Congress (IFAC 2002), Barcelona, Spain , July, pp. 21-26, 2002.

CHANG C. L. **Data mining. In: Fuzzy-logic-based programming**. Series advances in fuzzy systems: Applications and theory, v.15. World Scientific,1997.

CHASBEN, H. **Inteligência Artificial na Educação: Sistemas Especialistas**. Notas de Aula.UFPR, 2003. http://www.cce.ufpr.br/~hamilton/iaed/iaed0003.htm , acesso em 25 de março de 2003.

CHATFIELD, C. **The Analysis of Time Series**. Chapman and Hall, 1984.

CHEN, J-L. HUANG, P-H. A fuzzy expert system for network fault management. IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence and Systems, Vol. 1, pp. 328-331, 1996.

CHRYSOSTOMOU, C; PITSILLIDES, A; ROSSIDES, L. **Fuzzy Logic Controlled RED: Congestion Control in TCP/IP Differentiated Services Networks**. Special Issue on The Management of Uncertainty in Computing Applications in Soft Computing Journal - A Fusion of Foundations, Methodologies and Applications, Vol 8, Number 2, pp. 79 - 92, December 2003.

COX, E.**The Seven Noble Truths of Fuzzy Logic** . Computer Design, Aptronix FuzzyNet, New York,1992.

DANTAS, M. **Tecnologias de Redes de Comunicação e Computadores**. Rio de Janeiro: Axcel Books, 2002.

DENIZ, J. **Neural Network Theory**. Article .MIT – Massachusetts Institute of Technology. U.S., 1998.

DICKERSON, J. E. ; DICKERSON, J. A. **Fuzzy Intrusion Detection**. IFSA World Congress and 20th North American Fuzzy Information Processing Society (NAFIPS) International Conference, Vancouver, British Columbia, Volume 3, pp. 1506-1510, July, 2001.

DICKERSON, J. E. ; DICKERSON, J. A. **Fuzzy Network Profiling for Instrusion Detection**. Proceedings of NAFIPS 19th International Conference of the North American Fuzzy Information Processing Society, Atlanta, July, pp. 301-306, 2000.

EDWARDS, A; CONNELL, N. A. D**. Expert Systems in Accounting**. Herfordshire, UK: Prentice Hall International (UK) Ltd, 1989.

EVSUKOFF, G. A; ALMEIDA, E. M. P. **Sistemas Inteligentes Neuro Fuzzy.** São Paulo: Manole, 2003.

FERNANDEZ, M. P., PEDROZA, A. C. P. and REZENDE, J. F. de. - **Converting QoS policy specification into fuzzy logic parameters** - 18 International Teletraffic Congress - Berlin – Alemanha, 2003.

FIALHO, F. A. P. **Raízes da Inteligência Artificial**. Seminário: Inteligência Artificial e Consciência.Artigo,2003.http://www.geocities.com/Athens/Sparta/1350/ia/raizes.html, acesso em 24 de março de 2003.

FLORES, G.; BRIDGES, S.; VAUGHN, R. **An Improved Algorithm for Fuzzy Data Mining for Intrusion Detection**. Proceedings of the North American Fuzzy Information Processing Society Conference (NAFIPS 2002), New Orleans, LA, June pp. 27-29, 2002.

FRANCESCHI, S. M.; ROCHA, M. ;WEBER, H.L.; WESTPHALL, C.B. **"Employing Remote Monitoring and Artificial Intelligence Techniques to Develop the Proactive Network Management".** Proceedings of the International Workshop on Applications of Neural Networks to Telecommunication 3. USA: Laurence Erlbaum Associates, Publishers. Mahwah, (NJ), 1997.

GANOULIS, J. G. **Engineering Risk Analysis of Water Pollution: probability and Fuzzy Sets**. New York: Cambridge, 1994.

GARCIA, A. C. B.; SICHMAN, J. S.; VAREJÃO, F. **Sistemas Inteligentes**. São Paulo: Manole, 2003.

GAVRILOV, A. V. **The model of associative memory of intelligent system**. The 6-th Russian-Korean International Symposium on Science and Technology. Proceedings. - No-vosibirsk, - Vol. 1.- pp. 174-177, 2002.

GOMES, J.; DASGUPTA, D. **Evolving Fuzzy Classifiers for Intrusion Detection**. Proceeding of the 2002 IEEE. Workshop on Information Assurance. United States Military Academy, West Point, NV, 2002.

GÜRER, D.; KHAN, I.; OGIER, R**. An Artificial Intelligence Approach to Network Fault Management**. SRI International, Menlo Park,California, USA, 1999.

HAN, J.; KAMBER, M. **Data Mining: Concepts and Techniques**. San Francisco: Morgan Kaufmann Publishers, 2001.

HARMON, P; KING, D. **Sistemas Especialistas**. Rio de Janeiro: Editora Campus, 1998.

HENRIQUES, O. A .P. **Compensação das Oscilações de Torque de um Acionamento de Relutância Chaveado**. Rio de Janeiro: Tese de Doutorado. UFRJ,1999.

JANG J-S. R. **ANFIS: adaptive-network-based fuzzy inference system**. IEEE Trans. on Systems, Man and Cybernetic 23 (3), pp. 665-685. 33. 1993.

JANG, R. **Stand-alone codes for fuzzy inference systems.** The MathWorks, Inc, 1994**.**

JONHSON, R. ; WICHERN. **Applied Multivariated Statistical**. USA: Prentice Hall, 1992.

JOY, B. **Why the future doesn't need us.** US: Random House Audio; Abridged edition, 2000.

KAGEYAMA, A.; LEONE, E. T. **Uma Tipologia de Municípios Paulistas com Base em Indicadores**. Notas de Aula. Campinas: Unicamp, 1999.

KARNOUSKOS, S.; VASILAKOS, A. **Neurofuzzy applications**: Active electronic mail. Proceedings of the 17th symposium on Proceedings of the 2002 ACM Symposium on applied computing. 2002.

KAUFMAN, L.; ROUSSEEUW, P.J. **Finding Groups in Data: An introducion to *Clusters* Analysis**. New York: Wiley, 1990.

KLIR, G.; YUAN, B.. **Fuzzy Sets and Fuzzy Logic** : Theory and Applications. USA: Prentice Hall, 1995.

LAI, W. S. **A Framework for Internet Traffic Engineering Measurement**. IETF. Internet Draft. Informational. Work in Progress. November, 2001.

LAMINEN, T.; KOIVISTO, H.; HONKANEN, T. **Profiling Network Applications with Fuzzy C-Means Clustering and Self-Organising Map**. International Conference on Fuzzy Systems and Knowledge Discovery, November 2002.

LAUREANO, M. A .P. **Sistemas para Identificação de Invasões.** Curitiba: Dissertação de Mestrado. PUC/PR, 2002.

LEE, C. **Fuzzy Logic in Control Systems: Fuzzy Logic Controller** – Part I. IEEE Transaction on System, Man and Cibernetics, vol.20 nº2 March/April, 1990.

LIEBOWITZ, Jay. **Introduction to Expert Systems**. Santa Cruz, CA: Mitchell Publishing, Inc, 1988.

MAMDANI, E. H. **Apllication of Fuzzy Algorithm for Control of Simple Dynamic Plant**. Proceeding of IEE Control and Science 121(12), pp. 1585-1588. 1975.

MATLAB®, MATHWORKS, INC. **Fuzzy Logic Toolbox**. 1998.

MATTEUCCI, M. **A Tutorial on Clustering Algorithms.** Article. Politecnico Di Milano, 2003.

MEECH, J. A. **Fuzzy Logic and Expert Systems**. Article. University of British of Columbia, Vancouver, Canada, 1997.

MELO, E. T. L. **Qualidade de Serviço em Redes IP com Diffserv: Avaliação através de Métricas**. Florianópolis: Dissertação de Mestrado. UFSC, 2001.

MEYSTEL, A .; MESSINAS, E. **The Challenge of Intelligent Systems**. Proceedings of the 15th IEEE International Symposium on Intelligent Control (ISIC 2000) Rio, Patras, GREECE 17-19 July, 2000.

MIYAMOTO, S. **Fuzzy sets in Information Retrieval and *Clusters* Analysis**. London: Kluwer, 1990.

NASSAR, M. S. **Sistemas Especialistas Difusos**. Notas de Aula. Florianópolis: UFSC, 2002.

NASSAR, M. S. **Tratamento da Incerteza**: Sistemas Especialistas Probabilísticos. Notas de Aula. Florianópolis: UFSC, 2001.

NDOUSSE, D. T. **Distributed Fuzzy Agentes: A Framework for Intelligent Network Monitoring.** IEEE International Conference on Communications, ICC '97, Towards the Knowledge Millennium, Monréal, Québec, Canada, Conference Record. IEEE, 867-871. 8-12 June 1997.

NETO, F.W. **Aplicando a Técnica de Séries Temporais em Gerenciamento Pró-Ativo de Redes de Computadores.** Anais do Simpósio Brasileiro de Redes de Computadores. Rio de Janeiro. Maio, 1998.

PALAZZO, A. L. ;CASTILHO M.V.J. **Algoritmos para Computação Evolutiva**. http://ia.ucpel.tche.br/ioia/Comput~1.doc , acesso em 09/01/2004.

PASSOS, E. L. **Inteligência Artificial e Sistemas Especialistas ao Alcance de Todos**. Rio de Janeiro: LTC – Livros Técnicos e Científicos. Ed.: Sociedade Beneficente Guilherme Guinle, 1989.

PEARSON, M. **Current techniques for Mesasuring and Modeling ATM Traffic**. Working Paper Series. New Zeland :University of Waikato, 1996.

PETERSON, T. **Microsoft SQL Server 2000 DTS**. Rio de Janeiro: Campos, 2001.

RABUSKE, R. A**. Inteligência Artificial.** Florianópolis: Editora da UFSC, 1995.

REZENDE, S**. Sistemas Inteligentes**. São Paulo: Manole, 2003.

RICARDO, P. M. CARRAPATOSO, M. E. **Análise e Modelização de Sistemas e Redes**. Portugal: Faculdade de Engenharia da Universidade do Porto, 2003.

RICH, E.; KNIGHT, K. **Inteligência Artificial**. São Paulo: Makron Books, 1993. 2a edição.

RICH, E.; KNIGHT, K. **Artificial Intelligence.** New York: McGraw-Hill, 1991.

ROCHA, M.A.; WESTPHALL, C.B. **"Proactive Management of Computer Networks using Artificial Intelligence Agents and Techniques"**. Proceedings of the Symposium on Integrated Network Management. San Diego (CA), USA. May, 1997.

ROSS, T. J. **Fuzzy Logic with Engineering Applications**. McGraw-Hill, 1995.

SHAW, I. ; SIMÕES, M. G. **Controle e Modelagem Fuzzy**. São Paulo: Edgard Blücher Ltda, 1a edição. 1999.

SHADBOLT, N. **Ambient Intelligence**. http://www.computer.org , acesso em julho/agosto de 2003.

SIMÕES, R. F. **Uma Análise de Fuzzy *Cluster***. Notas de Discussão No. 26. Belo Horizonte: UFMG, 2003.

SILVA, G. **Controle Não Linear**. Escola Superior de Tecnologia Setúbal. Artigo. Portugal. 2001.

SIMPLEWEB. **Simpleweb - links and information on network management** http://www.simpleweb.org/, acesso 15/03/2004.

SPECIALSKI, S. E. **Gerência de Redes de Computadores e Telecomunicações**.Notas de Aula. UFSC, 2002.

SUGENO, M; YASUKAWA, T. **A fuzzy-logic-based approach to qualitative modeling**. IEEE Trans. on Fuzzy Systems pp. 1, 7-31.1993.

TAKAGI, T.; SUGENO, M. **Fuzzy identification of systems and its applications to modeling and control**. IEEE Trans. on Systems, Man and Cybernetic SMC-15, pp. 116-132. 1985.

TANSCHEIY, R. **Lógica Fuzzy, Raciocínio Aproximado e Mecanismos de Inferência**. http://www.ica.ele.puc-rio.br , acesso em 01/09/2003.

TONSIG, S. **Simulando o Celebro**: Redes Neurais. Notas de Aula. São Paulo: PUCCamp, 2000.

VELLASCO, M. M. B. R. ICA: **Núcleo de Pesquisa em Inteligência Computacional Aplicada.** PUC-Rio. http://www.ica.ele.puc-rio , acesso em 01/09/2003.

VIEIRA, E.; WESTPHALL, C.B. **Using Fuzzy Specifications to Manage QoS**. Second IEEE Latin American Network Operations and Management Symposium.Belo Horizonte (MG), Brazil, 30 August, pp. 269-280. September, 2001.

WEBER, L.; KLEIN T. A. P. **Aplicações da Lógica Fuzzy em Software e Hardware.** Canoas (RS): Ed. Ulbra, 2003.

WESTPHALL, C.B. **Conception et développement de l'architecture d'administration d'un réseau métropolitain**. Thèse de doctorat nouveau régime. L' université Paul 109 Sabatier. Toulouse, le 16 juillet, 1991.

WESTPHALL, C.B.; KORMANN, L. F. **Usage of the TMN Concepts for Configuration Management of ATM Network**. International Symposium on Advanced Imaging and NetWork Technologies. Berlim, Alemanha Out. 7-11, 1996.

XEREZ, M. **Sistemas Difusos**. Notas de Aula. Santa Catarina: Unoesc, 1999.

ZADEH, L. **In Quest of Performance Metrics for Intelligent Systems** - A Challenge that Cannot be Met with Existing Methods. http://www.isd.mel.nist.gov/research_areas/research_engineering/Performance_Met

rics/PerMIS_2002_Proceedings___.Zadeh_Banquet_Speech.pdf, acesso em 02/09/2003.

ZADEH, L **Fuzzy sets. Information and Control,** pp**.** 338-353, 1965**.**

ZADEH, L. **Fuzzy Sets as the Basis for a Theory of Possibility**. **Fuzzy Sets and Systems**, 1, pp. 3-28, 1978.