Universidade Federal do Ceará

Mestrado em Ciência da Computação

Redes Neurais na Estimativa da Capacidade Requerida em Comutadores ATM

Miguel Franklin de Castro

DISSERTAÇÃO DE MESTRADO

Fortaleza-CE

24 de abril de 1999

Redes Neurais na Estimativa da Capacidade

Requerida em Comutadores ATM

Miguel Franklin de Castro

Dissertação de Mestrado

Redes Neurais na Estimativa da Capacidade Requerida em Comutadores ATM

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Miguel Franklin de Castro e aprovada pela Banca Examinadora.

Fortaleza-CE, 24 de abril de 1999.

Prof. Dr. Mauro Oliveira
(Orientador)

Dissertação apresentada ao Programa de Mestrado em Ciência da Computação da UFC, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Mestrado em Ciência da Computação

Universidade Federal do Ceará

Redes Neurais na Estimativa da Capacidade Requerida em Comutadores ATM

Miguel Franklin de Castro

Abril de 1999

Banca Examinadora:

- Prof. Dr. Mauro Oliveira
  Centro Federal de Educação Tecnológica do Ceará - CEFET-CE (Orientador)
- Prof. Dr. José Neuman de Souza
  Universidade Federal do Ceará - UFC
- Prof. Dr. Manoel Camillo Penna Neto
  Centro Federal de Educação Tecnológica do Paraná - CEFET-PR

Resumo

Capacidade Requerida em redes ATM é a quantidade mínima de largura de banda que deve ser alocada a uma fonte de tráfego de modo a satisfazer os parâmetros de Qualidade de Serviço desta rede. Esta informação pode ser utilizada como parâmetro para Controle de Admissão de Conexões (CAC) ou para Gerenciamento de Recursos. O objetivo deste trabalho é validar a utilização de Redes Neurais Artificiais para a estimativa da Capacidade Requerida em comutadores ATM. Com este intuito, foi desenvolvida uma abordagem específica baseada em parâmetros que definem o comportamento do tráfego agregado que chega a um comutador, em detrimento do uso de descritores de tráfego de aplicações em métodos analíticos.

Abstract

The Required Capacity, the minimum amount of bandwidth that must be alocated to a traffic source in order to grant the system's Quality of Service, can be used as parameter for Connection Admission Control (CAC) and Resource Management. The main purpose of this paper is to experiment and validate the usage of Artificial Neural Networks to estimate the required capacity on ATM networks, based on parameters that define the behavior of the aggregate traffic that reaches the switch, instead of using traffic descriptors on analytical methods.

Conteúdo

Lista de Tabelas

Lista de Figuras

Introdução

"The most likely way for the
world to be destroyed, most experts agree,
is by accident.
That's where we come in;
we're computer professionals.
We cause accidents."
– Nathaniel Borenstein

Este final de milênio tem sido marcado por várias inovações tecnológicas. Há quem diga que a produção científica nos últimos 10 anos é maior do que todo o restante da produção tecnológica e científica produzida pelo homem. Esta escalada não linear da produção científica tem se dado em todas as áreas da atividade humana, mas com especial destaque no que se refere à Tecnologia da Informação.

O final do século XX tem a marca da Tecnologia da Informação. Nada provocou tantas conseqüências científicas, culturais, sociais e econômicas quanto à rapidez com que a informação vem sendo cada vez mais rapidamente processada: "A notícia do assassinato de Abraham Lyncoln levou 13 dias para chegar à Europa enquanto os efeitos da queda da bolsa de valores em Hong Kong levou apenas 13 segundos para afetar outras bolsas em Nova Iorque, São Paulo, etc, por ocasião do primeiro *crash* do Plano Real"[1]. Percebe-se, assim, que as redes de com-

[1] Jornal Folha de São Paulo, Caderno de Economia

putadores que transportam eletronicamente esta informação tem tido um papel cada vez mais determinada no perfil tecnológico dos países com conseqüências imediatas na qualidade de vida destes povos. Isto tem motivado o desenvolvimento de redes cada vez mais rápidas, eficientes e – esta é a grande novidade do ano 2000 – para o transporte de multi-meios.

A tecnologia de pacotes imperou durante mais de três décadas como mecanismo eficiente no transporte de dados textuais, enquanto a comutação de circuitos vem exercendo há mais tempo (desde o início do século) a função do transporte do serviço de voz. Eis que as exigências são outras no milênio novo. O transporte de dados textuais, voz e vídeo integrados passou a ser o estado da arte.

A tecnologia ATM (*Asynchronous Transfer Mode*) vem sendo a solução dominante para a implementação desta integração de serviços tanto em nível local quanto em ambientes geograficamente dispersos.

Um dos principais diferenciais das redes ATM é a garantia de Qualidade de Serviço (QoS) para suas aplicações. Para que esta garantia possa ser oferecida, torna-se essencial o uso de bons mecanismos de Gerenciamento de Recursos. Portanto, recursos como largura de banda nos enlaces e espaço em *buffer* nos comutadores devem ser distribuídos da melhor maneira possível com o intuito de maximizar o uso da rede e a qualidade do serviço oferecido.

Uma das questões envolvidas no Gerenciamento de Recursos é a alocação lógica de largura de banda a aplicações. Como cada aplicação apresenta como uma de suas características a taxa máxima de transmissão (taxa de pico), uma solução para o problema seria a alocação de uma largura de banda suficiente para cobrir esta taxa de pico. Entretanto, este procedimento causa desperdício de recursos em momentos que a aplicação não está transmitindo a taxa de pico. Assim, uma outra possibilidade é a alocação de uma quantidade menor de largura de banda do que a taxa de pico para cada aplicação. No entanto, esta quantidade, denominada *Capacidade Requerida*, deve ser suficiente para que a Qualidade de Serviço desta aplicação e das demais que utilizam a rede não seja degradada.

Neste contexto de pró-atividade, este trabalho apresenta a RENATA (**Re**des **N**eurais **A**plicadas ao **T**ráfego **A**TM) [58], uma arquitetura com a finalidade de apresentar uma abordagem pró-

ativa baseada em redes neurais artificiais para a solução de problemas de tráfego ATM.

O objetivo deste trabalho é, então, validar a utilização da arquitetura RENATA para o problema da estimativa da Capacidade Requerida de aplicações em comutadores ATM. Esta estimativa é baseada em parâmetros que descrevem o comportamento do tráfego agregado, ao invés do uso de descritores de tráfego.

Este trabalho está organizado da seguinte forma:

- A Parte I apresenta a tecnologia ATM e uma introdução em Redes Neurais Artificiais, conceitos básicos indispensáveis à compreensão dos aspectos específicos que motivaram este trabalho.

- A Parte II trata do tráfego ATM e descreve em detalhes o conceito de Capacidade Requerida, elemento chave da presente proposta.

- A Parte III apresenta a contribuição do trabalho,que consiste na concepção de uma arquitetura baseada em Redes Neurais para a gerência pró-ativa de tráfego ATM e a validação desta arquitetura por intermédio de um estudo sobre a estimativa da Capacidade Requerida em comutadores ATM. Esta validação é acompanhada da descrição detalhada de um cenário de experimentação, bem como da prototipação de um mecanismo de estimativa da Capacidade Requerida baseado em medições de tráfego. Finalmente, ainda na Parte III, é feita uma análise dos resultados que comprovam a viabilidade da proposta.

Parte I

Conceitos

Capítulo 1

Tecnologia ATM

"Computers are useless.
They can only give you answers."
– Pablo Picasso

1.1 Introdução

A existência de redes distintas para telefonia e transporte de dados textuais, no passado recente, provocou a discussão sobre a utilização de uma infraestrutura única que contemplasse tanto às chamadas redes de telecomunicações (telefonia), quanto às redes de computadores (dados textuais). Posteriormente, a digitalização do serviço de telefonia iniciou a concretização desta idéia de integração. Entretanto, as diferenças entre comutação de circuitos e comutação de pacotes ainda se opunham a este ideal. As redes de comutação de circuitos, utilizadas no sistema de telefonia, desfavorece tráfegos variáveis, ao passo que a comutação de pacotes desfavorece mídias de vazão constante e sensíveis ao atraso. Paralelamente, a evolução das redes de comunicação de dados, representada pelo surgimento das redes locais e metropolitanas, e de redes públicas de comutação de pacotes, como a RENPAC (Rede Nacional de Comutação de Pacotes), trouxe uma maior interligação e, conseqüentemente, um maior crescimento desta infraestrutura de comunicação de dados. O acréscimo de estruturas de comunicação a nível

internacional, como a Internet, adicionou ainda mais complexidade a este conjunto.

Portanto, iniciou-se a busca de uma infraestrutura que representasse um meio termo entre os dois tipos de comutação. Inicialmente, era também necessário que esta integração tivesse como cenário a infraestrutura de telefonia já existente. Disso resultou o conceito das RDSI (Redes Digitais de Serviços Integrados).

A característica principal das RDSI é dar suporte a uma vasta gama de serviços através de um conjunto de interfaces de acesso únicas e padronizadas. Os tipos de tráfego que devem ser suportadas podem, então, apresentar características tanto de comutação de circuitos quanto de comutação de pacotes. As RDSI-FE (Redes Digitais de Serviços Integrados de Faixa Estreita), apresentadas inicialmente em 1972 pela CCITT (hoje ITU-T) e descritas em [26], representaram o último estágio da evolução da infraestrutura da rede telefônica.

A implementação da rede de comunicação por trás dos pontos de acesso ao usuário pode apresentar diferentes níveis de integração. Exemplos são [67]:

- Utilização de várias redes existentes, sendo apenas o acesso do usuário integrado (Figura 1.1);

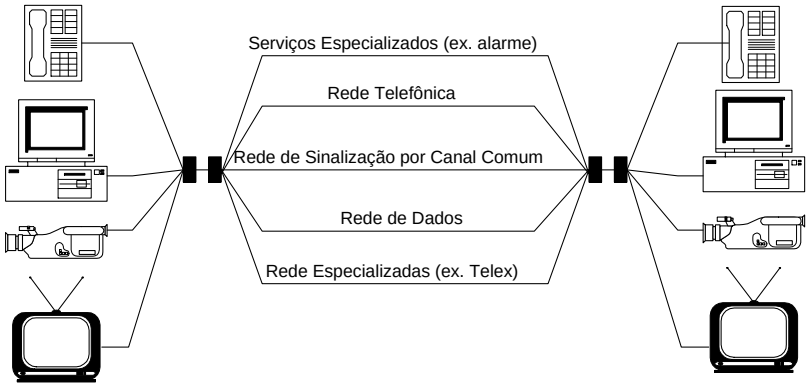


Figura 1.1: Primeiro Nível de Integração

- Implantação de uma rede única com recursos integrados, caso em que a integração é total (Figura 1.2).

Com as RDSI, a transmissão digital é prolongada até o ponto de acesso do usuário (linha do usuário). Este fato representou uma das evoluções em relação à rede telefônica, que se

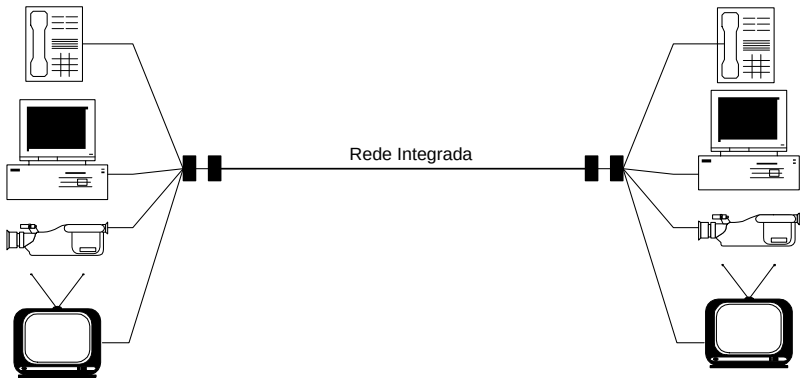


Figura 1.2: Integração Total

encontrava em um nível de digitalização apenas em nível de central e entroncamentos.

A tecnologia de transmissão, multiplexação e comutação utilizada para a transferência de informação é denominada pelo ITU-T como o *modo de transferência*. As RDSI-FE definem a utilização do STM (*Synchronous Transfer Mode*) como seu modo de transferência. Posteriormente, a disponibilidade de canais com maior capacidade (canais H) trouxeram a discussão sobre a flexibilidade do STM em acomodar serviços de diversas naturezas através de alocação de canais com taxas tão altas.

O Modo de Transferência Assíncrono (ATM) tenta eliminar as limitações do STM, tirando vantagem do ganho estatístico de serviços de tráfego com taxa variável, ao mesmo tempo garantindo um desempenho aceitável para serviços com tráfego contínuo [67]. Diferentemente do STM, no modo de transferência assíncrono a banda passante é dividida em segmentos de informação de tamanho fixo, denominados *células*. Cada célula possui um cabeçalho, seguido por um campo de informação. Desta forma, as células de cada conexão são identificadas pelos dados dos seus cabeçalhos, e não pela posição de um *frame* ao longo do tempo, como é o caso do STM [58].

Assim, as Redes Digitais de Serviços Integrados de Faixa Larga (RDSI-FL), evolução natural das RDSI-FE, adotaram o ATM como modo de transferência, criando-se, assim, o limiar entre aplicações e infraestruturas de faixa estreita e de faixa larga.

Entretanto, o ATM não se restringiu apenas à aplicação como modo de transferência das

RDSI-FL. Com o passar do tempo, viu-se no ATM um grande potencial para outros empregos. Devido à operação em grandes taxas de transmissão, o ATM foi adaptado para aplicação como *backbone* para redes de computadores do tipo MAN (*Metropolitan Area Network*) e WAN (*Wide Area Network*). Existe também uma forte tendência de aplicação do ATM em redes locais (LAN - *Local Area Network*), o que encorajou o desenvolvimento de padrões como *LAN Emulation* [2] e *IP over ATM* [42, 13].

Portanto, a partir de sua concepção, a tecnologia ATM vem alcançando novos patamares e assumindo diferentes formas, desviando-se de seu berço: as RDSI-FL.

1.2 Redes Digitais de Serviços Integrados

O avanço da tecnologia digital levou todos os tipos de mídia para mais próximo da *Integração de Serviços*. Portanto, a integração dos diversos tipos de informações em uma única infra-estrutura se tornou mais palpável.

Entretanto, a simples digitalização dos diversos serviços não foi suficiente para se chegar a este objetivo, visto que barreiras impostas pelas características de cada mídia ainda precisavam ser transpostas. Cada modelo de informação possui seus próprios requisitos, fazendo com que a unicidade desejada esteja dependente também da satisfação destas exigências. Portanto, um novo projeto que vise esta unificação deverá dar ênfase não só à integração, mais também à Qualidade de Serviço (QoS).

As Redes Digitais de Serviços Integrados (RDSI) – *Integrated Services Digital Network* (ISDN) – surgiram para suprir estas necessidades, realizando a integração desejada e utilizando uma infra-estrutura de telecomunicações já consolidada, permitindo inovações na forma de transportar digitalmente as informações.

O primeiro modelo de RDSI foi apresentado pela ITU-T[1] em 1984 [59]. Foram acrescentadas novas funções e características tanto de comutação de circuitos como de comutação de pacotes, com o intuito de prover tanto os serviços já existentes quanto novos tipos de maneira

[1] *International Telecommunication Union - Telecommunication Standardization Sector* (antigo CCITT)

integrada.

As vias digitais de acesso das RDSI são compostas a partir de combinações de múltiplos canais TDM (*Time Division Multiplexing*) síncronos. Os tipos de canais disponíveis para as RDSI são:

- Canal *Bearer* (*B*)

  Fornece uma taxa de transmissão de 64 Kbps.

- Canal *High-speed Bearer* (*H*)

  Combinação de canais *B* para obter larguras de banda maiores.

  - $H0$: 384 Kbps;
  - $H11$: 1.536 Kbps; (utilizado no $T1$)
  - $H12$: 1.920 Kbps; (utilizado no $E1$)
  - $H10$: 1.472 Kbps. (utilizado apenas nos E.U.A.)

- Canal *Dialogue* (*D*)

  Utilizado para realizar a negociação do estabelecimento da conexão entre o terminal e a rede. Duas larguras de banda estão disponíveis, variando de acordo com o tipo de interface:

  - 16 Kbps para interfaces do tipo Estrutura de acesso básico ($2B+D$)
  - 64 Kbps para interfaces do tipo Estrutura de acesso primário ($23B+D$ no $T1$ e $30B+D$ no $E1$)

As aglomerações de canais são classificadas de acordo com as variedades de canais utilizados. Estas opções são descritas na recomendação I.412 [28] da ITU-T. Os principais tipos são:

- Estrutura de acesso básico:

  Esta estrutura é composta por dois canais do tipo B e um canal do tipo D, totalizando 192 Kbps.

- Estrutura de acesso primário:

  Estrutura combinada de canais B e canais D de 60 Kbps com capacidades maiores, como o T1, de capacidade total de 1.544 Kbps, e o E1, totalizando 2.048 Kbps.

As RDSI se utilizavam da infra-estrutura de telefonia existente, o que limitava o poder destas redes. Além disto, surgia a necessidade de um maior poder de transmissão com o desenvolvimento da multimídia. Necessitava-se, então, de uma nova forma de transportar os dados digitais em uma infra-estrutura mais potente. Desta necessidade nasceram as Redes Digitais de Serviços Integrados de Faixa Larga (RDSI-FL). Desta forma, as RDSI já existentes foram nomenclaturadas de Redes Digitais de Serviços Integrados de Faixa Estreita (RDSI-FE), e uma fronteira foi definida entre as duas vertentes. Portanto, aplicações que necessitassem de uma quantidade de largura de banda inferior ou igual a 1,544 Mbps (nos EUA) ou 2,048 Mbps (na Europa e Japão) são consideradas aplicações de faixa estreita, enquanto aplicações que necessitem de largura de banda maior do que estes limites são classificadas como aplicações de faixa larga.

1.2.1 RDSI de Faixa Larga

As Redes Digitais de Serviços Integrados de Faixa Larga (RDSI-FL) representaram uma evolução sobre as RDSI's até então existentes, acrescentando-se mais largura de banda para que novas aplicações mais exigentes quando a recursos – mais notadamente as multimídia – pudessem ser desenvolvidas.

A ITU-T definiu o termo *faixa larga* como sendo "um serviço ou sistema que requer canais de transmissão capazes de suportar taxas de transmissão superiores à taxa de acesso primária (*primary rate*)" [59].

O conceito das RDSI-FL encontrado em [30] é como segue:

> *"As RDSI-FL são capazes de suportar conexões multiplexadas, permanentes, semipermanentes, ponto-a-ponto e ponto-a-multiponto; e de prover serviços reservados e permanentes sob demanda.*
>
> *Conexões nas RDSI-FL suportam comutação de pacotes e de circuitos com características mono ou multimídia, de natureza orientada ou não a conexões, e com configuração unidirecional ou bidirecional.*
>
> *As RDSI-FL contêm capacidades inteligentes com a finalidade de suportar serviços e ferramentas de operação e manutenção, bem como gerenciamento e controle de rede."*

A ITU-T classificou em [33] as possíveis aplicações de faixa larga em:

1. Serviços Conversacionais;
2. Serviços de Recuperação;
3. Serviços de Mensagens;
4. Serviços de Distribuição:
    a) *Com* controle individual de apresentação pelo usuário;
    b) *Sem* controle individual de apresentação pelo usuário.

Os serviços conversacionais, de recuperação e de mensagens são considerados interativos. Os serviços de distribuição são subdivididos em duas subcategorias: distribuição com e sem a intervenção do usuário. A Tabela 1.1 exemplifica cada tipo de serviço de banda larga.

Os serviços conversacionais envolvem comunicação de informações em tempo-real, podendo ser uni ou bidirecional. Um exemplo de serviço conversacional citado pela ITU-T é a Vídeo-telefonia, onde os interlocutores se comunicam por voz e imagem em movimento, de uma forma ponto-a-ponto. A Vídeo-conferência é uma variação do serviço de Vídeo-telefonia, onde podem existir mais de dois interlocutores se comunicando através de voz, vídeo, e ainda

Tipo de Informação	Exemplo de Serviço
CONVERSACIONAL	
Som e imagens em movimento	Videofone de Banda Larga
	Videoconferência de Banda Larga
MENSAGENS	
Vídeo e som	Serviço de Vídeo-mail
Documentos Multimídia	Serviço de Document-mail
RECUPERAÇÃO	
Texto, dados, imagens e sons	Videotexto de Banda Larga
	Serviço de Recuperação de Vídeo
	Serviço de Recuperação de Dados
	Serviço de Recuperação de Documentos
DISTRIBUIÇÃO	
(sem interação do usuário)	
Som e Imagens em Movimento	Serviço de Distribuição de Vídeo
Vídeo	TV Paga
	Distribuição de TV de Alta-Resolução
DISTRIBUIÇÃO	
(com interação do usuário)	
Texto, Gráfico, som e imagens	Vídeográfico de Banda Larga

Tabela 1.1: Exemplos de Serviços de Banda Larga

através de uma imagem representando um quadro de anotações. Este serviço também é classificado pela ITU-T como um serviço de banda larga conversacional.

Diferentemente dos conversacionais, os serviços de mensagens não apresentam característica de tempo-real. Ao invés disto, realizam funções *store-and-forward*, de *mailbox* e de manipulação de mensagens. Nesta categoria incluem-se o correio eletrônico de voz e vídeo e o correio eletrônico de documentos, os quais podem ter características multimídia.

Os serviços de recuperação envolvem a capacidade do usuário de recuperar informações armazenadas em algum lugar da rede. Portanto, uma central deve armazenar as informações, que são transferidas através da rede a partir de uma solicitação do usuário até o seu ponto de rede (terminal). Um exemplo deste tipo de serviço é o Vídeo-texto Multimídia, onde as informações envolvidas não se restringem apenas ao texto, como o sistema de Vídeo-texto já difundido atualmente. Outro exemplo destes serviços é o vídeo sob demanda, onde o usuário solicita um vídeo digital do tipo MPEG, por exemplo, que pode ter como finalidade apenas o entretenimento (substituindo as TV's por assinatura ou aluguel de vídeo) ou fins educacionais

(educação a distância).

Os serviços de distribuição são tipicamente os serviços *multicast*, onde uma fonte gera informações que são transmitidas a mais de um receptor. Esta categoria é subdividida em dependentes ou não da interação do usuário. A primeira subcategoria diz respeito a informações que são repassadas a pontos autorizados (assinantes) de forma contínua, ordenada e linear, *i.e.*, o usuário não participa da decisão acerca da forma que esta informação é apresentada. O segundo subtipo envolve os serviços cuja apresentação das informações podem ser alteradas pelo usuário quanto à ordem ou quanto à seleção de conteúdo. Um exemplo de serviço de distribuição é a substituição eletrônica de algumas mídias impressas, como jornal ou revista. No caso, periodicamente estas informações são enviadas integralmente aos seus assinantes. Um exemplo de serviço de distribuição dependente da interação humana é uma sofisticação dos serviços eletrônicos de mídia pública citados no exemplo anterior. No caso, a ordem e a forma da apresentação e a seleção do conteúdo competem ao usuário (leitor), que pode, por exemplo, selecionar notícias e reportagens sobre os assuntos que mais o interessam, e colocá-los na ordem que desejar.

1.2.2 Modelo de Referência das RDSI-FL

O processo de desenvolvimento e padronização das RDSI-FL tem sido realizado por órgãos diversos e finalizado pela ITU-T. A primeira recomendação da ITU-T sobre as RDSI-FL foi publicada em 1988. Depois disto, 13 outras especificações acerca destas redes foram aprovadas até 1990. Dentre estas recomendações está a descrição do Modelo de Referência dos Protocolos das RDSI-FL, ilustrado na Figura 1.3.

A Camada Física é responsável por funções como temporização de bits, sincronização e taxa de erro de bits máxima.

A Camada do Modo de Transferência é responsável pela representação das informações provenientes das camadas superiores à Camada Física, e vice-versa.

A Camada de Adaptação é responsável pela convergência dos diversos tipos de serviços oferecidos ao padrão imposto pelo modo de transferência. Um exemplo de operação de adaptação

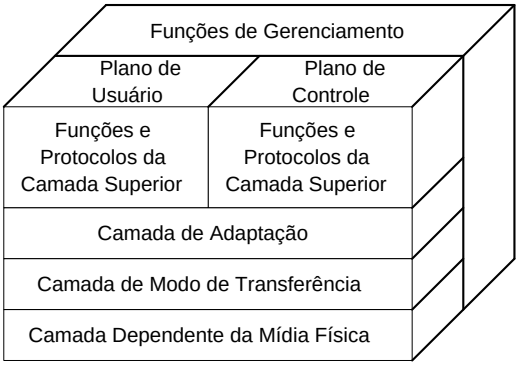


Figura 1.3: Modelo de Referência das RDSI-FL

é a conversão de serviços não orientados a conexão e serviços de comutação de circuitos e pacotes.

O Plano de Controle e o Plano de Usuário formam as camadas superiores, onde o primeiro é responsável por funções operacionais na rede, como estabelecimento e liberação de conexões, e o último se responsabiliza pela transferência de informações das aplicações de usuários, tão logo a conexão seja estabelecida através do plano de controle.

Por fim, as Funções de Gerenciamento são responsáveis por fornecer interface para monitoramento e controle de todas as camadas. Ao contrário das redes até então existentes, as RDSI-FL já foram concebidas com a preocupação prévia com a gerência. Isto torna a tarefa de gerenciamento mais organizada, transparente e integrada e menos dispendiosa ao sistema.

1.2.3 Configuração de Referência

A função da Configuração de Referência das RDSI-FL é especificar o conjunto de interfaces padrão para a operacionalização destas redes, definindo papéis e funções de cada unidade.

A Configuração de Referência das RDSI-FL teve como base o modelo das RDSI-FE, já definido na especificação I.411 da ITU-T [27], com a diferença das RDSI-FL apresentarem o caractere "*B*" na nomenclatura de seus componentes, o que indica serem de faixa larga (*Broadband*). Outras diferenças são observadas entre as duas configurações de referência, onde a maior

parte delas diz respeito à substituição do Modo de Transferência Síncrono (STM - *Synchronous Transfer Mode*) das RDSI-FE pelo Modo de Transferência Assíncrono (ATM - *Asynchronous Transfer Mode*). A introdução do ATM nas RDSI-FL representou a superação das limitações do STM quanto ao potencial de atuação. Representou, também, a introdução de novos mecanismos que aumentam a utilização dos enlaces de dados, através do aproveitamento estatístico das aplicações que operam a taxa de transmissão variável [67].

A Figura 1.4 ilustra a Configuração de Referência das RDSI-FL [29].

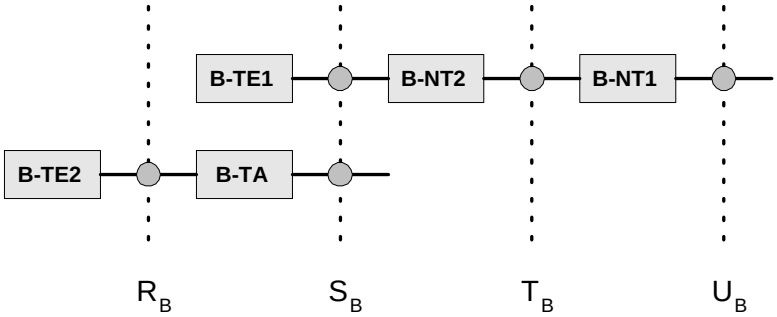


Figura 1.4: Configuração de Referência das RDSI-FL

O Grupo Funcional B-NT1 apresenta funções de mais baixo nível, como a terminação da linha de transmissão e a terminação física e elétrica da linha de assinante. Diferentemente das RDSI-FE, o B-NT1 não pode ser utilizado para multiplexação de células, o que limita a existência de apenas uma interface T_B [67]. Este equipamento pode ser controlado pela companhia provedora de serviços de rede, definindo, assim, a limitação física da rede.

O B-NT2 provê funções de mais alto nível, como a multiplexação e demultiplexação do tráfego, alocação de recursos, enfileiramento nos *buffers* e chaveamento de conexões internas. Assim como nas RDSI-FE, este equipamento é utilizado como concentrador de rede.

Os Equipamentos Terminais (TE's - *Terminal Equipments*) são grupos funcionais que efetivamente fazem uso da rede. São eles: B-TE1 e B-TE2.

O B-TE1 representa a terminação da interface padrão na RDSI-FL, realizando a terminação de todos os protocolos de rede, desde o mais baixo nível até a última camada.

Os B-TE2 são utilizados no caso da presença de uma interface RDSI-FL fora dos padrões. Neste caso, necessitam de Adaptadores de Terminal (B-TA) para que sejam interconectados à RDSI-FL. Portanto, o B-TA realiza todas as operações de conversão necessárias para a conexão de uma interface qualquer a uma RDSI-FL.

Outro equipamento, chamado Unidade de Interoperação (IWU - *Interworking Unit*), pode ser utilizado para a interconexão de LAN's como *Token Ring* ou *Ethernet* à RDSI-FL, fornecendo mecanismos específicos de acesso ao meio para realizar esta interconexão.

Os Pontos de Referência do modelo das RDSI-FL incluem: U_B, T_B, S_B e R_B. O ponto U_B representa a terminação da linha do assinante, representando a fronteira entre a rede pública e o ambiente privado. O T_B separa o equipamento fornecido pelo provedor e o equipamento do usuário. O ponto T_B é único para cada B-NT1. O Ponto S_B corresponde à interface individual de um terminal RDSI-FL, que separa o equipamento do usuário do equipamento de rede. O R_B representa uma interface qualquer entre um equipamento fora do padrão RDSI-FL e um Equipamento de Adaptação (B-TA).

A Figura 1.5 apresenta alguns exemplos de configurações de RDSI-FL. A configuração (A) apresenta um B-NT2 utilizado como concentrador de dois equipamentos B-TE1 através de pontos S_B, e um equipamento B-TE2, fora dos padrões RDSI-FL, que é adaptado ao B-NT2 através de um Equipamento de Adaptação (B-TA). A configuração (B) mostra um B-TE1 e uma rede local ou metropolitana (LAN ou MAN), interligados a um B-NT2 através de seus pontos S_B. A interconexão da LAN ou MAN é feita através do equipamento IWU. Na configuração (C), as funcionalidades do terminal (B-TE1) e do equipamento de rede (B-NT2) são combinados em um único equipamento. A configuração (D) ocorre quando a provedora de serviços de rede incorpora em um mesmo equipamento as funções do B-NT1 e do B-NT2.

Os trabalhos de padronização das RDSI-FL geraram 13 especificações na ITU-T entre 1988 e 1990. Logo após, em 1991, com a criação do ATM Forum, grupos de pesquisas criaram diversas *recomendações* acerca desta tecnologia. Apesar desta organização não ter poder de padronização, suas recomendações servem de base para a implementação nas indústrias, garantindo a interoperabilidade entre produtos.

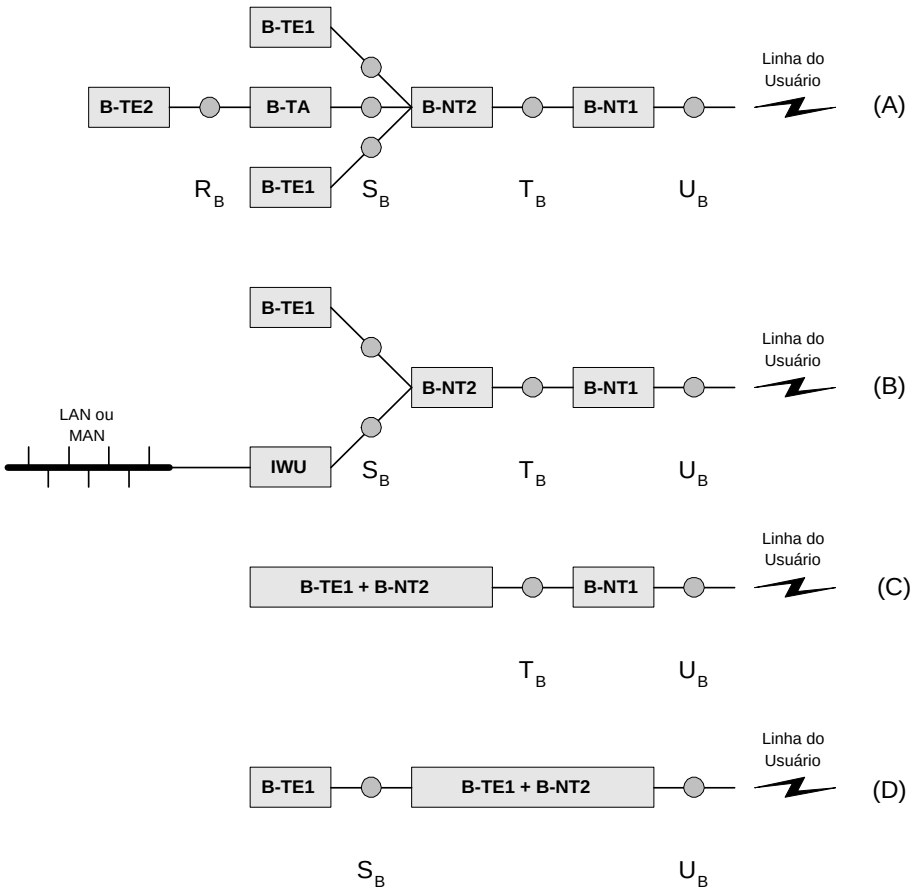


Figura 1.5: Exemplos de Topologias RDSI-FL

Um exemplo de resolução do ATM Forum foi a definição das Interfaces Usuário-Rede (UNI - *User-Network Interface*) pública e privada e da Interface de Nó de Rede (NNI - *Network Node Interface*). A UNI privada representa o ponto de referência S_B, enquanto a UNI pública é formada pelos pontos T_B e U_B. Atualmente, os esforços da ATM Forum estão mais direcionados à aplicação do ATM em redes locais.

Além da ITU-T e do ATM Forum, a IETF (*Internet Engineering Task Force*) também trabalha no sentido do desenvolvimento das redes ATM. Esta organização, que é responsável pelo desenvolvimento da Internet, estuda a integração das redes ATM ao TCP/IP, vigente atualmente na Internet. Assim, a IETF pretende oferecer um maior potencial para aplicações multimídia na Internet, através da adoção do ATM. O *Classical IP over ATM* é um exemplo de trabalho desen-

volvido pela IETF, assim como o *ARP over ATM*, ambos lançados em 1994 na RFC 1577 [42].

1.3 Modo de Transferência Assíncrono

A tecnologia ATM foi inicialmente adotada como modo de transferência das RDSI-FL. Porém, após algum tempo, esta tecnologia ganhou novos cenários. O ATM tem despertado o interesse como tecnologia de suporte para *backbones* de alta velocidade para redes WAN e MAN, e também como suporte às LANs.

O ATM representou uma evolução em relação ao STM, modo de transferência utilizado nas RDSI-FE. Enquanto no STM a banda passante é compartilhada através de *slots* de tempo, no ATM a comutação é feita através da divisão da banda passante em *slots* de células (*cell slots*), onde as unidades de cada conexão são dispostas assincronamente e identificadas através do cabeçalho encontrado em cada *cell slot*. As unidades de dados transportadas uma em cada *cell slot* são denominadas "*células*". Portanto, diferentemente do STM, não é necessário no ATM uma divisão igualitária e cronometrada da largura de banda disponível, visto que cada segmento de dados já é devidamente identificado.

O ATM é um modo de transferência orientado a conexões que tipicamente opera em altas taxas de transmissão em modo *full-duplex*. Apesar da orientação para conexões, o ATM é também capaz de se adaptar de modo a suportar serviços *connectionless*. Uma das características marcantes do ATM pela possibilidade de garantir Qualidade de Serviço (QoS) às conexões. Esta tarefa se torna ainda mais completa porque o ATM suporta serviços tanto de comutação de pacotes quanto de comutação de circuitos.

Em termos gerais, uma rede ATM é composta por equipamentos comutadores (*switches*) e Sistemas Finais (*End Systems*), interligados através de enlaces (*links*) físicos, com taxa de transmissão que pode ultrapassar o patamar dos Gigabits por segundo. Cada Sistema Final possui pelo menos uma Interface Usuário Rede (UNI - *User-Network Interface*), interligada através de um *link* físico a um comutador. Comutadores são interligados por *links* físicos através de suas Interfaces de Nó de Rede (NNI - *Network Node Interface*). Os links físicos suportam

comunicação *full-duplex, i.e.*, em dois sentidos independentemente. A estrutura física de uma rede ATM é exemplificada na Figura 1.6.

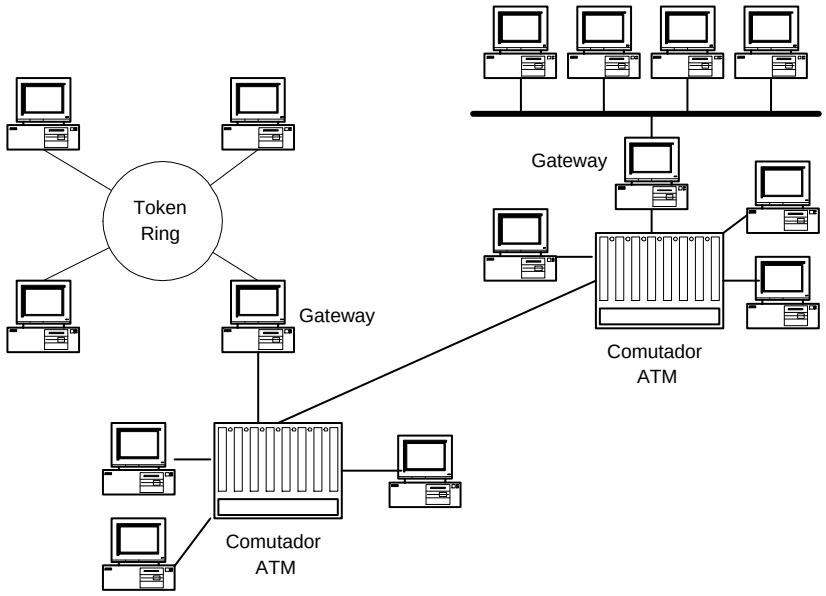


Figura 1.6: Estrutura Física de uma Rede ATM

1.4 Célula ATM

Nas redes ATM, as informações são transportadas sob a forma de pacotes de tamanho fixo (53 octetos) denominados "*células*". Uma célula ATM possui 5 octetos de cabeçalho (PCI - *Protocol Control Information*) e 48 octetos de carga útil (*payload*).

O cabeçalho de uma célula ATM é composto pelos seguintes campos:

- VPI (*Virtual Path Identifier*)

  Identifica o Caminho Virtual (VP) na qual a célula trafega.

- VCI (*Virtual Channel Identifier*)

  Identifica o Canal Virtual (VC) à qual a célula está ligada. O par VPI/VCI identifica biunivocamente a conexão à qual a célula pertence.

- PT (*Payload Type*)

  Indica que tipo de informação está sendo transportado na carga da célula.

- CLP (*Cell Loss Priority*)

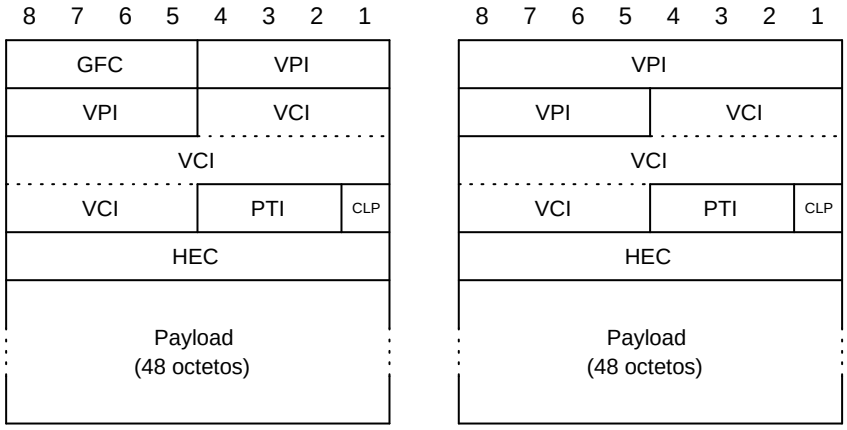
  Bit que define a prioridade da célula quanto ao descarte. Células com valor CLP = 1 são preferencialmente descartadas caso haja necessidade. Este artifício é útil para o controle de tráfego e de congestionamento e para a manutenção de um bom grau de utilização da rede.

- HEC (*Header Error Check*)

  Campo que contém informações para a detecção de erros de bit no cabeçalho de uma célula. Uma das formas de realizar esta tarefa é a utilização do mecanismo CRC (*Cyclic Redundancy Check*). Este é o único campo do cabeçalho da célula que não é controlado (calculado e checado) na camada ATM.

- GFC (*Generic Flow Control*)

  Este campo está presente apenas em células das interfaces UNI (*User-Network Interface*). Nas interfaces NNI (*Network Node Interface*), os 4 bits referentes a este campo são incorporados ao campo VPI, que passa de 8 bits para 12 bits.



(a) Célula UNI (b) Célula NNI

Figura 1.7: Estrutura da Célula ATM

A Figura 1.7 mostra a estrutura de células na UNI (a) e na NNI (b).

1.5 Conexões ATM

Em uma rede ATM, conexões podem ser predimensionadas, criadas através de funções de gerenciamento ou criadas através de negociação por sinalização (*signaling*). No primeiro caso, trata-se de Canais Virtuais Permanentes (PVC - *Permanent Virtual Channel*), enquanto o uso de sinalização gera Canais Virtuais Comutados (SVC - *Switched Virtual Channel*). As conexões suportadas por estes canais propiciam a comunicação entre sistemas que fazem parte da rede.

Para o estabelecimento de uma conexão entre sistemas da rede, os parâmetros que identificam e dimensionam o tráfego da aplicação, bem como os requisitos de Qualidade de Serviço (QoS) são comunicados ao controle de rede. Esta conexão só será aceita se houver, ao longo de todo o percurso da conexão, recursos suficientes para que a QoS desejada seja mantida, tanto da nova conexão quanto das já em operação. Portanto, a conexão só será liberada se a rede conseguir arcar com as necessidades desta nova conexão, sem, entretanto, comprometer o funcionamento das demais. Tão logo a rede tenha decidido acerca da aceitação da nova conexão, valores de Identificador de Caminho Virtual (VPI - *Virtual Path Identifier*) e de Identificador de

Canal Virtual (VCI - *Virtual Channel Identifier*) são assinalados e utilizados como identificação da conexão. Entretanto, estes valores têm valor apenas em nível da interface local. Os valores dos pares VPI/VPC nas duas extremidades de uma conexão ponto-a-ponto geralmente são distintos, assim como nos nós intermediários [47].

Conexões ATM se classificam em dois níveis hierárquicos: VPC (*Virtual Path Connection*) e VCC (*Virtual Channel Connection*). Uma VCC suporta um fluxo simples de células (em duas direções). A VPC representa um grupo de VCC's [68].

Uma conexão é formada pela concatenação de enlaces físicos. Desta forma, uma VCC é formada pela concatenação de VCL's (*Virtual Channel Links*), enquanto um VPC é formado por uma concatenação de VPL's (*Virtual Path Links*). A Figura 1.8 ilustra a disposição de Links Físicos, VPL's e VCL's.

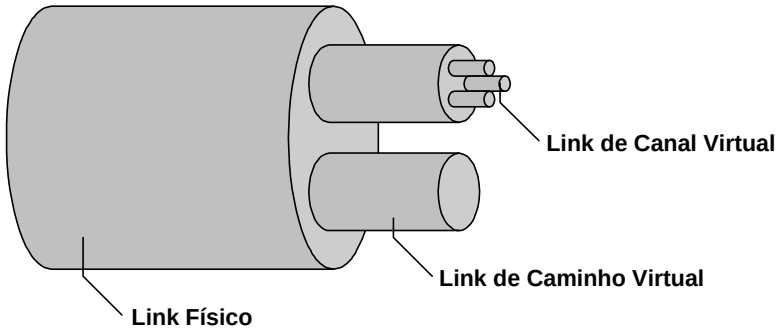


Figura 1.8: Estrutura Hierárquica de Caminhos e Canais Virtuais

Cada VPC é identificada por seu valor de PCI, enquanto um VCC é identificado não só por seu valor de VCI, mas também pelo valor de PCI que identifica o VPC a qual pertence.

O roteamento e a comutação das células de uma conexão são realizados através de seis valores VPI/VCI. Através destes valores, um elemento comutador determina o enlace de saída das células da conexão que utiliza este nó como parte de sua rota. Esta determinação é feita através da identificação do enlace de entrada e do par de valores VPI/VCI. Uma característica importante da comutação ATM é que os valores de VPI/VCI geralmente mudam a cada nó intermediário. Assim, a cada nó comutador, o valor do par VPI/VCI da célula de chegada, bem como a identificação da porta de entrada são utilizados para a determinação da porta de saída

Conexão	Link Entrada	VPI Antigo	VCI Antigo	Link Saída	VPI Novo	VCI Novo
1	1	15	86	4	25	86
2	1	15	84	4	25	84
3	1	11	86	4	12	86
4	2	17	95	3	25	90
5	2	15	84	4	35	84

Tabela 1.2: Tabela de Rotas de Comutação ATM

e dos novos valores de VPI/VCI da célula de saída. Esta determinação é feita através de uma tabela que recebe informações de conexões e suas rotas. Esta tabela é atualizada a cada conexão que é inicializada ou que é finalizada. Depois de definidos, através da tabela de rotas, os novos valores VPI/VCI e o enlace de saída correspondente, células da conexão são submetidas ao link de saída, com seu par VPI/VCI substituído pelos novos valores.

Um exemplo de comutação de células é ilustrado na Figura 1.9 [59].

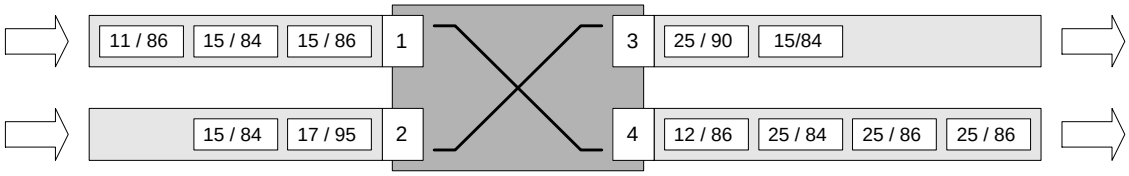


Figura 1.9: Exemplo de Comutação de VPC/VCC

Neste exemplo, considera-se um comutador com dois enlaces de entrada (1 e 2) e dois enlaces de saída (3 e 4), onde 5 conexões são comutadas. Por exemplo, células que atingem o elemento de comutação através do enlace 1 e que têm par VPI/PCI igual a 11/86 pertencem à conexão 3. Portanto, consultando-se a tabela, verifica-se que os novos valores VPI/VCI de cada célula da conexão 3 devem ser 4/12, e que o enlace de saída para estas células é o 4. Portanto, para cada célula da conexão 3, seus valores VPI/VCI são modificados pelo comutador para 4/12 e são repassados para o enlace 4.

1.6 O Modelo de Referência ATM

O Modelo de Referência das redes ATM, padronizado pelo ITU-T [34], apresenta inovações com relação aos modelos das redes OSI e TCP/IP. Estas tecnologias apresentam estrutura em camadas sobre duas dimensões, enquanto o ATM se apresenta em três dimensões, acrescentando um plano específico para gerência. Nesse caso, o gerenciamento é realizado tanto em nível de camadas quanto a nível global. O Modelo de Referência para as redes ATM é ilustrado na Figura 1.10.

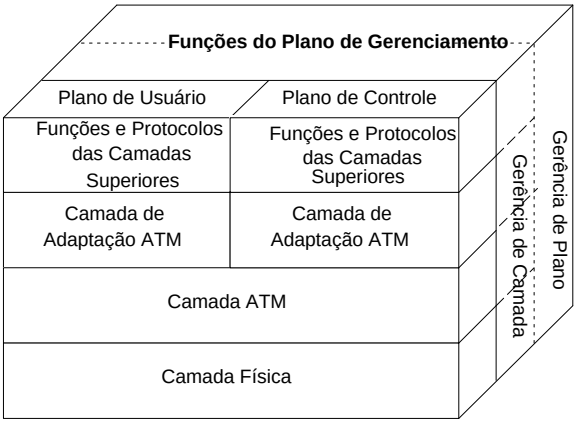


Figura 1.10: Modelo de Referência das Redes ATM

O Modelo de Referência ATM se assemelha ao Modelo de Referência das RDSI (mostrado na Figura 1.3). A principal diferença entre elas é que o modelo ATM já possui um nível de detalhamento maior, tendo, por exemplo, a subdivisão das Funções de Gerenciamento em Gerenciamento de Plano e Gerenciamento de Camada.

Três planos são definidos no modelo de referência ATM:

- Plano de Usuário

  Este plano propicia a transferência de informações de usuário pela rede. Mecanismos de controle associados às conexões de usuários também se utilizam deste plano. O segmento de camada ATM do Plano de Usuário é responsável pelas funções de comutação das células de usuário e controle de fluxo e de erro. A secção da camada AAL referente a este plano varia de acordo com o tipo de aplicação de usuário.

- Plano de Controle

  O Plano de Controle é responsável pela sinalização (*signaling*) [37] necessária para o estabelecimento, manutenção e finalização de chamadas e conexões. Considera-se uma *chamada* como sendo uma conexão ou conjuntos de conexões entre dois ou mais usuários [67].

- Plano de Gerenciamento

  O Plano de Gerenciamento é responsável por várias funções, dentre elas o gerenciamento de camadas e o tratamento dos fluxos de informações de operação, administração e manutenção (OAM). As funções OAM, assim como seus fluxos de informação, estão descritos na recomendação I.610 da ITU-T [35].

A interação e as responsabilidades de cada plano da configuração de referência das redes ATM estão ilustradas na Figura 1.11 [19].

Uma rede ATM como descrita na Figura 1.6 é composta por nós intermediários e sistemas finais. Nos nós intermediários, geralmente a pilha de protocolos ATM se limita a duas camadas (Física e ATM). Isto porque os requisitos primordiais para a função deste nó – endereçamento, roteamento e sinalização – são supridos em nível de camada ATM. Portanto, geralmente não há necessidade de outros níveis como AAL e Superior. A Figura 1.12 mostra o caminho de uma conexão ATM ao longo das camadas dos nós intermediários e sistemas finais.

Cada camada explicitada na Figura 1.10 apresenta suas funções específicas. A estrutura e as funções das camadas do Modelo de Referência ATM serão descritas nas subseções a seguir.

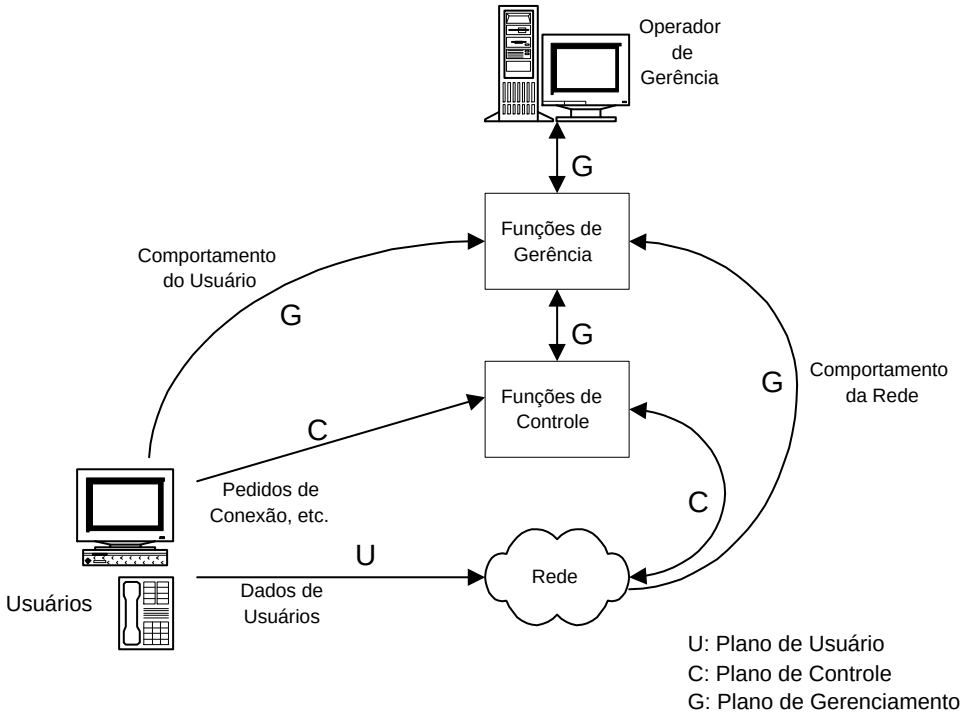


Figura 1.11: Interação entre Planos da Configuração de Referência ATM

1.6.1 Camada Física

A Camada Física tem como função transportar células entre Camadas ATM adjacentes. A implementação da Camada Física deve ser específica para o meio de comunicação utilizado (*e.g.* coaxial, fibra óptica, etc) e ao mesmo tempo deve oferecer um serviço unificado à camada ATM, mantendo-a abstraída dos detalhes do meio de comunicação utilizado, como temporização de bits, por exemplo.

A Camada Física é subdividida em: Subcamada de Convergência de Transmissão (TC - *Transmission Convergence*) e Subcamada de Meio Físico (PMD - *Physical Media Dependent*), sendo que a primeira é comum em todos os tipos de mídia física, enquanto a última é específica para cada tipo de meio físico. Algumas funções de cada subcamada são relacionadas na Tabela 1.3, enquanto as principais interfaces de Camada Física ATM, com suas respectivas características, são relacionadas na Tabela 1.4.

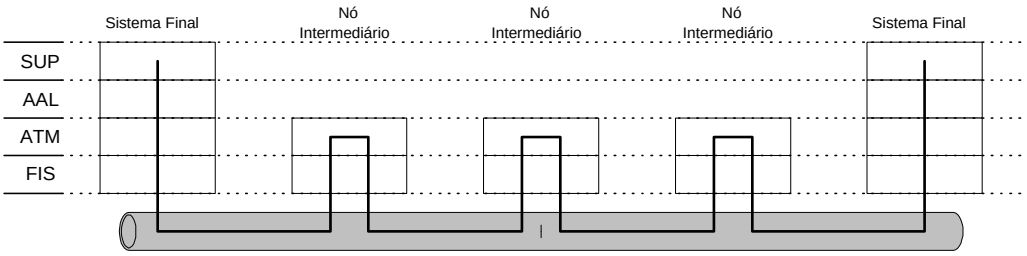


Figura 1.12: Conexão ATM fim-a-fim

Subcamada de Convergência	
	Geração e chegagem do campo HEC
	Delineação de células
	Manutenção de taxa de células
Subcamada de Meio Físico	
	Codificação de transmissão
	Temporização e sincronização
	Transmissão (elétrica/óptica)

Tabela 1.3: Funções da Camada Física

Subcamada de Convergência de Transmissão

A Subcamada de Convergência de Transmissão tem como funções primordiais montar os conteúdos das unidades de transmissão do meio físico e preparar as informações de protocolo necessárias para a transmissão destes dados. Outras funções desempenhadas por esta subcamada incluem:

- Geração e Verificação de campo HEC

  O campo HEC de cada célula é gerado pela subcamada TC. Este campo é utilizado pela própria camada a fim de detectar algum possível erro de composição no ponto de recepção.

- Geração e Recuperação de Quadros de Transmissão

  Geralmente, a transmissão no meio físico é composta por unidades chamadas quadros (*frames*). A subcamada TC é responsável por gerar quadros a partir de células recebidas da Camada ATM e por reconstruir células a partir dos quadros recebidos do meio físico.

	Taxa Bruta (Mbps)	*Throughput* (Mbps)	Sistema	Mídia	Campus/ WAN
DS-1 (T-1)	1,544	1,536	PDH	Coaxial	Ambos
E-1	2,048	1,920	PDH	Coaxial	Ambos
DS-3 (T-3)	44,736	40,704	PDH	Coaxial	WAN
E-3	34,368	33,984	PDH	Coaxial	WAN
E-4	139,264	138,240	PDH	Coaxial	WAN
SDH STM-1 / SONET STS-3c	155,520	149,760	SDH	F. óptica monomodo	WAN
SDH STM-4c / SONET STS-12c	622,080	599,040	SDH	F. óptica monomodo	WAN
FDDI-PMD	100,000	100,000	Block coded	STP / F. óptica multimodo	Campus
STS 3-C	155,520	149,760	SONET	UTP-5	Campus

Tabela 1.4: Interfaces de Camada Física ATM

- Delineação de Células

  Em mais baixo nível, quadros são gerados e enviados a uma taxa constante correspondente à largura de banda disponível no enlace físico. Entretanto, a demanda por largura de banda por parte das aplicações pode não chegar a completar os quadros disponíveis para transmissão. Portanto, a subcamada TC é responsável por inserir nestes quadros células vazias para que a taxa de transmissão de quadros permaneça constante.

Subcamada de Meio Físico

A subcamada de Meio Físico desempenha funções que são intrínsecas ao meio físico em operação (coaxial, par trançado, fibra óptica, etc). Esta subcamada se aproxima em comportamento às camadas físicas das redes clássicas. A função básica desta subcamada é receber da subcamada TC bits e transparentemente transportá-los através de um enlace físico, e, por outro lado, receber sinais (ópticos ou elétricos) do meio físico e convertê-los em bits, que serão repassados à subcamada TC.

Outra função importante desta subcamada é a codificação e decodificação de blocos de bits. Este artifício é importante para a detecção dos limites dos bits no meio físico.

1.6.2 Camada ATM

A camada ATM é responsável pela transmissão de células entre entidades de camada ATM. Esta camada utiliza os serviços oferecidos pela Camada Física de modo a ter acesso à rede física. Em um ponto de transmissão, a carga de uma célula é recebida do usuário pela camada ATM, que inclui 4 octetos do cabeçalho (excluindo o campo HEC) e repassa os dados para a camada física, que deverá calcular o campo HEC e transmitir a célula através do meio físico. No ponto de destino, a célula é recebida, o cabeçalho é extraído e a carga é passada para o usuário destino [59]. Células que apresentam erro no cabeçalho são descartadas ao longo dos nós intermediários. A camada ATM não é responsável por detectar erros em cargas de células, sendo esta tarefa responsabilidade dos nós finais.

A camada ATM suporta conexões com diferentes características de Qualidade de Serviço (QoS - *Quality of Service*). Isto requer que células de diferentes conexões sejam tratadas pelo comutador de forma distinta. A camada ATM é responsável por criar mecanismos que efetivem esta distinção, como reserva de *buffer* para classes especiais de conexões, controle de prioridade para ocupação dos *buffers*, etc [68].

Sinalização

A *sinalização* é o mecanismo automatizado pelo qual conexões de rede podem ser estabelecidas sob demanda e posteriormente eliminadas. Um processo de sinalização bem sucedido faz com que a rede encontre um caminho entre fonte e destino, ajuste as tabelas de comutação ao longo deste caminho e reserve todos os recursos necessários para a operação normal da conexão, como espaço em *buffer* e largura de banda. Após este processo, a conexão estará pronta para iniciar sua operação. Quando uma conexão não é mais necessária, a sinalização é novamente ativada com o intuito de fechar esta conexão, liberando os recursos alocados e atualizando tabelas de comutação.

O ATM Forum definiu, como parte de sua especificação UNI 3.1 [8], um mecanismo de sinalização baseado nas definições do documento Q.2931 [37] da ITU-T. O estabelecimento de uma conexão através de uma UNI segue o padrão ilustrado na Figura 1.13.

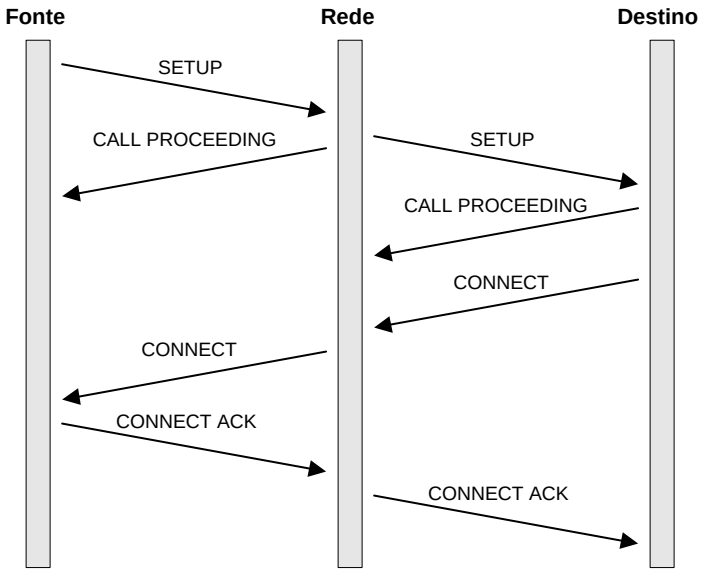


Figura 1.13: Sinalização para Estabelecimento de Conexão

O usuário envia uma mensagem de SETUP para a rede. Esta mensagem deve incluir todas as informações necessárias para identificar a fonte, como classe de serviço, os parâmetros de descritor de tráfego necessários e o endereço do ponto destino. A rede envia ao usuário uma mensagem de CALL PROCEEDING, que indica que o pedido está sendo processado, e envia uma mensagem de SETUP para o ponto de destino. Para encontrar um caminho entre fonte e destino, um sistema de roteamento é necessário. Esta função é parte do protocolo das interfaces do tipo NNI (*Network Node Interface*). O ponto de destino pode aceitar ou rejeitar a conexão solicitada. Se for o caso de aceitar, uma mensagem de CONNECT é enviada em direção à fonte, caso contrário, a mensagem RELEASE é enviada à fonte. Quando a conexão não é mais necessária, sua liberação é realizada através do envio da mensagem RELEASE. Após a fonte receber a mensagem CONNECT confirmando a disponibilidade do destino em estabelecer a conexão, uma mensagem CONNECT ACK é enviada ao ponto de destino com o intuito de notificar o recebimento da mensagem CONNECT.

A decisão acerca da falha do estabelecimento de uma conexão depende de um conjunto de valores de *time-out*. Por exemplo, após a fonte da conexão enviar a mensagem SETUP, deve esperar por um determinado intervalo de tempo até que a mensagem CONNECT seja recebida.

Se este intervalo de tempo superar o valor de *time-out* desta operação, o pedido de conexão é considerado falho.

A transmissão de células com informações de usuários na camada ATM é considerada um serviço não confiável. Como o processo de sinalização é muito sensível a perda de informações, um serviço confiável de mais baixo nível exclusivo para sinalização é oferecido com o intuito de evitar esta perda. O serviço confiável para o processo de sinalização consiste em um tipo específico de camada AAL sobre as mesmas camadas ATM e Física utilizadas pelos dados de usuários. Esta AAL é chamada "*Signalling AAL*" (SAAL) [36] e oferece um serviço confiável de transporte de informações de sinalização através do *Service Specific Connection Oriented Protocol* (SSCOP). Este protocolo utiliza os mesmos serviços não confiáveis em nível de camada ATM, mas acrescenta mecanismos de *time-out* e retransmissão em nível de SAAL [68].

Endereçamento

Apesar do tráfego de células ATM ser direcionado por tabelas de comutação nos nós intermediários que fazem parte da rota da conexão, é necessário um esquema de endereçamento que seja capaz de identificar biunivocamente um sistema final para fins de sinalização.

Os endereços ATM são formados por 20 octetos e são estruturados de forma que cada parte é utilizada num nível de roteamento. Existem basicamente 3 formatos de endereços para o ATM. Sao eles: ITU-T E.164 [39], IEEE 802 DCC e OSI ICD. Estes formatos estão ilustrados na Figura 1.14.

O ATM Forum definiu que comutadores ATM de redes privadas devem suportar todos os três formatos de endereços. Não obstante, os comutadores de rede pública devem suportar pelo menos os endereços do formato ITU-T E.164.

Roteamento

O estabelecimento de uma conexão necessita que uma rota seja traçada através da rede entre fonte e destino. Se ambos os sistemas finais envolvidos na conexão estiverem ligadas ao mesmo equipamento comutador, a função de roteamento é desnecessária. Como as rotas são compostas

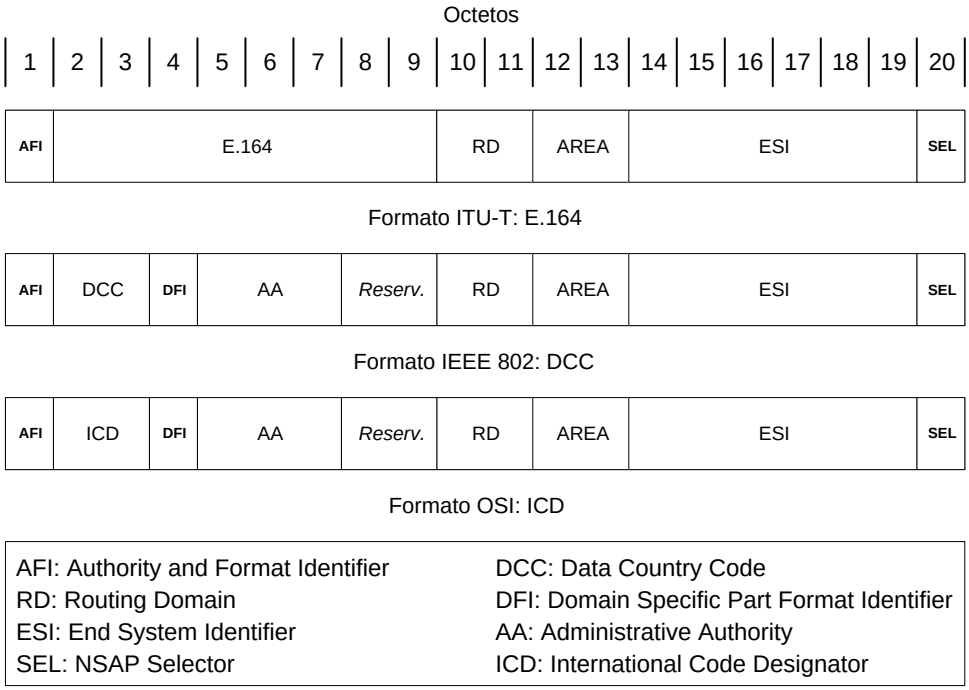


Figura 1.14: Formatos de Endereçamento ATM

por sistemas intermediários interligados através de enlaces de dados, a função de roteamento deve ser primordialmente responsabilidade do protocolo das interfaces NNI.

A Figura 1.15 mostra uma situação de topologia onde há diferentes rotas entre os sistemas finais A e B para o estabelecimento de uma conexão. Após o sistema final A iniciar a requisição do estabelecimento de uma conexão até o sistema final B, o comutador A, ao qual o sistema final A está diretamente ligado, descobre e seleciona uma das rotas possíveis até o comutador B, ao qual está conectado o sistema final B, destino da conexão [68].

A escolha da rota deve ser feita de tal forma que o custo da transmissão de células ao longo dessa rota seja minimizado. Entre os fatores que influenciam na definição do custo de cada rota estão o número de pontos intermediários (*hops*) e as capacidades dos enlaces que formam a rede e a quantidade de recursos disponível.

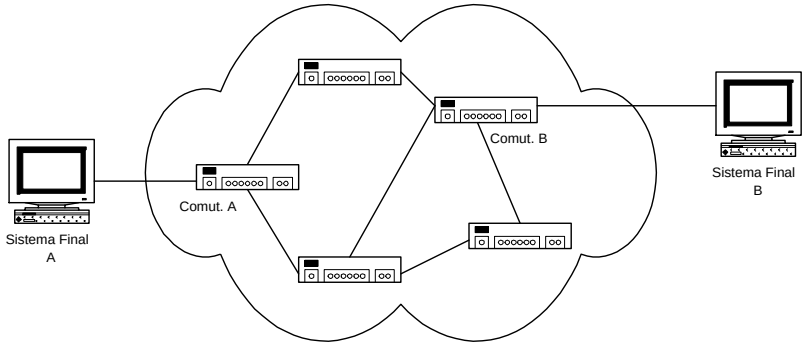


Figura 1.15: Exemplo de Roteamento entre Interfaces ATM

1.6.3 Camada de Adaptação (AAL)

A camada AAL (*ATM Adaptation Layer*) se encontra entre as camadas ATM e Superior. A principal função desta camada é adaptar PDUs (*Protocol Data Units*) de serviços da camada Superior ao formato de cargas de células. Esta adaptação é feita sobre propriedades como tamanho de pacote, probabilidade de perda e relação de tempo [68]. A adaptação do serviço *Frame-Relay* [31] é um exemplo de tarefa desempenhada pela camada AAL. Neste caso, os pacotes do *Frame-Relay* são quebrados em cargas de células.

A camada ATM é caracterizada por não oferecer garantias de integridade de cargas de células, assim como não oferece mecanismos de detecção de perda ou desordenação de células. Além disto, a camada ATM também não oferece maneiras de determinar a variação do atraso nas células. O principal motivo porque a camada ATM não deve se encarregar destas tarefas é que nem todos os tipos de aplicação necessitam de todos estes requisitos. Por exemplo, uma aplicação típica de transferência de dados é extremamente sensível a perda de informação e razoavelmente indiferente ao atraso de transmissão, enquanto uma aplicação de telefonia não é tão sensível a perda de informações, enquanto o controle do atraso é tarefa imprescindível. Portanto, estas pendências operacionais devem ser compensadas pela camada AAL, de modo fim-a-fim.

Como a variedade de aplicações existentes exige tratamento diferenciado do tráfego de acordo com os seus requisitos, classes de aplicações são definidas em nível da camada AAL.

Esta classificação é definida de acordo com os requisitos das aplicações. Como a variedade de aplicações é potencialmente muito grande, a classificação é feita de forma geral, de acordo com os seguintes parâmetros:

- Sincronismo entre fonte e destino: Necessário ou Desnecessário;
- Taxa de Transmissão: Constante ou Variável;
- Modo de Conexão: Orientado a conexão ou Não orientado a conexão.

As classes de aplicações, definidas pela ITU-T em [32], são descritos na Tabela 1.5.

	Classe A	Classe B	Classe C	Classe D
Sincronismo	Necessário		Desnecessário	
Orientação a Conexão	Sim			Não
Taxa de Transmissão	Constante	Variável		
Protocolo AAL	Tipo 1	Tipo 2	Tipo 3/4 Tipo 5	Tipo 3/4

Tabela 1.5: Classes de Serviços da AAL

Adicionalmente a estas 4 classes, duas outras foram definidas pela ATM Forum [6]:

- Classe X

  Nesta classe, o tipo de AAL, o tipo de tráfego (constante ou variável) e os requisitos cronológicos são definidos pelo usuário.

- Classe Y

  Esta classe é definida para aplicações caracterizadas por requisitos que podem mudar após o estabelecimento da conexão. Desta forma, os requisitos solicitados pelo usuário são completamente atendidos pelo sistema a princípio. Posteriormente, estes requisitos são re-adaptados de acordo com a disponibilidade de recursos.

A camada AAL é organizada em duas subcamadas lógicas: Subcamada de Convergência (CS - *Convergence Sublayer*) e Subcamada de Segmentação e Remontagem (SAR - *Segmentation and Reassembly Sublayer*). A Subcamada de Convergência fornece as funções necessárias para o suporte de aplicações específicas usando a camada AAL. Cada usuário se comunica com a AAL através de seu Ponto de Acesso a Serviços (SAP - *Service Access Point*), que simplesmente representa o endereço da aplicação. Portanto, esta subcamada é dependente de serviço.

A subcamada SAR é responsável por receber informações da subcamada CS e segmentá-las em cargas de células e, na outra extremidade da conexão, remontar as unidades de dados a partir de múltiplas células que chegam da camada ATM.

Para suportar as diferentes classes básicas de serviço, um conjunto de protocolos em nível de camada AAL foi definido. Para cada classe A, B, C e D foi, então, criado um tipo de AAL, sendo esses denominados por Tipos 1 a 4. Posteriormente, os tipos 3 e 4 se fundiram, formando a AAL Tipo 3/4, e uma nova variedade, AAL Tipo 5, foi criada [69].

AAL Tipo 1

Este tipo suporta aplicações de taxa constante. Portanto, a única tarefa da subcamada SAR é montar e desmontar as cargas de células de acordo com a unidade de informação da aplicação. Em cada unidade de dados, um campo é incluído com informações que possibilitem detectar e talvez corrigir erros de bit devidos à transmissão.

Em nível de subcamada CS, nenhum tipo especial de PDU foi definido. Portanto, as funções desempenhadas por esta subcamada se restringem ao controle cronológico e de sincronização.

AAL Tipo 2

Este tipo de AAL é destinado a aplicações como vídeo e áudio, que demandem controle cronológico, mas não apresentem taxa constante.

AAL Tipo 3/4

Na especificação inicial, as PDU's dos Tipos 3 e 4 de camada AAL eram muito semelhantes. Por isto, foi decidido fundir estes dois protocolos em um só.

Os serviços suportados pela AAL Tipo 3/4 podem ser caracterizados em dois parâmetros:

- Orientação da Conexão:

  O serviço pode ser orientado a conexão ou não orientado a conexão (*conectionless*). No primeiro caso, é possível definir múltiplas conexões lógicas em nível de SAR para uma única conexão ATM, definindo múltiplos fluxos virtuais. No último caso, cada bloco de dados apresentado à subcamada SAR é tratado independentemente.

- Modo de Serviço:

  O serviço pode ser baseado em modo de mensagem ou em modo de seqüência (*stream*). O primeiro modo é baseado em quadros. Portanto, os serviços e protocolos de redes baseados em OSI, como LAPD e *Frame Relay*, estariam mais adequados a este tipo de serviço. O segundo modo é baseado em pequenos blocos de dados de tamanho fixo, que podem chegar a ter apenas um octeto. Neste caso, cada bloco é transportado em uma célula. Serviços deste tipo são caracterizados por um fluxo de dados contínuo e de baixa velocidade, com requisitos de baixo atraso de transmissão de células.

AAL Tipo 5

O Tipo 5 de AAL foi desenvolvido com o intuito de minimizar o *overhead* em nível dessa camada. Conseqüentemente, muitas tarefas que eram designadas à camada AAL se tornaram responsabilidade das aplicações na camada Superior. Outro ponto modificado foi a redução do *overhead* de transmissão, através da eliminação de campos desnecessários em sua PDU. Por exemplo, as aplicações na camada Superior são requisitadas a se responsabilizarem pelo gerenciamento da conexão, enquanto é exigido da camada ATM que a taxa de erro seja baixa. Assim, com a primeira medida reduz-se o *overhead* de processamento, enquanto a última medida reduz o *overhead* de transmissão.

Este tipo de AAL está sendo amplamente popularizado, principalmente para aplicações em LAN sobre ATM.

1.6.4 Camada Superior

Esta camada representa as diferentes aplicações que podem ser sobrepostas à Camada de Adaptação ATM. Esta camada não é objeto de padronização pela ITU-T.

1.7 LAN Emulation

LAN Emulation (LANE) é um serviço desenvolvido no ATM Forum [2] para permitir que aplicações LAN já existentes executem sobre uma rede ATM. Para isso, esse serviço deve emular as características e o comportamento de redes locais convencionais. Assim, um serviço não orientado à conexão deve ser suportado. Tráfego *broadcast* e *multicast* também devem ser permitidos. A LANE deve possibilitar a interconexão de LANs tradicionais com LANs emuladas, mantendo o endereço MAC (*Media Access Control*) de cada dispositivo da LAN, para deste modo preservar a imensa base de aplicações LAN existentes, permitindo que funcionem sem alterações sobre uma rede ATM. Esta é a maior vantagem da LANE e sua maior desvantagem, pois as aplicações não utilizam características de garantia de QoS do ATM.

A LAN emulada pode ser *Ethernet* (IEEE 802.3) ou *Token Ring* (IEEE 802.5). A LANE define para cada instância de LAN emulada um conjunto de serviços: configuração, resolução de endereços (MAC para ATM) e *broadcast*.

A participação numa LAN emulada não é determinada pela localização física do dispositivo, mas pela associação lógica com o conjunto de serviços. Por isso, LANE é ideal para a construção das chamadas *LANs Virtuais* (VLANs) [2].

Numa LAN emulada, também é possível o acesso direto à rede ATM, permitindo que duas estações estabeleçam um caminho virtual e tirem proveito da largura de banda e das garantias de QoS oferecidas pelo ATM. Contudo, a maioria das aplicações devem executar utilizando os serviços da LANE, sem garantias de QoS [50].

1.7.1 Arquitetura

A Figura 1.16 mostra as camadas funcionais da arquitetura da LANE. As camadas Física e ATM são as usuais. LANE usa serviços AAL tradicionais para as transferências de dados de usuário (AAL 5) e de sinalização (SAAL). A camada LANE apresenta para a camada superior uma interface em nível de MAC e um esquema de endereçamento, com o objetivo de parecer às aplicações estarem, para todos os efeitos, executando realmente sobre uma LAN.

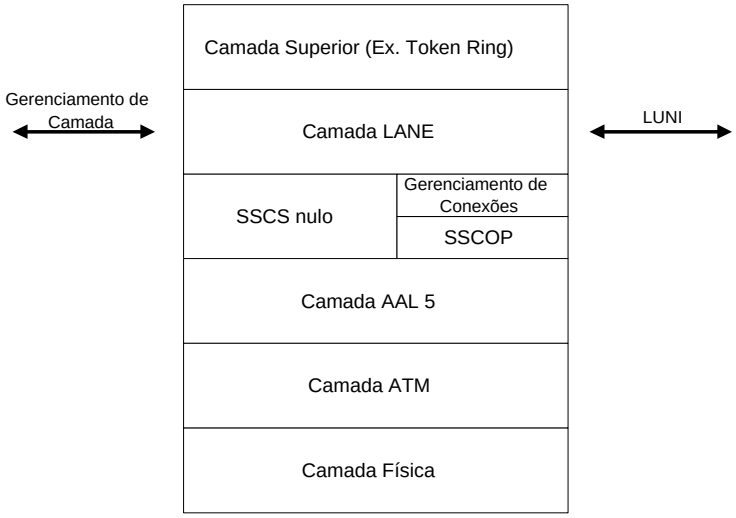


Figura 1.16: Estrutura da LANE

Clientes e servidores LANE interagem sobre a LUNI (*LANE User-Network Interface*) [4]. As funções executadas sobre a LUNI são: inicialização, transferências de dados, registro e resolução de endereços.

1.7.2 Componentes

Os componentes de um LAN emulada são assim definidos:

- LEC (*LAN Emulation Client*)

  Um LEC pode ser uma estação, uma ponte ou um roteador. O LEC faz resolução de endereços, repasse de dados para outros LECs e outras funções de controle [5]. O LEC

apresenta uma interface de nível MAC e implementa numa LUNI comunicação com outros componentes de uma LAN emulada.

- LES (*LAN Emulation Server*)

  O LES age como servidor de registro e resolução de endereços, sendo dedicado a uma LAN emulada. O LES oferece meios para que LECs registrem seus endereços MAC e ATM. Um LEC pode pedir ao LES a resolução de um endereço MAC para ATM [3]. O LES pode responder diretamente esse pedido ou repassá-lo para outros LECs.

- LECS (*LAN Emulation Configuration Server*)

  O LECS é usado para inicializar um LEC com informações específicas da LAN emulada para qual o LEC está entrando. Por exemplo, o LECS fornece o endereço ATM do LES para o LEC (de acordo com seu endereço MAC ou ATM).

- BUS (*Broadcast and Unknown Server*)

  O BUS trata quadros para o endereço de broadcast MAC. Todo tráfego multicast e unicast é enviado por um LEC antes que o endereço ATM do destino seja resolvido. Todos os LECs mantêm uma conexão com o BUS e são destinos de um VCC ponto-a-multiponto que tem o BUS como origem. Isso permite que LECs enviem quadros sem estabelecer uma conexão primeiro.

Os servidores (LES, LECS e BUS) podem executar numa única máquina ou em máquinas separadas. Uma LAN emulada só precisa de um LES e de um BUS, mas que só podem executar um tipo de LANE (*Ethernet* ou *Token Ring*).

1.7.3 Conexões

Mensagens de controle e quadros de dados trafegam em VCCs diferentes. VCCs de Controle levam mensagens de um LEC para outro LEC, para um LES ou um LECS. Já os VCCs de Dados são estabelecidos entre LECs ou de um LEC para o BUS.

- VCCs de Controle:
  - VCC de Configuração Direta

    VCC bidirecional ponto-a-ponto entre o LEC e o LECS. É usado pelo LEC para obter informações de configuração, como o endereço ATM do LES.
  - VCC de Controle Direto

    VCC bidirecional ponto-a-ponto entre o LEC e o LES para o envio de informações de controle.
  - VCC de Controle Distribuído

    VCC unidirecional ponto-a-ponto ou ponto-a-multiponto que é opcionalmente estabelecido do LEC para um ou mais LECs, durante a fase de inicialização.
- VCCs de Dados:
  - VCC de Dados Direto

    VCC bidirecional ponto-a-ponto estabelecido entre dois LECs. Quando um LEC (origem) deseja se comunicar com outro (destino), mas não sabe seu endereço ATM, o LEC origem faz um pedido de resolução de endereços para o LES que é mandado pelo VCC de Configuração Direta. Assim que a resposta é recebida, o LEC origem pode estabelecer um VCC de Dados Direto com o LEC destino.
  - VCC de Envio Multicast

    VCC bidirecional ponto-a-ponto estabelecido entre o LEC e o BUS para o envio de quadros de dados multicast e unicast com endereço ATM do destino desconhecido pelo LEC. Um LEC pode receber quadros de dados sobre esse VCC.
  - VCC de Repasse Multicast

    Pode ser um VCC unidirecional ponto-a-ponto ou ponto-a-multiponto que é estabelecido do BUS para um ou mais LECs. Esse VCC é usado para o repasse de quadros de dados multicast para os membros da LAN emulada.

1.8 IP sobre ATM

Visando permitir o suporte direto do IP sobre a AAL/5, o IETF especificou um mecanismo conhecido como IP sobre ATM, definido na RFC 1577 [42]. A imensa base de estações utilizando IP deve ter meios de migração para o ATM.

A política de dividir as redes em subredes de acordo com domínios administrativos e de trabalho é utilizada nesta abordagem. Assim, uma rede ATM pode ser vista como várias LIS (*Logical IP Subnetwork*). Uma LIS é um conjunto de estações e roteadores ATM conectados, dentro de uma subrede IP comum. A Figura 1.17 [48] mostra uma rede ATM dividida em LIS.

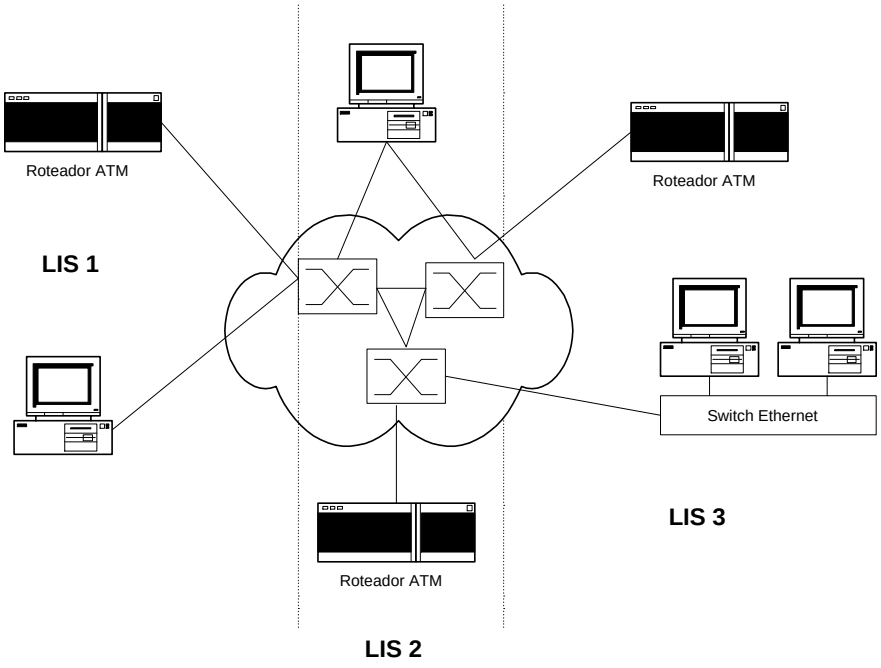


Figura 1.17: IP Sobre ATM

Cada LIS contém um servidor de ARP (*Address Resolution Protocol*), responsável pelo mapeamento de endereços IP em ATM e vice-versa. Todos os membros da LIS são registrados com o servidor de ARP. Todos os pedidos de resolução de endereços a partir de membros da LIS são tratados pelo servidor, que pode ser um processo executando no roteador.

Os pedidos de resolução de endereços IP são passados para o servidor de ARP. Quando a

estação que originou o pedido recebe o endereço ATM correspondente, ela pode estabelecer uma conexão direta com esta máquina. Duas grandes mudanças foram realizadas no protocolo tradicional ARP: a criação da mensagem ATMARP (para pedir resolução de endereços) e da mensagem InATMARP (para registro inverso de endereços) [50].

No IP sobre ATM, cada estação deve ser configurada manualmente com o endereço do servidor de ARP, diferentemente da LANE onde há descoberta dinâmica. Uma desvantagem é a possibilidade que duas estações na mesma rede ATM física, tenham que se conectar através de um roteador por estarem em diferentes LIS. Esta restrição pode diminuir o *throughput* da rede e aumentar o retardo. Uma vantagem do modelo IP sobre ATM é o tamanho da MTU (*Maximum Transfer Unit*) ser de 9180 bytes. Uma MTU grande pode aumentar o desempenho de estações ligadas a uma rede ATM [48].

Capítulo 2

Redes Neurais Artificiais

"Let schoolmasters puzzle their brain,
with grammar, and nonsense, and learning.
Good liquor, I stoutly maintain,
gives genius a better discerning."
– Oliver Goldsmith

2.1 Introdução

As Redes Neurais Artificiais (ou Redes Neurais) são técnicas computacionais que têm a possibilidade de processar e extrair conhecimento através de experiência. O ponto de partida para o projeto das Redes Neurais é o cérebro dos animais. Pesquisadores vêm tentando encontrar uma representação matemática que seja capaz de simular o funcionamento deste órgão em um sistema de processamento de informações altamente complexo, não-linear e paralelo [20]. O cérebro é capaz de realizar tarefas muito complexas (como reconhecimento de padrões [52, 11, 76], percepção e controle motor) muito mais rapidamente e com mais eficácia do que o melhor computador digital existente atualmente. Acredita-se que todo este poder se deve à capacidade

do cérebro de auto-organizar seus neurônios e ligações sinápticas.

Pouco se sabe ainda sobre o real funcionamento do cérebro. Não se conhece bem a forma com que a experiência e o conhecimento são armazenados, bem como a forma como os neurônios interagem entre si. Acredita-se que a aquisição de conhecimento por parte do cérebro é realizada através da criação de novas conexões entre neurônios ou através de mudanças nas voltagens (intensidades) das ligações já existentes. Um outro ponto a se considerar é que estas mudanças são realizadas a uma grande velocidade e podendo envolver milhões de neurônios ao mesmo tempo [71].

A rede formada entre neurônios do cérebro é bastante complexa. O córtex humano, por exemplo, é composto por mais ou menos 10 milhões de neurônios e mais de 60 trilhões de sinapses ou conexões [66].

As Redes Neurais Artificiais representam, então, uma tentativa de simular o funcionamento do cérebro, com a finalidade de herdar parte da funcionalidade e do poder de realizar funções específicas que caracteriza este órgão.

Redes Neurais são usualmente implementadas através de componentes eletrônicos ou através de simulações em software.

Uma rede neural é composta por unidades atômicas de processamento denominadas neurônios artificiais, que se interconectam através de ligações sinápticas (ou sinapses). Estas ligações são caracterizadas por suas intensidades (pesos). O processo pelo qual uma rede neural adquire seu conhecimento é denominado *Treinamento*.

Segue, então, uma definição mais concisa de uma rede neural, segundo [20]:

Rede Neural Uma rede neural é um processador extremamente paralelo e distribuído que tem propensão natural de armazenar conhecimento experimental, deixando-o disponível para utilização. Esta rede remonta o cérebro em dois aspectos:

1. O conhecimento é adquirido pela rede através de treinamento;
2. As intensidades das sinapses entre neurônios artificiais são utilizadas para armazenar conhecimento.

2.2 Breve Histórico

Acredita-se que o ponto de partida para as redes neurais é o trabalho dos pesquisadores McCulloch e Pitts, datado de 1943 [46]. Neste artigo, McCulloch e Pitts descrevem o cálculo lógico para as redes neurais artificiais.

A filosofia das redes neurais foi, então, descrita no estudo do psicólogo Donald Hebb, em seu livro "*Organization of Behavior*", de 1949 [21]. Este livro descreveu pela primeira vez o comportamento das unidades do cérebro (e suas sinapses) e o processo de treinamento do ser humano. Este idéia foi estudada e aprofundada por Rosemblatt, no MIT (USA), culminando na publicação de "*Perceptrons*", em 1958 [63].

Entretanto, o perquisador Minsky mostrou em sua publicação também intitulada "*Perceptrons*", de 1969 [49], que o futuro do modelo descrito por Rosemblatt não era promissor, devido à sua imensa complexidade matemática.

A idéia das Redes Neurais permaneceu latente até 1986, quando os pesquisadores David E. Rumelhart e James L. McClelland publicaram o famoso livro "*Parallel Distributed Processing: Exploration in the microstructure of Cognition*" [64], em dois volumes. Esta obra apresentou um método que propiciou pela primeira vez o treinamento de uma rede neural em modo supervisionado. Este método ficou conhecido como *Backpropagation* [9].

A partir disto, o treinamento e a utilização de redes neurais se tornou tarefa mais palpável, o que propiciou a explosão de desenvolvimento nesta área. Novas técnicas de treinamento foram surgindo, reduzindo consideravelmente o tempo e a complexidade do processo de treinamento e novas tecnologias foram adaptadas e utilizadas em Redes Neurais, como é o caso dos Algoritmos Genéticos e da Lógica Difusa (*Fuzzy*). O comportamento potencialmente caótico das Redes Neurais fez com que fossem aplicadas a esta área novas abordagens, como a Teoria do Caos, a Geometria Fractal e a Teoria da Catástrofe.

Atualmente, as Redes Neurais já apresentam um desenvolvimento considerável, permitindo que aplicações cada vez mais sofisticadas possam ser desenvolvidas.

2.3 O Neurônio Artificial

As Redes Neurais – assim como o cérebro – são caracterizadas pela presença de unidades atômicas de processamento interconectadas. O neurônio artificial apresenta comportamento semelhante ao comportamento do neurônio biológico. O funcionamento do neurônio artificial é baseado em sinais de estímulo ou de inibição, que são processados e repassados. A ligação entre neurônios é caracterizada por sua intensidade, que é representada matematicamente por um valor denominado "*peso*". Portanto, quanto maior o valor do peso, mais intensa é a conexão entre dois neurônios.

Cada neurônio possui terminais de entrada (dendritos) e de saída (terminais sinápticos). Através dos terminais de entrada, os sinais de estímulo ou inibição são recebidos pelo neurônio. Os valores que alimentam um neurônio são agrupados através de uma função denominada *Função de Junção*, onde os valores provenientes de cada terminal de entrada são agregados proporcionalmente ao seu peso. O valor resultante, denominado *Valor de Ativação*, será submetido à *Função de Ativação* do neurônio. O valor resultante é, então, repassado para os neurônios aos quais os terminais de saída estão conectados, ou podem ainda representar, em parte ou integralmente, a saída de toda a rede neural.

Os neurônios artificiais são basicamente classificados por suas funções ou localização na rede neural. Neurônios artificiais podem ser:

1. Neurônios de Entrada

   Recebem a informação que se deseja processar através de um único terminal de entrada (sem peso). Este tipo de neurônio meramente repassa o valor recebido como entrada para todos os neurônios subseqüentes. Portanto, nenhum tipo de processamento é realizado no âmbito de um neurônio de entrada.

2. Neurônios de Processamento

   Recebem valores provenientes de outros neurônios, representando sinais de estímulo ou inibição. Estes valores estão associados aos pesos das conexões por onde os valores

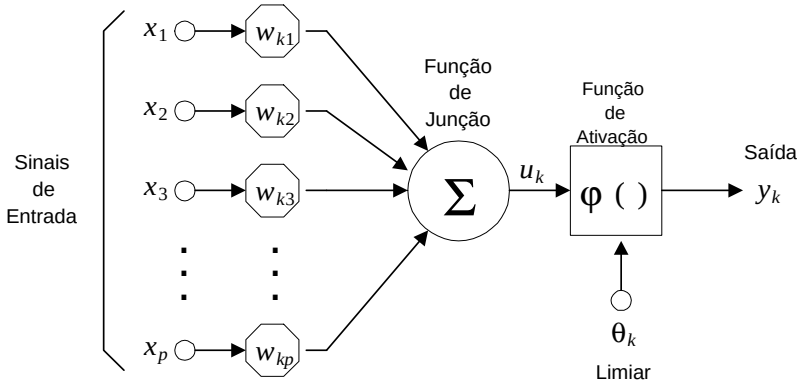


Figura 2.1: Arquitetura de um Neurônio Artificial

são repassados. O valor de saída do neurônio é, então, repassado a todos os neurônios subseqüentes.

3. Neurônios de Saída

   Representam o último passo de processamento antes do resultado da rede neural ser apresentado. Através desses neurônios, os valores provenientes de outros neurônios sofrem o último processamento e seu resultado é apresentado através de um único terminal de saída, representando em parte ou integralmente a resposta da rede neural.

A Figura 2.1 representa a arquitetura de um neurônio artificial k. Os valores x_i (com $i = 1, 2, \cdots, p$) representam os sinais de entrada. Estes valores são associados aos seus respectivos pesos w_{ki} (com $i = 1, 2, \cdots, p$) e agregados através da *Função de Junção*. O valor de ativação u_k é passado à *Função de Ativação* $\varphi(.)$, que recebe como parâmetro adicional um certo limiar denominado θ_k. Este limiar representa um referencial de comportamento do neurônio, que define até onde o neurônio é estimulado ou inibido. O resultado da Função de Ativação, y_k, é repassado aos neurônios subseqüentes (no caso de neurônios de processamento) ou simplesmente apresentado como resultado da rede neural (no caso de neurônio de saída).

A *Função de Junção* (combinador linear) pode ser definida como o somatório com pesos dos valores de entrada do neurônio. Desta forma, tem-se que o valor de ativação v_k do neurônio k é dado por:

$$v_k = \sum_{j=1}^{p} w_{kj}.x_j.$$

2.3.1 Tipos de Função de Ativação

A função de ativação, denotado por $\varphi(.)$, define o valor de saída de um neurônio em termos do nível de atividade em sua entrada [20]. Pode-se identificar três tipos básicos de funções de ativação:

Função Limiar

Esta função, representada na figura 2.2(a), é definida por:

$$\varphi(v) = \begin{cases} 1, & \text{se } v \geq 0; \\ 0, & \text{se } v < 0. \end{cases}$$

O valor de saída do neurônio é, então, definido por:

$$y(v) = \begin{cases} 1, & \text{se } v_k \geq 0; \\ 0, & \text{se } v_k < 0. \end{cases}$$

onde v_k é o valor de atividade interna do neurônio, dado por:

$$v_k = \sum_{j=1}^{p} w_{kj}.x_j - \theta_k$$

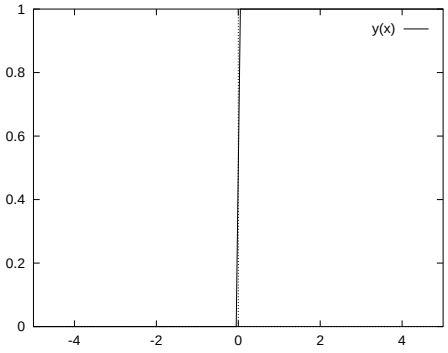
Este tipo de função de ativação para neurônios é também conhecido como *Modelo McCulloch-Pitts*, em alusão aos idealizadores das Redes Neurais.

Função Linear Piecewise

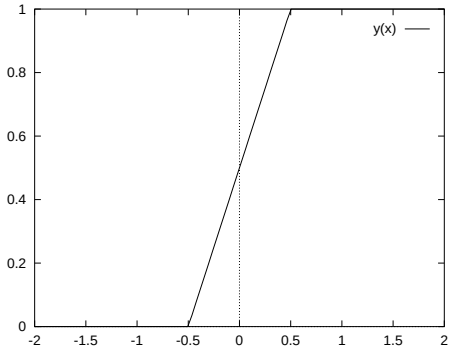
Esta função é descrita na Figura 2.2(b) e é dada por:

$$\varphi(v) = \begin{cases} 1, & \text{se } v \geq \frac{1}{2}; \\ v, & \text{se } \frac{1}{2} > v > -\frac{1}{2}; \\ 0, & \text{se } v \leq -\frac{1}{2}; \end{cases}$$

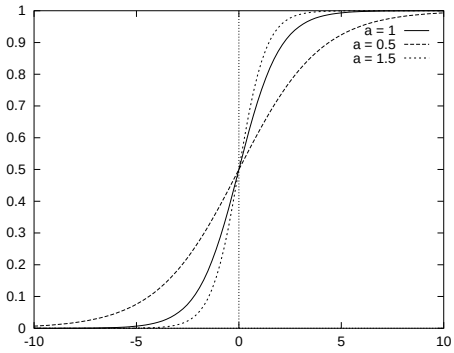
Esta função pode ser considerada uma aproximação de um amplificador não-linear.



(a) Função Limiar



(b) Função Linear Piecewise



(c) Função Sigmoidal

Figura 2.2: Tipos de Função de Ativação

Função Sigmoidal

Este modelo é o tipo mais comum de função de ativação utilizado para projeto de redes neurais. É uma função estritamente crescente que apresenta propriedades assintóticas e de suavidade. Um exemplo de função sigmoidal é a função *logistic*, definida por:

$$\varphi(v) = \frac{1}{1+exp(-av)}$$

onde *a* representa o parâmetro de declividade da função sigmoidal. O parâmetro *a* define as diferentes declives da função sigmoidal, representadas na Figura 2.2(c).

2.4 Arquiteturas de Redes Neurais

A maneira na qual os neurônios são organizados e interconectados define a arquitetura da rede neural. O tipo de organização dos neurônios está diretamente ligado ao tipo de algoritmo ou regra de aprendizagem que deve ser utilizado. Desta forma, pode-se identificar 4 tipos básicos de redes neurais:

- Redes Feedforward Unicamada (Perceptron)
- Redes Feedforward Multicamada
- Redes Recorrentes
- Estruturas Lattice

Cada uma destas classes será descrita a seguir.

2.4.1 Redes Feedforward Unicamada (Perceptron)

Uma rede neural baseada em camadas tem como característica a organização de neurônios de mesma função em camadas. Desta forma, neurônios de uma são conectados através de sinapses aos neurônios de sua camada subseqüente, o que define o termo *Feedforward*. Assim, a camada de entrada engloba neurônios de entrada para a rede neural, ao passo que a camada de saída compreende seus neurônios de saída. Uma rede neural como a ilustrada na Figura 2.3 é considerada uma rede unicamada. Isto porque a primeira camada, composta por neurônios de entrada, apenas repassa os sinais de entrada, não desempenhando nenhum processamento.

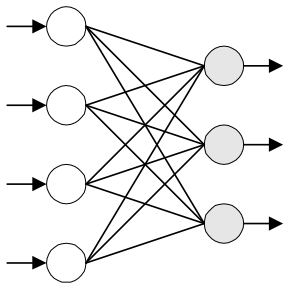


Figura 2.3: Rede Neural Feedforward Unicamada

Uma aplicação de memória associativa não-linear é um exemplo de rede neural unicamada. Esta aplicação é capaz de associar um padrão de saída (vetor) ao seu respectivo padrão de entrada (vetor) [20].

2.4.2 Redes Feedforward Multicamada

Este tipo de rede neural é caracterizado pela presença de uma ou mais *Camadas Ocultas*, além das camadas de entrada e de saída. Na Figura 2.4, tem-se uma rede neural *feedforward* multicamada com uma camada oculta. Esta rede neural pode ser caracterizada como 7–4–2, visto que possui 7 neurônios na primeira camada (entrada), 4 neurônios na segunda camada (oculta) e 2 neurônios na terceira camada (saída). Uma rede neural é considerada totalmente conectada se cada neurônio de uma camada estiver conectado a todos os neurônios da camada subseqüente. Caso contrário, é considerada uma rede neural parcialmente conectada.

2.4.3 Redes Recorrentes

Este tipo de redes neurais difere dos tipos *feedforward* pelo fato de apresentarem um ou mais laços de retro-alimentação. Por exemplo, uma rede neural pode possuir apenas uma camada, onde a saída de cada neurônio é retro-alimentada na entrada dos outros neurônios da camada, como é o caso da Figura 2.5(a). Na Figura 2.5(b), tem-se uma rede neural recorrente com duas camadas, que possui dois terminais de entrada e dois terminais de saída.

Existem ainda estruturas especiais, que são envolvidas aos laços de retro-alimentação, e são

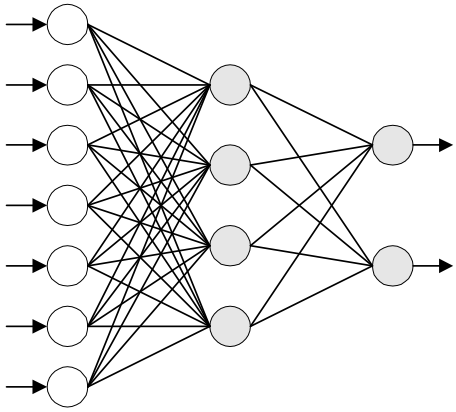


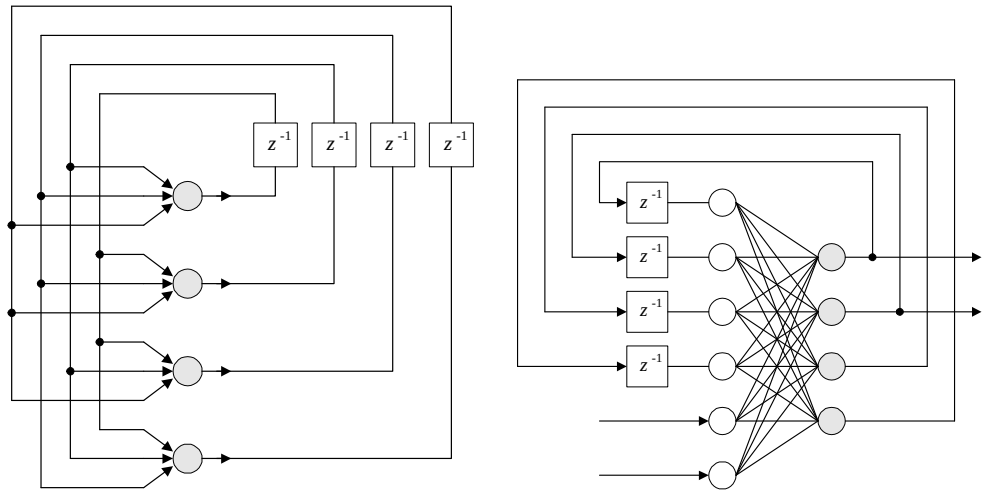
Figura 2.4: Rede Neural Feedforward Multicamada

denominados *Elementos Unit-Delay*, denotados por z^{-1}. Este elemento é responsável pelo comportamento dinâmico não-linear da rede neural, devido à natureza essencialmente não-linear dos neurônios. Este comportamento é bastante importante na função de armazenamento de uma rede recorrente.

2.4.4 Estruturas Lattice

Uma estrutura Lattice consiste em um vetor unidimensional, bidimensional ou multidimensional de neurônios, sendo que uma camada extra composta por neurônios de entrada é responsável de repassar os sinais de entrada para os neurônios que compõem a matriz. A Figura 2.6(a) mostra um exemplo de Estrutura Lattice unidimensional com 3 neurônios, enquanto a Figura 2.6(b) apresenta uma estrutura bi-dimensional de 3-por-3 neurônios. Ambas as estruturas apresentam 3 neurônios na camada de entrada.

Uma estrutura Lattice pode ser considerada uma rede *feedforward* com os neurônios de saída arranjados em linhas e colunas.



(a) Rede Recorrente Unicamada (b) Rede Recorrente Multicamada

Figura 2.5: Rede Neural Recorrente

2.5 Processo de Aprendizagem

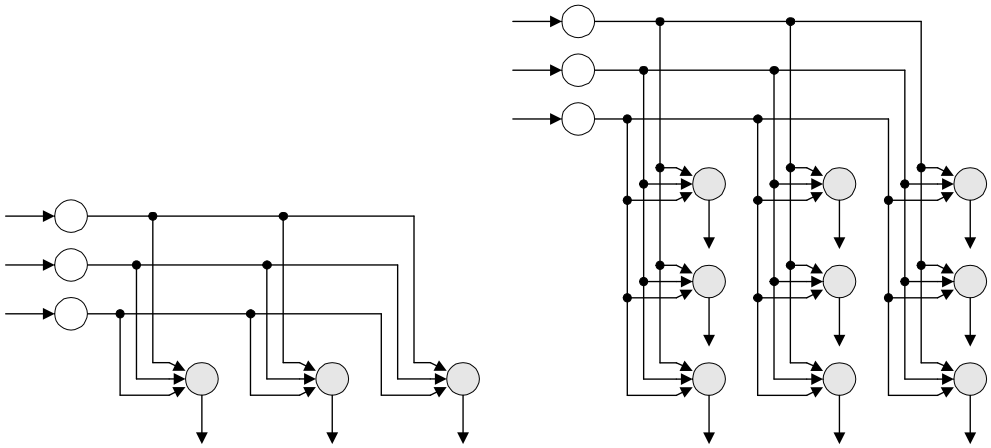
A aprendizagem é o processo pelo qual a rede neural adquire conhecimento de seu ambiente. Este processo é matematicamente descrito pelo sucessivo ajuste dos pesos e limiares que compõem a rede, de forma a se adaptar ao seu ambiente. Este processo envolve basicamente duas decisões. A primeira é acerca de qual paradigma de treinamento deve ser utilizado. A outra decisão é tomada acerca de que variedade de algoritmo (regra) de treinamento que deve ser utilizado (Figura 2.7). Diversos algoritmos foram desenvolvidos apresentando diferentes abordagens matemáticas para que o ajuste dos pesos seja efetuado.

O ajuste dos pesos é modelado da seguinte forma:

$$w_{jk}(n+1) = w_{jk} + \Delta w_{jk}$$

Portanto, o algoritmo de aprendizagem deve apresentar uma abordagem para o cálculo de Δw_{jk} de forma que o custo computacional deste cálculo e o número de iterações necessárias para se chegar a uma taxa de erro aceitável sejam minimizados.

A Tabela 2.1 mostra alguns algoritmos de aprendizagem com seus tipos propícios de aplicação



(a) Estrutura de 3 neurônios (b) Estrutura Bi-dimensional 3-por-3

Figura 2.6: Estruturas Lattice

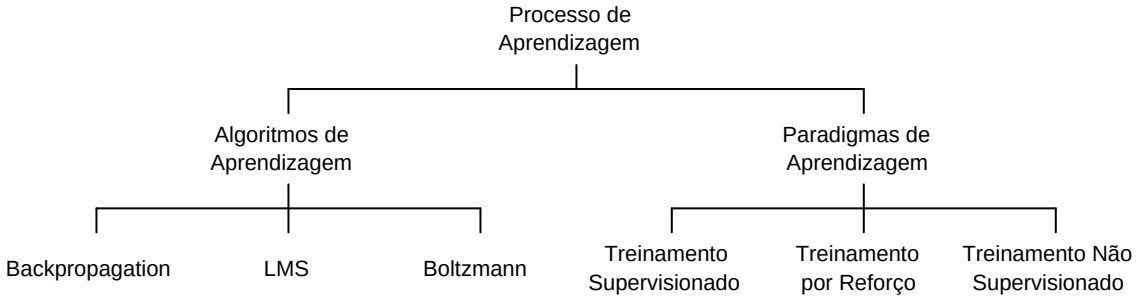


Figura 2.7: Taxonomia do Processo de Treinamento

e de topologia de RNA.

Modelo	Paradigma de Treinamento	Topologia	Funções Primárias
Adaptative Resonance Theory	Não Supervisionado	Recorrente	Agrupamento
ARTMAP	Supervisionado	Recorrente	Classificação
Backpropagation	Supervisionado	*Feedforward*	Classificação
Radial Basis Function Networks	Supervisionado	*Feedforward*	*Time-series*, Classificação
Kohonen feature Maps	Não Supervisionado	*Feedforward*	Agrupamento
Recurrent Back Propagation	Supervisionado	Recorrente	*Time-series*

Tabela 2.1: Modelos e Funções de Redes Neurais

Os paradigmas para treinamento de redes neurais artificiais são:

- Treinamento Supervisionado
- Treinamento por Reforço
- Treinamento Não Supervisionado

Cada um destes paradigmas apresenta características que propiciam classes específicas de aplicações. Adicionalmente, cada algoritmo de aprendizagem está intrinsecamente ligado à topologia de rede neural utilizada.

Cada um destes paradigmas será explicitado a seguir.

2.5.1 Treinamento Supervisionado

Este paradigma é caracterizado pela presença de um "*professor*", que indica à rede neural qual o resultado desejado para cada entrada apresentada. Este valor é, então, comparado com o valor obtido pela rede neural de forma a se conhecer o erro obtido. O "professor" é representado por um conjunto de vetores de entrada associados aos seus respectivos vetores de saída desejados. Estas informações compõem o conjunto de padrões de treinamento. A Figura 2.8 ilustra a interação entre o "professor" e o sistema de aprendizagem.

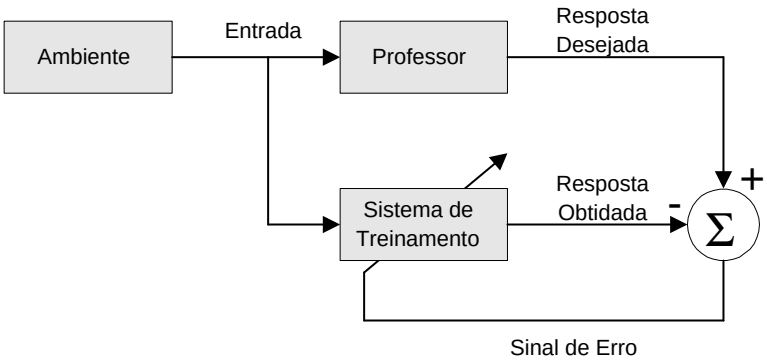


Figura 2.8: Treinamento Supervisionado

O processo de treinamento supervisionado é dividido em ciclos. Em cada ciclo, cada um dos *N* padrões (entrada + saída) do conjunto de padrões de treinamento é apresentado à rede neural. O ajuste dos pesos e limiares pode ser feito de duas formas: ajuste a cada padrão que é apresentado à rede neural (Modo Padrão) ou ajuste a cada ciclo, *i.e.*, depois que todos os *N* padrões são apresentados à rede neural (Modo Batch).

Adicionalmente, o treinamento supervisionado pode ser realizado de duas formas: *on-line* e *off-line*. No treinamento *off-line*, a rede neural é treinada a partir de um conjunto de treinamento até que um erro máximo desejado seja obtido. Após isto, a rede neural é estabilizada e posta em operação. No treinamento *on-line*, a rede neural é treinada em tempo-real, através do conhecimento que lhe é apresentado em tempo de operação. A grande desvantagem do método *on-line* é que o custo computacional para que o treinamento possa ser realizado é, muitas vezes, proibitivo. Por outro lado, a desvantagem do treinamento *off-line* é que uma vez estabilizada, a rede neural não é mais capaz de atualizar seus conhecimentos acerca do ambiente com novas experiências que possam vir a surgir. Para suprir esta desvantagem, o treinamento por reforço é o mais indicado. Este tipo de treinamento será descrito a seguir.

Dentre os algoritmos de aprendizagem supervisionada, o mais comumente utilizado é o *Backpropagation* [73]. Este algoritmo representou a generalização para o algoritmo LMS (*Least-Mean-Square*) [74].

2.5.2 Treinamento por Reforço

O treinamento por reforço se dá através de adaptação de pesos de acordo com um *Sinal de Reforço*. Este sinal, proveniente de um "*crítico*" externo, é responsável por sinalizar se uma resposta produzida pela rede neural foi ou não satisfatória. O esquema do Treinamento por Reforço é ilustrado na Figura 2.9.

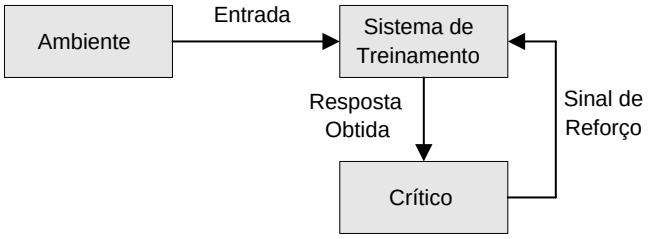


Figura 2.9: Treinamento por Reforço

Se a resposta produzida pela rede neural for satisfatória, um sinal de reforço favorável é produzido pelo crítico, o que faz a rede neural aceitar como boa a resposta produzida. Caso o crítico produza um sinal de reforço não favorável, a resposta é considerada insatisfatória, e a rede neural absorve este conhecimento através do reajuste de seus pesos, com o intuito de não mais repetir esta resposta.

A filosofia do Treinamento por Reforço é originária de uma teoria de adestramento animal que define que uma resposta favorável obtida deve ser recompensada, enquanto respostas desfavoráveis devem ser repreendidas. Desta forma, o animal seria treinado (ou adestrado) a sempre responder a contento, esperando a sua recompensa.

2.5.3 Treinamento Não Supervisionado

O Treinamento Não Supervisionado, também conhecido como *Auto-organização*, é caracterizado pela ausência de qualquer agente externo, seja "professor" (Treinamento Supervisionado" ou "crítico" (Treinamento por Reforço). Ao invés disto, as informações sobre o ambiente são passadas à rede neural sem qualquer informação de classificação. Uma vez adaptada às reg-

ularidades estatísticas dos dados de entrada, a rede neural desenvolve a habilidade de formar representações internas para codificação das entradas e então criar novas classes automaticamente. A Figura 2.10 ilustra o Treinamento Não Supervisionado.

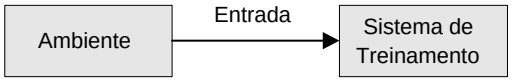


Figura 2.10: Treinamento Não-Supervisionado

Parte II

Motivação

Capítulo 3

Tráfego ATM

"There are no such things as applied sciences,
only application of science."
– Louis Pasteur

3.1 Introdução

As redes ATM são caracterizadas por suportar aplicações com aspectos amplamente diversificados. Esta complexidade torna cada vez mais necessária uma classificação de tráfego de células ATM capaz de organizar e viabilizar a operação de toda esta diversidade, oferecendo e mantendo os requisitos de Qualidade de Serviço para cada fonte de tráfego envolvida no sistema. Assim, um fator importante para a operação eficaz de uma rede ATM é seu controle de tráfego. Novos mecanismos de gerência são necessários para que os novos requisitos do tráfego ATM possam ser mapeados.

Este capítulo descreve o tráfego ATM, seus requisitos e mecanismos de controle.

3.2 O Tráfego ATM

Entende-se por *Tráfego ATM* o fluxo de células provenientes de fontes de tráfego por enlaces e comutadores de uma rede ATM. Estas fontes de tráfego representam as entidades de rede que produzem células e as submetem à rede através de suas interfaces (UNI ou NNI). Estas entidades podem se concretizar como aplicações de usuários ou como o próprio sistema.

A arquitetura dos serviços suportados pela camada ATM consiste das seguintes categorias [7]:

- CBR (*Constant Bit Rate*)

  Aplicações cuja taxa de transmissão não varia durante toda a duração da conexão. A classe CBR é passível de suportar aplicações em tempo-real que demandem um restrito controle da variação de atraso de transmissão de células (*e.g.* voz, vídeo, emulação de circuito), mas não se restringe só a estes tipos de serviço.

- rt–VBR (*Real-Time Variable Bit Rate*)

  Aplicações caracterizadas por taxa de transmissão variável ao longo do tempo, mantendo-se um controle restrito de variação de atraso de células. Essa classe engloba, dentre outras, aplicações de voz e vídeo. Esse tipo de aplicação é caracterizado pelo tráfego em rajada (*bursty traffic*).

- nrt–VBR (*Non-Real-Time Variable Bit Rate*)

  Aplicações como as rt-VBR, mas que não exigem controle de atraso de células. Essa classe pode suportar a multiplexação estatística. Aplicações dessa classe também apresentam tráfego em rajada.

- ABR (*Available Bit Rate*)

  Essa classe de aplicações difere dos outros pelo fato da variação da taxa de transmissão nessa categoria depender da quantidade de largura de banda disponível e não do comportamento da aplicação em si. Mecanismos de controle de fluxo baseados em *feedback* são

definidos para essa classe de forma a controlar a taxa de transmissão de aplicações de acordo com a disponibilidade de recursos. Esse controle é realizado através de células especiais denominadas *Células de Gerenciamento de Recursos* (RM-cell). É através dessas células que as informações de *feedback* trafegam pela rede. Essa classe de aplicações não exige controle de variação de atraso, o que não propicia aplicações de tempo-real.

- UBR (*Unspecified Bit Rate*)

  Essa classe de serviços não apresenta nenhuma descrição de tráfego ou pré-requisitos. A largura de banda deixada ociosa por aplicações dos tipos CBR e VBR é utilizada por aplicações UBR. Portanto, a quantidade de largura de banda disponível para essas aplicações depende do comportamento das demais fontes VBR e CBR. Essa classe de aplicações é geralmente mais aplicada para adaptação de serviços baseados no tráfego TCP, que apresentam um esquema de utilização do *melhor esforço*.

A Figura 3.1 ilustra a ocupação da largura de banda de um enlace entre as classes de aplicação. As aplicações do tipo CBR ocupam uma quantidade de largura de banda constante, enquanto o restante é utilizado pelas fontes de tráfego VBR. A capacidade deixada sem uso por estas aplicações é, então, dividida entre fontes ABR e UBR.

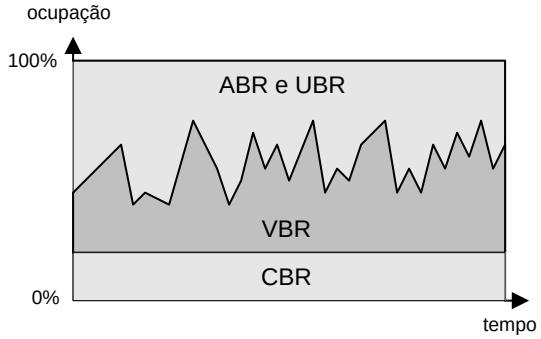


Figura 3.1: Utilização de Largura de Banda por Classes de Aplicação

3.3 Contrato de Tráfego

O ATM Forum definiu em [7] alguns parâmetros que podem caracterizar o padrão de tráfego de um fluxo de células [69]. As características de uma fonte de fluxo ATM são capturadas em um *descritor de fonte de tráfego*, que inclui as seguintes informações:

- Taxa de Pico de Transmissão (PCR - *Peak Cell Rate*);
- Taxa Sustentável de Células (SCR - *Sustainable Cell Rate*);
- Tamanho Máximo de Rajada (MBS - *Maximum Burst Size*);
- Taxa Mínima de Células (MCR - *Minimum Cell Rate*).

O parâmetro PCR define o limite máximo na taxa de transmissão de células de uma fonte de tráfego. O parâmetro SCR define o limite superior para a taxa de transmissão média de células, que é calculado durante um determinado intervalo de tempo. Para aplicações de taxa constante, o valor do parâmetro SCR coincide com o PCR ($SCR = PCR$), enquanto em aplicações de taxa variável, tem-se que $SCR < PCR$.

O parâmetro MBS representa o número máximo de células que podem ser transmitidas a taxa de pico (PCR) a cada rajada. Portanto, após uma rajada, a taxa de transmissão de células deve decrescer de forma a se manter até o limite SBR.

O parâmetro MCR é utilizado em aplicações do tipo ABR e indica qual a quantidade mínima de largura de banda exigida por uma conexão. Desta forma, o sistema poderá controlar a taxa de transmissão de uma fonte ATM, mantendo-a entre os valores inferior MCR e superior PCR.

Outros parâmetros são definidos pelo ATM Forum para caracterizar a Qualidade de Serviço necessária para cada aplicação. Estes parâmetros incluem:

- *Peak-to-Peak Cell Delay Variation* (peak-to-peak CDV);
- *Maximum Cell Transfer Delay* (maxCTD); e
- *Cell Loss Ratio* (CLR).

O parâmetro CTD descreve o tempo passado entre a transmissão do último bit de uma célula na UNI fonte e a recepção do primeiro bit desta célula na UNI destino. Em termos gerais, CTD é uma variável que tem sua função de densidade de distribuição típica semelhante à Figura 3.2. Nesta figura, existe um atraso mínimo, chamado atraso fixo, que inclui o atraso de propagação através do meio físico, atrasos induzidos pelo sistema de transmissão e componentes fixos de atraso de processamento em comutadores. A porção variável do atraso (CDV) é devido ao enfileiramento e ao escalonamento de células [69].

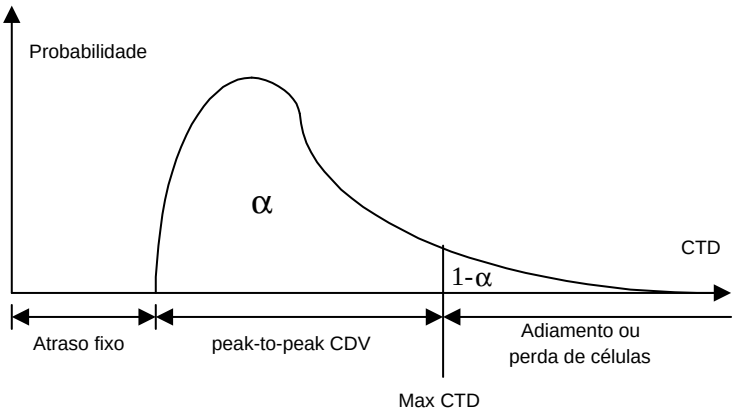


Figura 3.2: Distribuição de Probabilidade para o Parâmetro CTD

Na Figura 3.2, maxCTD define o atraso máximo tolerado pela conexão. A fração $(1-\alpha)$ de todas as células vai exceder este limite e deve ser descartada ou enviada posteriormente. A proporção restante α encontra-se dentro da QoS desejada. Portanto, a quantidade de atraso incidente nestas células está dentro do intervalo entre o atraso fixo e o maxCTD. Este intervalo é denominado peak-to-peakCDV [7].

O parâmetro CLR representa a razão entre o número de células descartadas (perdidas) e o total de células transmitidas na conexão.

As classes de tráfego com suas respectivas propriedades e requisitos são explicitados na Tabela 3.1.

No momento do estabelecimento de uma conexão, o sistema precisa ter conhecimento das características e requisitos da nova aplicação de forma a avaliar a viabilidade da alocação de

	Categoria de Serviço				
Atributo	CBR	rt–VBR	nrt–VBR	UBR	ABR
Parâmetros de Tráfego					
PCR e CDVT (4,5)	Especificado			Especificado (2)	Especificado (3)
SCR, MBS, CDVT (4,5)	N/A	Especificado		N/A	
MCR (4)	N/A				Especificado
Parâmetros de Qualidade de Serviço					
peak-to-peak CDV	Especificado		Não Especificado		
maxCTD	Especificado		Não Especificado		
CLR (4)	Especificado			Não Especificado	(1)
Outros Atributos					
Feedback	Não Especificado				Especificado

1. O CLR é baixo para fontes que ajustam seu fluxo de células em resposta a informações de controle.
2. Pode ou não ser controlado por procedimentos de CAC e UPC
3. Representa a taxa máxima de uma aplicação ABR. A taxa real depende de informações de controle.
4. Estes parametros podem ser implicita ou explicitamente especificados para PVC's ou SVC's.
5. CDVT não é negociado. Em geral, o CDVT precisa ter um único valor para a conexão.

Tabela 3.1: Atributos das Categorias de Serviços ATM

recursos. Para isto, cada nova conexão deve apresentar à rede parâmetros que a descrevam. Uma conexão é, portanto, caracterizada por estes três componentes:

- Descritor de fonte de tráfego

  Parâmetro que descreve o comportamento do fluxo de células gerado pela fonte de tráfego. Esta caracterização é composta por parâmetros quantitativos como PCR, MBS e SBR.

- Parâmetro CDVT

  Este parâmetro especifica um limite máximo de atraso de transmissão que a aplicação pode suportar. Este parâmetro é necessário para que se possa realizar o policiamento acerca dos parâmetros de Qualidade de Serviço negociados no momento do estabelecimento da conexão.

  O parâmetro de QoS CDV não deve ser confundido com o parâmetro de descritor de tráfego CDVT. O CDV é geralmente negociado durante o estabelecimento da conexão, enquanto o CDVT normalmente é explicitado na UNI e não é negociado.

- Definição de conformidade

  Parâmetro que é utilizado para determinar quais células estão em conformidade com os

parâmetros negociados e quais não estão. A função GRCA (*Generic Cell Rate Algorithm*) é um exemplo de algoritmo utilizado para esta função.

3.4 Mecanismos de Gerência de Tráfego

O Gerenciamento de Tráfego tem como função monitorar e controlar o fluxo de células de uma rede ATM, de modo a evitar ou reagir a congestionamentos, mantendo a sincronia entre a transmissão e a recepção de células em conexões. As duas principais tarefas para a gerência de tráfego são, então, controle de fluxo e controle de congestionamento [58].

O controle de fluxo lida com a sincronização entre fonte e receptor de tráfego, de modo que a geração de células na fonte seja compatível com a disponibilidade de recepção na outra ponta da conexão.

O controle de congestionamentos visa evitar, reagir ou minimizar os efeitos de situações de perda excessiva de células, como um congestionamento. Um congestionamento é observado quando há um aumento excessivo na demanda de largura de banda, provocando perda excessiva de células devido à limitação de largura de banda nos *links* de saída e de espaço no *buffer*, causando a degradação da Qualidade de Serviço das aplicações envolvidas.

A probabilidade de ocorrência de congestionamentos depende diretamente do tipo de multiplexação utilizada: Determinística ou Estatística. Na Multiplexação Determinística, a cada conexão está alocada a sua taxa de pico. Desta forma, a taxa de utilização do *link* dificilmente superará a sua capacidade. A Multiplexação Estatística permite que o somatório das taxas de pico das conexões ultrapasse a capacidade do *link*, mantendo, entretanto, a probabilidade de um congestionamento abaixo de um certo limiar ε. Apesar da multiplexação determinística praticamente impossibilitar situações de congestionamento, há subutilização dos recursos, uma vez que as conexões nem sempre transmitem à taxa de pico. No caso de conexões do tipo CBR, entretanto, a Multiplexação Determinística não proporciona subutilização de recursos, pois as taxas de pico dessas conexões coincidem com suas taxas médias de transmissão.

A complexidade da tarefa do controle de congestionamento se deve principalmente às seguintes

dificuldades:

- Diferença de taxa de transmissão entre diferentes fontes;
- Uma única fonte pode gerar vários tipos de tráfego (voz, dados, vídeo) com diferentes características;
- Necessidade de lidar adicionalmente com a variação do atraso das células, atraso máximo e desvios estatísticos;
- Serviços possuem diferentes parâmetros de Qualidade de Serviço;
- As características de tráfego de diversos tipos de serviços ainda não estão bem especificados;
- A alta velocidade de transmissão limita o tempo disponível para o processamento nos nós intermediários.

Diversos mecanismos iniciais estão definidos em [7] para o controle de congestionamento em redes ATM. Dentre eles, estão:

1. Controle de Admissão de Conexões (CAC);
2. Controle de Parâmetros de Uso e de Rede (UPC/NPC);
3. Técnicas de Notificação de Nós Terminais;
4. Descarte Seletivo;
5. Remodelagem de Tráfego;
6. Descarte de Quadros;
7. Controle de Fluxo ABR.

Cada mecanismo será descrito a seguir.

3.4.1 Controle de Admissão de Conexões (CAC)

O Controle de Admissão de Conexões é um conjunto de ações tomadas pela rede durante o estabelecimento ou renegociação de uma conexão, com o intuito de determinar se a rede pode ou não aceitar esta conexão. Esta decisão deverá ser tomada de forma a somente aceitar novas conexões se a QoS das conexões já ativas não for afetada. O mecanismo de controle de admissão de novas conexões está ilustrado na Figura 3.3.

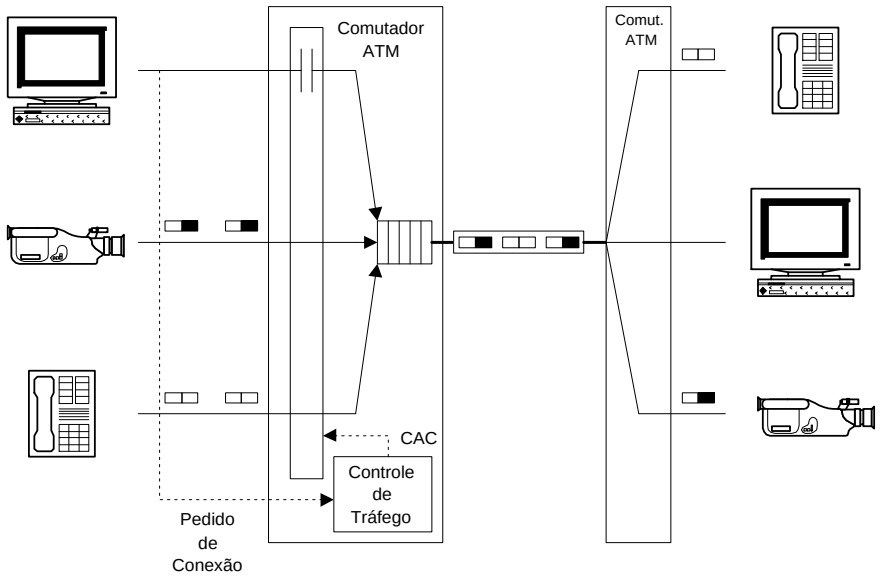


Figura 3.3: Funcionamento do CAC

O processo de decisão acerca da aceitação da nova conexão é realizado no controle de tráfego, onde mecanismos e algoritmos são definidos para estimar o impacto de uma nova conexão no sistema como um todo, levando em conta a disponibilidade de recursos e a Qualidade de Serviço desejada.

As informações utilizadas pelo controle de conexões são as descrições de suas características de tráfego e os parâmetros de QoS desejados (se aplicável). Para uma conexão ser aceita, deve-se haver uma avaliação sobre o impacto desta conexão sobre a rede, através de uma projeção de seu comportamento [59]. A Figura 3.4 mostra o diagrama de decisão de aceitação de uma nova VCC.

Alguns algoritmos como o *Equivalent Bandwidth* (EB) [15] e a Aproximação Gaussiana [41]

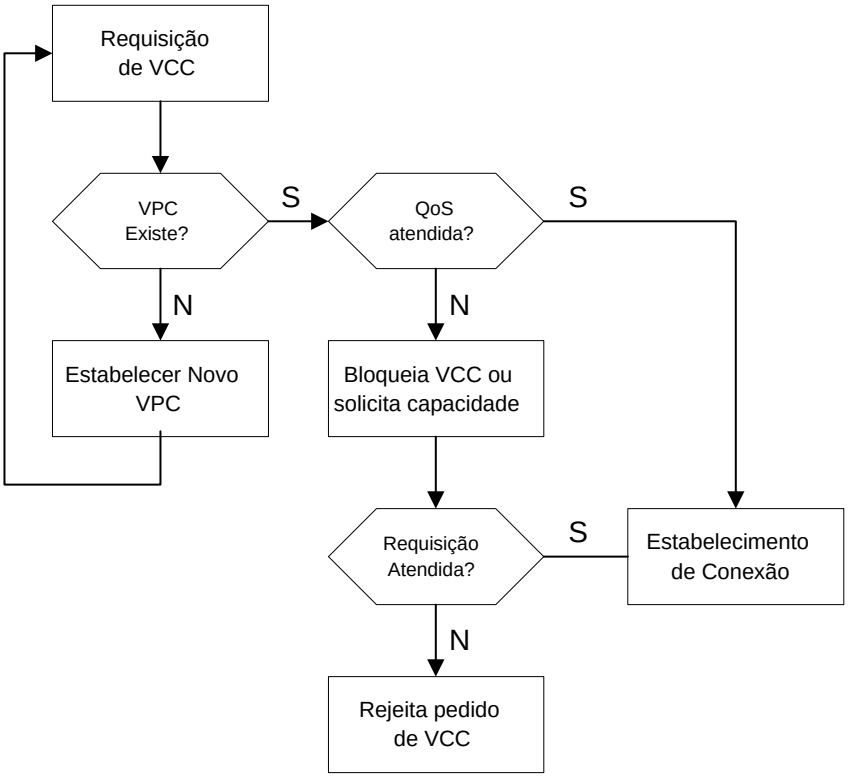


Figura 3.4: Diagrama de Decisão do CAC

tentam definir uma quantidade de largura de banda para cada fonte de tráfego de tal forma que a probabilidade de haver perda de células esteja abaixo de um certo limiar arbitrário. O algoritmo EB é descrito em maiores detalhes no Capítulo 4.

3.4.2 Controle de Parâmetros de Uso e Rede

Uma vez estabelecida uma conexão, de acordo com os parâmetros declarados em seu contrato de tráfego, é necessário se certificar de que este contrato seja devidamente cumprido. Esta é a tarefa do Controle de Parâmetros de Uso (UPC - *Usage Parameter Control*) e do Controle de Parâmetros de Rede (NPC - *Network Parameter Control*).

O Controle de Parâmetros de Uso lida com o policiamento de tráfego nas interfaces UNI da rede, enquanto o Controle de Parâmetros de Rede atua sobre o tráfego nas interfaces NNI.

Os Controles de Parâmetros de Uso e de Rede se utilizam de funções específicas para a detecção de anormalidades de tráfego. Estas funções incluem:

- Checagem da validade de valores de VPI e VCI;
- Monitoramento do volume de tráfego que entra na rede a partir de todas as conexões virtuais ativas, de forma a assegurar que os parâmetros do contrato de tráfego estão sendo cumpridos;
- Monitoramento do total de tráfego aceito nos *links*.

A partir da utilização destas funções, pode-se detectar conexões que desrespeitam seus contratos de tráfego. O sistema de UPC/NPC deve, então, atuar sobre o tráfego destas conexões. Esta atuação deverá garantir ou obrigar o cumprimento dos parâmetros estabelecidos.

Neste caso, as ações que podem ser tomadas pelo sistema de UPC/NPC incluem:

- Descarte de células excedentes;

  Neste caso, todas as células que não estão em conformidade com os limites fixados no contrato de tráfego serão descartadas.

- Atraso na transferência de células;

  Antes de acessar a rede, o tráfego da fonte infratora é submetido a uma fila, que adapta a sua taxa de saída à taxa de transmissão acordada em seu contrato de tráfego.

- Redução de prioridade de células excedentes;

  Em caso de congestionamento nos *links* de saída, as células marcadas de baixa prioridade serão imediatamente descartadas. A diminuição da prioridade das células excedentes é feita através do campo CLP no cabeçalho de cada célula.

- Controle de *feedback* à fonte.

  Quando um contrato de tráfego começar a ser violado, o sistema de UPC/NPC deverá comunicar esta desconformidade à sua fonte.

A maioria dos esquemas definidos para o controle de parâmetros se utiliza das informações de taxas média, taxa de pico e duração do período ativo para realizar o policiamento sobre conexões. De posse dessas informações, algoritmos como *"Leaky Bucket"* monitoram e controlam o comportamento de conexões na rede. O algoritmo *Leaky Bucket* será descrito a seguir.

Leaky Bucket

A idéia básica deste algoritmo, que foi posteriormente estendida, é a utilização de *tokens* (fichas) em um reservatório (*token pool*), caracterizado por sua capacidade, e uma fonte de *tokens*, que gera fichas a uma taxa constante.

Cada célula, antes de ter acesso à rede, deverá consumir um *token* do reservatório. Se o *token pool* estiver vazio, as células que solicitarem *token* ou serão descartadas ou serão marcadas com baixa prioridade em caso de congestionamento.

A Figura 3.5 mostra o esquema *Leaky Bucket* mais simplificado, onde células que não disponham de *tokens* são descartadas. Algumas implementações dispõem de um *buffer* para armazenar um determinado número de células enquanto novos *tokens* são gerados. Outras

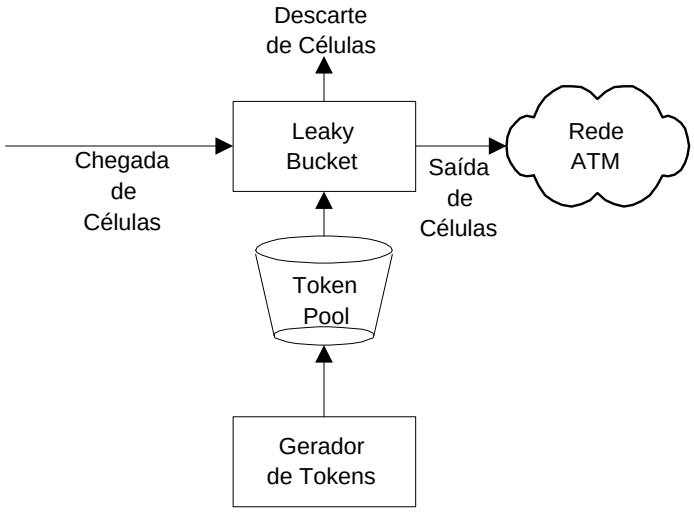


Figura 3.5: Algoritmo Leaky Lucket

implementações podem, ao invés de descartar células sem *token* disponível, marcá-las como de baixa prioridade no caso de descarte.

O esquema *Leaky Bucket* pode realizar policiamento de tráfego tanto em nível de VC quanto em nível de VP. Para uma concessionária, por exemplo, é mais interessante o policiamento de tráfego em nível de VP, visto que cada cliente é visto como um conjunto de valores agregados de suas conexões VC.

Os parâmetros do *Leaky Bucket* determinam a tolerância do policiamento de tráfego. O tamanho do *token pool* determina o tamanho máximo de rajada e a taxa de geração, a tolerância a atrasos e as variações de atraso; enquanto a taxa de geração de novos *tokens* controla a taxa média ou de pico da conexão.

Este esquema foi escolhido e recomendado tanto pela ATM Forum quanto pela ITU-T para o policiamento de tráfego das redes ATM.

Técnicas de Janelas

Outra técnica de policiamento de tráfego é a limitação do número de células transmitidas em um certo intervalo fixo de tempo, denominado janela. Portanto, após a transmissão de seu

número limite de células por janela, as demais células serão descartadas ou marcadas de baixa prioridade, até que a janela seja finalizada.

Algumas variações deste algoritmo sugerem janelas não consecutivas ou tamanho de janelas variável como formas de aumentar o desempenho e reduzir a complexidade de processamento.

3.4.3 Técnicas de Notificação de Nós Terminais

Uma vez detectado um congestionamento em um nó intermediário, é necessário que os nós terminais (fonte e destino) sejam notificados, de modo a reagir ao problema. Três técnicas principais foram propostas para a notificação de congestionamento. São elas:

- *Estimation by the End Nodes*

  Nesta técnica, uma fonte de tráfego gera células especiais com registro de tempo (*time stamp*) chamadas *probe cells*. Estas células são repassadas através da rede pelos nós intermediários como células normais. Somente no nó terminal de destino estas células são processadas. Ao chegar ao destino, as *probe cells* são utilizadas para estimar o atraso de transmissão entre fonte e destino. Uma vez detectado um possível congestionamento, a fonte deverá ser notificada.

  O principal problema desta técnica é a necessidade de submissão de tráfego extra na rede. Apesar do custo de transmissão de uma célula em uma conexão ser ínfimo, a existência de milhares de conexões pode causar desperdício de recursos. Além disso, esta técnica não localiza o congestionamento, apenas notifica a possibilidade de sua existência.

  Esta técnica não foi adotada pelos órgãos de padronização das redes RDSI-FL, sendo deixada como opção para os provedores de serviços de rede.

- *Explicit Backward Congestion Indication*

  Nesta técnica, cada nó intermediário monitora as filas de seus *links*. Quando uma dessas filas ultrapassa um limite de ocupação especificado, células especiais são enviadas a todas as fontes de tráfego que utilizam este nó nas rotas de suas conexões. Os campos dessas

células podem ser utilizados para carregar informações acerca do nó congestionado. Uma vez recebida a notificação de congestionamento pelo nó terminal, o ponto de conexão deve reduzir ou suspender a submissão de tráfego à rede até que a situação se normalize.

Apesar de sua eficácia, este método não tem sido adotado pelos órgãos de padronização, devido à necessidade de processamento extra de células nos nós intermediários. Além disso, se uma conexão for configurada para um só sentido, o nó congestionado deverá estabelecer uma nova conexão para que a notificação seja entregue à outra extremidade da conexão.

- *Explicit Forward Congestion Indication*

  Nesta técnica, um nó congestionado ou em iminência de congestionamento deverá comunicar os nós seguintes deste problema. Esta notificação é feita através da ativação do bit EFCI constante no cabeçalho, definido pela ITU-T. A recepção por parte do ponto de conexão de células com o bit EFCI marcado indica que há um nó em congestionamento ao longo do caminho da conexão. Entretanto, não há como saber em que ponto exatamente a rede está congestionada.

  Devido ao atraso pelo tempo em que as células chegam ao destino, é possível que o nó que iniciou a notificação não mais esteja congestionado. Portanto, o destino não deve reagir imediatamente a uma notificação de congestionamento EFCI. Em vez disso, o destino deve coletar informações estatísticas que possam ratificar a situação de congestionamento contínuo. Se o congestionamento for realmente observado, o destino deverá enviar uma notificação de congestionamento à fonte de tráfego, a qual deverá reduzir ou suspender a submissão de tráfego desta conexão à rede.

  Esta técnica foi sugerida pela ATM Forum como mecanismo opcional para gerenciamento de tráfego, em sua especificação [7].

3.4.4 Descarte Seletivo

Como foi citado anteriormente, o gerenciamento de tráfego visa controlar o fluxo de células por *links* e *buffers* nas entidades que compõem esta rede. Entretanto, a necessidade de maximizar a utilização dos recursos expõe a rede ao perigo de congestionamento e de *buffer overflows*.

No caso de um congestionamento, o procedimento é o descarte de células excedentes. Porém, devido à possibilidade de distinção de prioridade entre células (através do bit de cabeçalho CLP), é aconselhável a distinção entre células no caso de necessidade de descarte.

Os mecanismos de descarte seletivo são formas de descarte de células que seguem o critério de que células de alta prioridade devem ter preferência na ocupação de *buffers* e de *links* em caso de congestionamento.

Em caso especial, se o número de células descartadas influir diretamente no requisito de CLR (*Cell Loss Ratio*) de uma determinada conexão, o controle de descarte deverá realizar algum tipo de compensação de forma a respeitar o requisito de CLR desta conexão.

Mecanismos especiais para o descarte seletivo de células foram desenvolvidos. Dois desses mecanismos serão descritos a seguir:

- *Push-Out*

  Neste mecanismo, células de alta e baixa prioridade são submetidas à rede, sob a condição de haver espaço disponível nos *buffers* dos nós intermediários. Uma célula de baixa prioridade será descartada se chegar a um nó cujo *buffer* esteja cheio. Caso uma célula de alta prioridade chegue a um nó e não houver células de baixa prioridade no *buffer* cheio, esta célula será descartada. Mas se houver célula de baixa prioridade no *buffer*, esta será substituída pela célula de alta prioridade que chegar. O mecanismo *Push-Out* é ilustrado na Figura 3.6.

  O principal problema deste esquema é a complexidade de implementação. Apesar da substituição de células parecer uma tarefa simples, é necessário que se mantenha a ordem de seqüência das células. Este fato torna este esquema mais complexo.

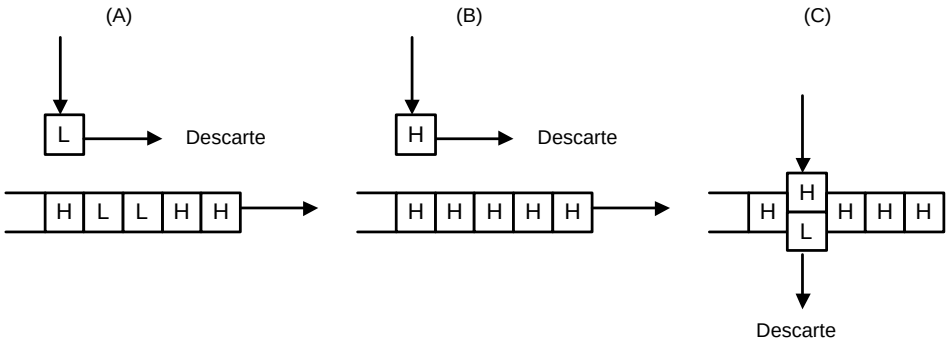


Figura 3.6: Push-Out

- *Threshold*

  Neste caso, utiliza-se se um valor limite (*threshold*) inferior ao tamanho do *buffer* para controlar a sua ocupação entre células de alta e baixa prioridade.

  Células de alta e baixa prioridade serão aceitas no *buffer* até que a sua ocupação limite seja alcançada. A partir desse ponto, células de baixa prioridade serão descartadas, até que a ocupação do *buffer* esteja abaixo do seu valor limite.

  O principal desafio desta técnica é o cálculo do valor limite. Se este valor for muito alto, o desempenho das células de alta prioridade será afetado, visto que não haverá muito espaço reservado para estas células. Por outro lado, se o valor limite for muito baixo, células de baixa prioridade serão descartadas desnecessariamente, uma vez que começarão a ser descartadas prematuramente, ainda havendo espaço relativamente suficiente para a operação normal.

  Embora a determinação do valor limite dependa das características do tráfego de ambas as seqüências de células (alta e baixa prioridades), ênfase maior é dada às características de tráfego das células de alta prioridade, de forma a garantir sua QoS. Portanto, é aconselhável o reajuste do parâmetro de valor limite quando as características de tráfego mudam.

  O algoritmo *Threshold* é exemplificado na Figura 3.7.

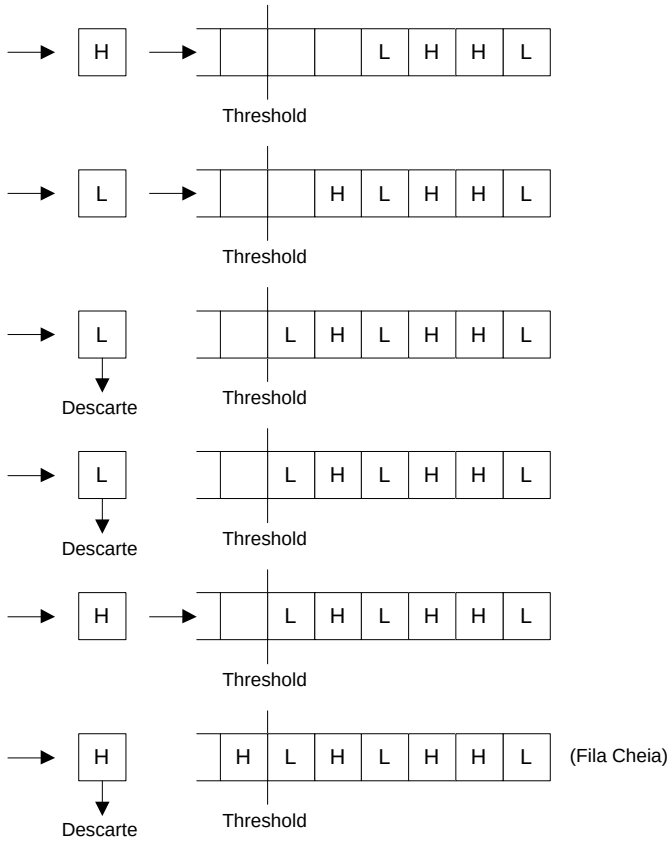


Figura 3.7: Algoritmo Threshold

Comparativamente, ambas as técnicas de descarte seletivo (*Push-Out* e *Threshold*) apresentam praticamente o mesmo comportamento de perda de células, se o valor limite da implementação *Threshold* estiver adequada ao seu tipo de tráfego. Porém, a técnica *Threshold* vem recebendo preferência por sua maior simplicidade de implementação, uma vez que o descarte de células nesta técnica não altera a ordem de seqüência das células no *buffer*.

3.4.5 Remodelagem de Tráfego

A modelagem de tráfego é um mecanismo que altera as características de tráfego originais de uma seqüência de células em uma conexão, de forma a alcançar uma maior eficiência na rede, despeitando a QoS da conexão, ou assegurar o cumprimento de seu contrato de tráfego.

De acordo com as informações de taxa de pico, taxa média e duração de período ativo, o limite de taxa de pico é remodelado a um valor mais baixo. Desta forma, uma rajada seria remodelada de forma a reduzir a taxa de pico e, conseqüentemente, aumentar a duração do período ativo. Isto é feito através do uso de filas de saída. Células de uma conexão são armazenadas em uma fila de saída, antes de serem submetidas à redes. A fila, por sua vez, libera as células à rede a uma taxa menor do que a taxa de pico da conexão, reduzindo a taxa de pico e aumentando o período ativo.

Entretanto, a modelagem de tráfego deverá levar em conta a taxa média de transmissão de modo a não desestabilizar a conexão. Portanto, o cálculo da nova taxa de pico deverá seguir critérios que respeitem a taxa média de transmissão de uma fonte.

Por exemplo, considere-se uma conexão VBR caracterizada por períodos ativos de transmissão a taxa de pico e nenhum tráfego no período de silêncio. Esta conexão tem taxa de pico de 16 Mbps e períodos ativo e de silêncio de 0,5 e 1,5 ms, respectivamente. Uma modelagem de tráfego que reduz a taxa de pico para 8 Mbps pode ser feita, aumentando o período ativo para 1 ms. É suposto que existe espaço suficiente em *buffer* para que esta remodelagem seja feita.

O principal problema é que a modelagem de tráfego influi negativamente em conexões sensíveis a atrasos. Isto ocorre porque as células são sujeitas a uma retenção na fila de saída, e são liberadas à rede a uma taxa menor do que a taxa real da conexão. Portanto, a eficiência da modelagem de tráfego depende diretamente da tolerância da conexão a atrasos.

3.4.6 Descarte de Quadros

Se um elemento de rede necessitar descartar células, em muitos casos o descarte em nível de quadros pode ser mais eficaz. Um quadro representa a Unidade de Dados de Protocolo (PDU - *Protocol Data Unit*) da Camada de Adaptação (AAL).

O descarte em nível de quadro pode ser feito sempre que for possível a identificação das delimitações de um quadro, através da análise de sua Unidade de Dados de Serviço (SDU - *Service Data Unit*). Entretanto, o descarte de quadros só poderá ser feito se a conexão estiver habilitada a esta ação, seja definido no momento do estabelecimento da conexão ou através de

sinalização.

Os mecanismos de decisão sobre o descarte de quadros são dependentes da implementação desta técnica.

Capítulo 4

Estimativa da Capacidade Requerida

"To invent, you need a good imagination
and a pile of junk."
– Thomas Edison

4.1 Introdução

Uma das principais características das redes ATM é a possibilidade de multiplexar diversas conexões de características distintas, com garantias de manutenção de Qualidade de Serviço aos diversos usuários desta rede.

Dentre esta variedade de conexões, as fontes de tráfego com comportamento baseado em taxa de transmissão constante já são bem compreendidas e estudadas, não representando maior complexidade para alocação de recursos e operação.

Por outro lado, as fontes de tráfego baseadas em taxa de transmissão variável incorporam maior complexidade, devido a novos requisitos. Em se tratando de caracterização de fontes de tráfego, observa-se que fontes do tipo CBR podem ser facilmente descritas. A caracterização de uma conexão (chamada) no sistema de telefonia convencional, por exemplo, é definida apenas

com a sua taxa de transmissão, que é 64 Kbps. Entretanto, o tráfego variável (VBR) inclui algumas variáveis que dificultam sua caracterização. A principal razão desta complexidade é a natureza estocástica do tráfego variável. Assim, torna-se difícil capturar o comportamento de fontes de tráfego deste tipo sob forma de variáveis descritoras.

A caracterização de uma fonte de tráfego VBR é calcada em valores que estatisticamente estimam seu comportamento real. Portanto, a tarefa de dimensionar o tráfego de uma fonte com comportamento estocástico é considerada complexa, e projetistas de aplicações nem sempre conseguem mapeá-la. Assim, técnicas de gerência de tráfego que utilizam os descritores de tráfego das aplicações como variáveis principais podem ser influenciadas negativamente por dimensionamentos imprecisos por parte do projetista da aplicação.

Neste capítulo, são discutidos tipos de multiplexação, Capacidade Requerida e Ganho Estatístico no tráfego ATM. O problema da Estimativa da Capacidade Requerida é mais bem definido e analisado.

4.2 Os Tipos de Multiplexação

Os dois tipos básicos de multiplexação aplicados à operação de uma rede ATM, de modo a suportar as duas naturezas básicas do tráfego, são a *Multiplexação Estatística* e a *Multiplexação Determinística* [59].

A Multiplexação Determinística aloca a cada fonte de tráfego largura de banda equivalente a sua taxa de pico (PCR). Desta forma, tem-se a garantia de que cada conexão tem, a qualquer momento, recursos disponíveis para a sua operação normal. Entretanto, conexões com taxa de transferência variável (VBR ou ABR) podem produzir um desperdício de largura de banda no momento em que uma conexão estiver transmitindo células abaixo de sua taxa de transmissão de pico. Portanto, nota-se que este tipo de multiplexação é mais indicado para conexões do tipo CBR, onde não existe a diferença entre taxa média e taxa de pico, sendo, portanto, o desperdício de largura de banda nesta multiplexação ínfimo ou nulo. Considerando que a incidência de fontes de tráfego que apresentam taxas variáveis é, em muitos casos, maior do que a incidência de conexões com taxa constante [47], a aplicação da Multiplexação Determinística nestes casos não é a mais aconselhável, por causa do desperdício de recursos gerado.

A Multiplexação Estatística permite que para cada fonte de tráfego seja alocada uma quantidade de largura de banda inferior à sua taxa de pico. Assim, o objetivo da Multiplexação

Estatística é fazer um aproveitamento das larguras de banda destas aplicações de taxa variável, utilizando recursos estatísticos para garantir a Qualidade de Serviço desejada para todo o sistema. Portanto, este modo de multiplexação permite que sejam agregadas aplicações cujas taxas de transmissão de pico totalizem uma quantidade de largura de banda maior do que a capacidade do enlace de saída.

Sejam L_{out} a capacidade de um enlace de saída, R a taxa de transmissão agregada instantânea de todas as fontes de tráfego destinadas a este enlace em um dado instante e ξ a capacidade do *buffer* de um comutador operando em Multiplexação Estatística. É possível identificar três estados distintos de operação neste equipamento:

1. Se $R < L_{out}$, o tamanho da fila de saída do comutador permanece constante caso esteja vazia, ou decresce a uma taxa $L_{out} - R$;

2. Se $R = L_{out}$, o tamanho da fila permanece constante;

3. Se $R > L_{out}$, o tamanho da fila aumenta a uma taxa $R - L_{out}$ até que atinja a sua capacidade máxima ξ.

Uma situação especial ocorre quando o último estado de operação citado persiste por tempo suficiente até que a ocupação do *buffer*, que cresce a uma taxa $R - L_{out}$, atinge a sua capacidade máxima ξ. A partir deste momento, células serão descartadas por não poderem ser transmitidas pelo enlace saturado e também por não poderem ser armazenadas em *buffer* para envio posterior. Esta situação de perda excessiva de células e degradação da Qualidade de Serviço das aplicações é denominada *congestionamento* [38]. Portanto, a Multiplexação Estatística possibilita a ocorrência de congestionamentos, uma vez que há probabilidade da demanda por largura de banda superar a capacidade do enlace de saída. Por outro lado, esta probabilidade de congestionamento deve ser controlada. A quantidade de largura de banda alocada para cada fonte deve ser definida de modo a observar este limite máximo de probabilidade de congestionamento [14].

A Multiplexação Determinística, por outro lado, praticamente extingue esta possibilidade de congestionamento, visto que o terceiro estado de operação não pode ocorrer: o valor de R é

sempre menor ou igual à capacidade do enlace (L_{out})[1].

Embora a Multiplexação Estatística possa propiciar situações de congestionamento, seu poder de aproveitamento de recursos e conseqüente maximização de utilização do sistema encorajam a sua adoção, em detrimento à Multiplexação Determinística.

4.3 Capacidade Requerida e Ganho Estatístico

A Multiplexação Estatística propõe que a capacidade ociosa de cada fonte de tráfego seja reaproveitada através da definição e alocação lógica de uma quantidade de largura de banda menor do que a taxa máxima (PCR) de cada aplicação. Desta forma, o sistema como um todo pode operar de forma satisfatória. Esta capacidade alocada é denominada *Capacidade Requerida*, a quantidade mínima de largura de banda que deve ser alocada a uma fonte de tráfego de modo a satisfazer os parâmetros de Qualidade de Serviço de todas as fontes de tráfego envolvidas no nó.

Com a Capacidade Requerida de uma fonte de tráfego, pode-se calcular o ganho proporcionado pela escolha da Multiplexação Estatística. Este ganho é uma medida da economia de recursos que se obtém quando da aplicação da Multiplexação Estatística. Apesar deste artifício expor o sistema a circunstâncias de congestionamento, a probabilidade de isso ocorrer deve ser mantida abaixo de um certo limiar ε pelo mecanismo de estimativa da Capacidade Requerida. Este valor representa um dos requisitos de Qualidade de Serviço (QoS) do sistema, o CLP (*Cell Loss Probability*).

O cálculo do ganho devido à Multiplexação Estatística é obtido pela diferença proporcional entre a taxa de pico de uma aplicação e a quantidade de largura de banda alocada (*Capacidade Requerida*). Portanto, quanto menor é a capacidade equivalente em relação à taxa de pico, maior é o ganho estatístico. Portanto, se uma aplicação j tem como características sua taxa de pico PCR_j e sua Capacidade Requerida CR_j. Então, o ganho estatístico GE_j desta aplicação é dado

[1]Excetua-se, neste caso, condições de desrespeito ao contrato de tráfego. Neste caso, uma ou mais fontes de tráfego chegam a transmitir a uma taxa maior do que a taxa de pico declarada. Portanto, cabe ao mecanismo de Policiamento de Tráfego evitar esta situação.

como:

$$GE_j = 1 - \frac{CR_j}{PCR_j}$$

Por exemplo, para uma aplicação qualquer j caracterizada por sua taxa de pico $PCR_j = 6,4\,Mbps$ e sua Capacidade Requerida $CR_j = 4,8\,Mbps$ apresenta um ganho estatístico de:

$$GE_j = 1 - \frac{CR_j}{PCR_j} = 1 - \frac{4,8\,Mbps}{6.4\,Mbps} = 1 - 0,75 = 0,25 = 25\%$$

O ganho devido à Multiplexação Estatística é influenciado por diversas variáveis, dentre elas a descrição do comportamento das conexões, o tamanho do *buffer* do comutador em questão e a capacidade dos enlaces de saída [61]. O comportamento de uma conexão é geralmente caracterizado por um conjunto de valores como: tipo de conexão (VBR, ABR, CBR, UBR), taxa de transmissão de pico (*PCR*), número de estados, tamanho médio de cada estado, taxa de transmissão de cada estado, etc.

A principal aplicação para a Capacidade Requerida (CR) é o mecanismo de Controle de Admissão de Conexões (CAC). Valores de CR podem ser utilizados para avaliar se a rede tem condições de receber uma nova conexão sem degradar a Qualidade de Serviço do sistema como um todo. Neste caso, uma nova conexão só será aceita se a CR de todas as aplicações (incluindo a nova conexão) multiplexadas em um determinado enlace de saída não superar a capacidade deste enlace. Outra aplicação para a CR é o Gerenciamento de Recursos. Neste caso, o valor da CR pode ser utilizado para a renegociação do contrato de tráfego de aplicações, redimensionamento de VP's, etc. A Capacidade Requerida é útil também para o policiamento de tráfego, dimensionamento e projeto de redes ATM, etc.

Uma das grandes dificuldades de se encontrar um valor para a Capacidade Requerida de uma conexão é ter uma descrição o mais confiável possível de seu tráfego gerado. Isto nem sempre é conseguido, visto que o tráfego ATM é um assunto ainda não muito bem desvendado [47]. Portanto, por mais aproximado que seja o método de estimativa da Capacidade Requerida, uma caracterização errônea do tráfego pode levar tanto a um valor superestimado, causando desperdício de recursos, quanto a um valor subestimado, aumentando a probabilidade

de congestionamento e podendo afetar a Qualidade de Serviço de todo o sistema.

4.4 Fontes de Tráfego VBR ON-OFF

A partir do tipo mais simples de tráfego, o CBR, é possível modelar aplicações de taxa variável como sendo fontes com mais de um estado. A cada estado a fonte produz tráfego a uma determinada taxa constante. Em ordem de complexidade, após o tráfego CBR tem-se o tráfego VBR de dois estados [57].

O tráfego VBR ON–OFF é classificado como tendo dois estados: o estado ON (período ativo) e o estado OFF (período inativo[2]). Durante o período ativo, células são produzidas a uma taxa constante equivalente à taxa de pico da aplicação, enquanto durante o período inativo nenhum tráfego é produzido pela fonte. O comportamento de uma fonte de tráfego VBR ON–OFF é ilustrado na Figura 4.1.

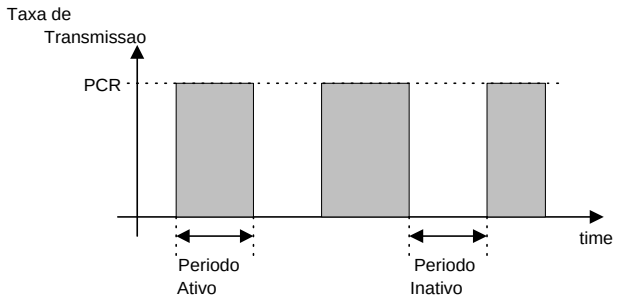


Figura 4.1: Exemplo de Fonte de Tráfego ON–OFF

Fontes deste tipo com duração de períodos ativo e inativo distribuídos como exponencial negativo têm sido freqüentemente estudadas e aplicadas à descrição de tráfego de dados e como um modelo geral para o tráfego em rajadas em um comutador ATM [61].

A Figura 4.2 mostra o diagrama de estados de uma fonte de tráfego ON-OFF. Após cada célula produzida pela fonte, uma outra célula é gerada com probabilidade $1 - \lambda$, ou a fonte passa para o estado inativo com probabilidade λ. Da mesma forma, no período inativo, a fonte

[2]Este termo é também conhecido como *Período de Silêncio*

se mantém sem produzir tráfego por mais um intervalo de célula (*cell slot*) com probabilidade $1-\rho$, ou alterna para o período ativo com probabilidade ρ.

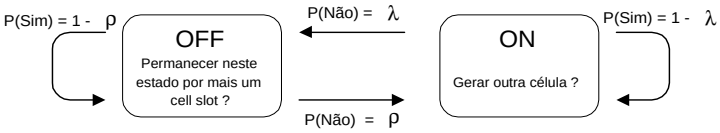


Figura 4.2: Diagrama de Estados de Fontes ON–OFF

O tráfego VBR ON-OFF pode ser modelado de outra forma. Ao invés de considerar o processo de geração de células no período ativo e de geração de *cell slots* no período inativo como processos de Bernoulli, pode-se simplesmente descrever o período ativo pelo número de células produzidas como sendo exponencialmente distribuído, e o período inativo como tendo número de *cell slots* produzidos distribuído geometricamente. O número médio de células produzidas no período ativo, $E(on)$, é, então, o inverso da probabilidade da fonte sair do período ativo. Da mesma forma, o número médio de *cell slots* produzidos durante o período inativo, $E(off)$, é igual ao inverso da probabilidade da fonte passar para o período inativo.

$$E(on) = \frac{1}{\lambda} \qquad\qquad E(off) = \frac{1}{\rho}$$

Vale ressaltar que as distribuições geométricas dos estados ativo e inativo têm diferentes bases de tempo. Para o período ativo, a unidade de tempo é $1/R$, onde R é a taxa de geração de células no período ativo (PCR), *i.e.*, $1/R$ representa o tempo entre chegada de células. O período inativo é parametrizado pela unidade de tempo $1/C$, onde C é a taxa de geração de *cell slots* do comutador. Estes períodos estão exemplificados na Figura 4.3.

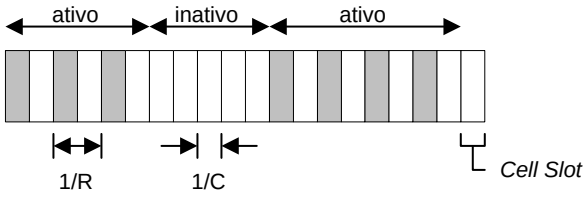


Figura 4.3: Padrão de *Cell Slots* para uma Fonte VBR ON–OFF

O tamanho do *cell slot* é definido de acordo com a capacidade do enlace de saída a qual a fila de saída está ligada. Por exemplo, para um enlace de capacidade $155,52$ *Mbps,* o *cell slot* é definido como:

$$C = \frac{53 \text{ (bytes/cell)} \times 8 \text{ (bits/byte)}}{155,52 \times 10^6} = 2,726 \times 10^{-6} s = 2,726 \mu s$$

Desta forma, a duração média do período ativo, t^{on} é dada por:

$$t^{on} = \frac{1}{R}E(on)$$

Da mesma maneira, a duração média do período inativo, t^{off}, é definida por:

$$t^{off} = \frac{1}{C}E(off)$$

No escopo deste trabalho, foram consideradas apenas aplicações do tipo VBR caracterizadas como ON-OFF. Para este tipo de tráfego, os parâmetros utilizados para sua caracterização são:

- Taxa de Transmissão de Pico (PCR);
- Tamanho médio do período ativo t^{on};
- Tamanho médio do período inativo t^{off}.

4.5 Métodos de Estimativa da Capacidade Requerida

O cálculo da Capacidade Requerida para uma conexão e para o tráfego agregado é uma tarefa complexa. Muitas variáveis têm que ser levadas em conta ao passo que a ausência de um método determinístico para um resultado exato influencia nesta dificuldade.

O mecanismo de estimativa de um valor para a Capacidade Requerida de uma fonte deverá levar em conta seu descritor de tráfego, que fornece informações acerca do seu comportamento esperado, e informações acerca do sistema, como a capacidade do *buffer* do comutador (tamanho da fila de saída). Mecanismos para este fim diferem entre si na quantidade e na qualidade das informações de entrada e na especialidade de tráfego à qual mais se adequa.

Existem diversas abordagens que tentam estimar o ganho estatístico obtido da Multiplexação Estatística. Dentre elas está a *Equivalent Bandwidth* (ou *Equivalent Capacity*).

4.5.1 Equivalent Bandwidth

O método *Equivalent Bandwidth* (EB), proposto em [15], faz a estimativa da Capacidade Requerida de cada conexão individual com tráfego do tipo VBR ON–OFF. Para isto, este método se utiliza das seguintes informações:

- Informações das Aplicações (para cada aplicação j):
  - Taxa de transmissão de pico PCR_j;
  - Tamanho médio do período ativo t_j^{on};
  - Tamanho médio do período inativo t_j^{off};
  - Taxa de utilização da fonte $\rho_j = \frac{t_j^{on}}{t_j^{on}+t_j^{off}}$.
- Informações do Sistema:
  - Tamanho do buffer ξ;
  - Probabilidade máxima de perda de células desejada ε.

Assim, a Capacidade Requerida segundo o método *Equivalent Bandwidth* é estimada como[3]:

$$EB_j = PCR_j \times \frac{y_j - \xi + \sqrt{(y_j-\xi)^2 + 4\xi\rho_j y_j}}{2y_j} \tag{4.1}$$

onde

$$y_j = \alpha t_j^{on}(1-\rho_j)PCR_j \quad \text{e} \quad \alpha = ln(1/\varepsilon)$$

Uma das características do método *Equivalent Capacity* é a sua escalabilidade. Desta forma, o valor *Equivalent Bandwidth* do tráfego agregado é obtido do somatório dos EBs das fontes de tráfego individuais [59].

$$EB = \sum_{j=1}^{N} EB_j$$

[3] A derivação do método EB está apresentada no Apêndice B

O método *EB* é considerado uma forma rápida e suficientemente precisa para o cálculo da Capacidade Requerida. Entretanto, a utilização de uma caracterização de tráfego equivocada, seja sub ou superestimada, pode levar a uma valor indesejado da Capacidade Requerida. Portanto, surge a necessidade de um método que leve em conta principalmente o comportamento real do tráfego, em detrimento de descritores de tráfego que podem deturpar o cálculo da Capacidade Requerida.

4.6 Estimativa da Capacidade Requerida baseada em Medições de Tráfego

Os descritores de tráfego de aplicações são as principais fontes de informação para mecanismos de estimativa da Capacidade Requerida em redes ATM. Porém, pode haver situações onde os descritores de tráfego não apresentem a precisão suficiente para uma resposta confiável, ou situações onde fontes de tráfego mudem o seu comportamento durante a vigência de seu contrato, tornando-o inconsistente. Além disto, o custo computacional de métodos analíticos para este fim dificultam a operação em tempo-real, visto que a maioria dos métodos existente se emprega a aplicações individuais, e Capacidade Requerida do tráfego agregado deve ser obtido pela aplicação deste método a cada fonte multiplexada.

Portanto, torna-se necessário um mecanismo para estimar a Capacidade Requerida de um certo padrão de tráfego agregado que seja mais flexível a imprecisões dos descritores de tráfego e que ao mesmo tempo apresentem dinamicidade suficiente para operação em tempo-real.

O método de estimativa de Capacidade Requerida baseado no comportamento do tráfego agregado deverá observar os padrões deste tráfego por um intervalo de tempo suficiente para que as informações de patamar de operação e variabilidade de tráfego possam ser definidos como um caso genérico. A partir disto, outras situações serão comparadas a este caso genérico e classificados de acordo.

Parte III

Proposta

Capítulo 5

Arquitetura RENATA

"Nothing is particularly hard
if you devide it into small jobs."
– Henry Ford

5.1 Introdução

Com o objetivo de propor uma solução para a gerência pró-ativa de redes ATM, foi idealizada a Arquitetura RENATA (**RE**des **N**eurais para a **A**dministração do **T**ráfego **A**TM) [58].

Esta arquitetura surgiu da insuficiência ou da ausência de abordagens pró-ativas para redes ATM em situações críticas, onde a generalização para casos novos deve se antecipar ao problema de forma a evitá-lo ou, pelo menos, minimizar os seus efeitos [65]. Esta é a idéia da pró-atividade. Outro fator importante é a escassez de soluções analíticas e exatas de custo computacional viável para certas classes de problemas. Além disto, soluções convencionais podem não ser viáveis para problemas mais complexos, que poderiam envolver um número de variáveis considerável, ou ainda podem não conseguir modelar a natureza essencialmente estocástica do tráfego ATM.

Nesta arquitetura, foi adotado o uso de simulação. Uma situação simulada serve de substrato para a geração de bases de conhecimento sobre um determinado problema [60]. Adicionalmente, o uso de simulação facilita experimentações em estados críticos, que não são de simples obtenção em ambientes reais.

Na RENATA, a extração de conhecimento para um determinado problema é realizada por uma Rede Neural Artificial, que se utiliza da experiência simulada para interagir com situações reais de tráfego ATM. A escolha de Redes Neurais para esta tarefa se deve a características próprias como adaptabilidade e generalização, que permitem extrapolar do universo simulado o ambiente real. Além disto, o tempo de resposta das redes neurais é importante para a gerência de redes ATM, pois a alta velocidade de transmissão e os requisitos de tempo-real para a gerência de tráfego ATM não permitem grande *overhead* de processamento para muitas das soluções analíticas existentes.

O objetivo da arquitetura RENATA é, portanto, gerar situações simuladas que refletem o ambiente real de modo a servir de fonte de conhecimento para que uma rede neural possa ser treinada acerca desta experiência e assim agir sobre os recursos reais. Esta idéia está ilustrada na Figura 5.1. Portanto, quanto mais próxima a simulação do ambiente real, maior a possibilidade de se extrair conhecimento relevante para o treinamento eficaz da rede neural.

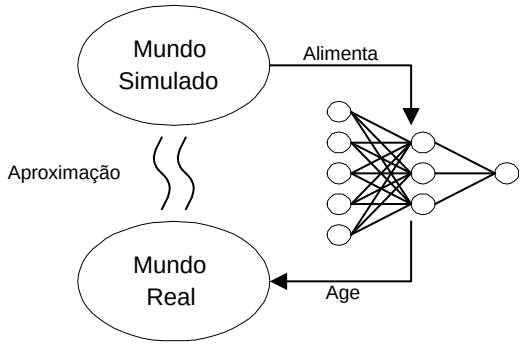


Figura 5.1: RENATA - Aproximação através de Simulação

O funcionamento básico da arquitetura RENATA é ilustrado na Figura 5.2 e obedece aos seguintes passos:

1. Uma rede ATM real serve de base para a geração de modelo de simulação (*Rede ATM Virtual*);

2. O modelo simulado gerado é alimentado ao simulador de redes ATM;

3. O produto do simulador ATM é utilizado para a formação de uma base de conhecimento (*baseline*) acerca do problema abordado;

4. A partir do conhecimento obtido de simulação, um banco de exemplos é gerado representando o conhecimento adquirido da simulação;

5. O banco de exemplos é utilizado como base para o treinamento de uma Rede Neural Artificial;

6. O resultado do processo de treinamento da rede neural é avaliado através de testes e validações;

7. A rede neural treinada para resolver o problema abordado é anexada a algum mecanismo de gerência de tráfego ATM;

8. O módulo gerado desta fusão é colocado em operação, interagindo com recursos reais da rede ATM.

Todos estes passos são ilustrados na Figura 5.2.

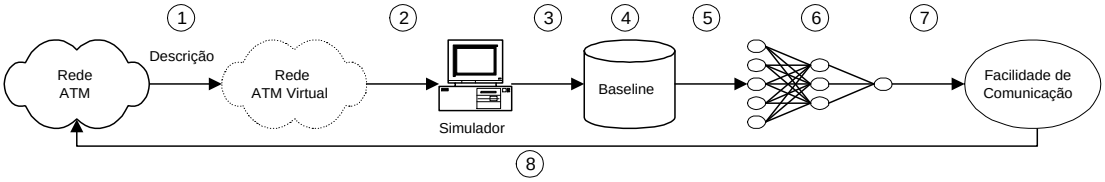


Figura 5.2: RENATA - Diagrama Geral de Solução de Problemas

O mecanismo funcional desta arquitetura permite que problemas de gerência de tráfego em redes ATM, como Controle de Admissão de Conexões, Policiamento de Tráfego, Gerenciamento de Recursos, Estimativa de QoS e Controle de Fluxo sejam mapeados. Em especial, o

problema da Estimativa da Capacidade Requerida do Tráfego ATM é um dos problemas específicos de Gerenciamento de Recursos. Este problema é abordado neste trabalho.

5.2 Arquitetura Funcional

A Figura 5.3 mostra a Arquitetura Funcional da RENATA, composta por 3 módulos: *Módulo de Treinamento*, *Módulo Neural* e *Módulo de Gerência*. A interação entre estes modelos é regida pelas Políticas de Monitoramento e pelas Políticas de Controle, descritas a seguir. Estes módulos são compostos por ferramentas integradas, com o intuito de oferecer a funcionalidade desejada.

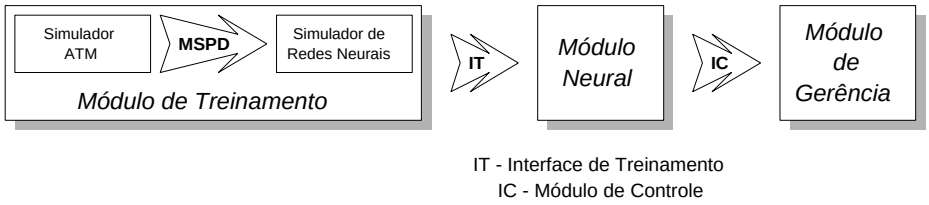


Figura 5.3: Arquitetura Funcional da RENATA

Os módulos da arquitetura RENATA são descritos a seguir.

5.2.1 Módulo de Treinamento

O *Módulo de Treinamento* tem como função produzir uma rede neural devidamente treinada e validada para a solução de um dado problema. Portanto, são tarefas do Módulo de Treinamento a produção de um ambiente simulado, a geração de um banco de conhecimento acerca do problema proposto, baseado nos resultados das simulações, o projeto de uma rede neural para a solução do problema abordado e a realização de seu treinamento sobre a base de conhecimento gerada.

O *Módulo de Treinamento* é composto por um simulador de redes ATM, um *Módulo de Seleção e Preparação de Dados* (MSPD) e um simulador de redes neurais. Ele corresponde aos passos de 1 a 7 da Figura 5.2.

A função do simulador de redes ATM é obter um *trace* (acompanhamento) da operação do ambiente simulado, registrando as informações relevantes que servem de fonte para a geração de uma base de conhecimento. O simulador ATM deve receber variadas configurações de rede sobre uma mesma topologia. Portanto, cada configuração deve conter informações estáticas como a topologia física da rede ATM real a ser gerenciada, capacidades de enlaces de dados e *buffers*; e as informações dinâmicas de cada configuração, como a quantidade e os descritores das fontes de tráfego envolvidas em cada ponto final da rede. A simulação deve ser realizada de modo a se obter uma variedade suficiente de configurações para a geração de um banco de conhecimento representativo da área de atuação do sistema. Esta área de atuação é característica do ambiente real e define sob quais limites inferior e superior a rede neural deverá inferir e generalizar. Por exemplo, uma rede neural pode ser treinada para inferir sobre um ambiente onde o número mínimo de aplicações envolvidas no ponto de rede seja N_{min} e o número máximo seja N_{max}. Portanto, a resposta obtida da aplicação da rede neural em um ambiente com mais de N_{max} ou com menos de N_{min} aplicações ativas poderá não ter a qualidade desejada, pois a rede neural foi treinada dentro desta área de atuação, e a sua generalização usualmente se restringe ao inferior deste intervalo.

Para cada configuração simulada é produzido um *log* dos parâmetros relevantes colhidos durante o tempo de simulação. A escolha destes parâmetros é regida pelas Políticas de Monitoramento, a serem descritas posteriormente.

O Módulo de Seleção e Preparação de Dados (MSPD) é responsável por colher do *log* produzido por cada configuração simulada as informações relevantes para a geração de seus exemplos representativos. Cada configuração pode gerar um ou mais exemplos. Dados brutos são colhidos do *log* de simulação de cada configuração que posteriormente são processados e normalizados de acordo com as resoluções das Políticas de Monitoramento. O resultado deste processo é um ou mais pares de vetores de entrada e de saída para a rede neural, representando o conhecimento acerca desta configuração.

Todos os pares de vetores de todas as configurações simuladas são unidos em um conjunto que representa o banco de conhecimento. Parte deste conjunto compõe o banco de exemplos

para treinamento da rede neural enquanto o restante compõe o banco de validação.

O último componente do Módulo de Treinamento é o simulador de redes neurais. É através desta ferramenta que a rede neural que faz parte da arquitetura é projetada, treinada, testada e validada. O simulador de Redes Neurais recebe como entrada o produto do Módulo de Seleção e Preparação de Dados (MSPD). Será, portanto, um banco de exemplos que irá servir de base para o treinamento supervisionado da rede neural definida. No final do processo de treinamento, são realizados testes e em seguida a validação do resultado obtido. É produzido, ao final do processo de treinamento, um código intermediário representando a rede neural *treinada*, que irá servir de base para a operação do *Módulo Neural*. Este código é repassado ao Módulo Neural através da *Interface de Treinamento* (IT).

5.2.2 Módulo Neural

O Módulo Neural representa o resultado de todo o processo do Módulo de Treinamento. Este componente pode se concretizar como o código em linguagem de programação da rede neural treinada, *i.e.*, com seus pesos já devidamente ajustados e armazenados; ou como um componente de *hardware* VLSI (*Very Large Scale Integration*) [43] dedicado à implementação desta rede neural.

A escolha do tipo de Módulo Neural implementada é definida de acordo com o tipo de problema abordado e com o tipo de integração que esta rede neural terá com recursos de rede reais. Por exemplo, se o problema abordado exigir tratamento em tempo-real, o mais indicado é a utilização de interface de *hardware* conectada diretamente ao objeto de ação, como um comutador, por exemplo. Porém, se o problema abordado exigir menor interação em tempo-real, o Módulo Neural pode se personificar como um código em linguagem C que é adaptado a algum mecanismo de gerência de redes, como o *Tivoli Netview*, por exemplo. Neste caso, há uma restrição de operacionalização, porque mecanismos de gerência baseados em protocolos de rede apresentam o *overhead* de transmissão de informações de monitoramento e de controle via redes, o que penaliza o seu uso em controles de tempo-real.

O Módulo Neural é responsável pela inferência sobre estados diversos da rede real acerca de

um determinado problema. Portanto, sua função é receber um estímulo e produzir uma resposta. Este estímulo representa a entrada da rede neural, que descreve o estado que se deseja inferir. Estes parâmetros são regidos pelas Políticas de Monitoramento. A resposta, então, representa o resultado obtido pela rede neural à entrada apresentada. Esta resposta deve ser interpretada como a ação de controle que deve ser tomada em relação à situação apresentada como entrada. Esta ação de controle é definida de acordo com as Políticas de Controle. Portanto, o Módulo Neural necessita de meios de se comunicar com os recursos reais de rede, de modo a receber estímulos e repassar respostas. Esta comunicação é realizada pelo Módulo de Gerência, através da Interface de Controle.

5.2.3 Módulo de Gerência

O Módulo de Gerência tem como função oferecer meios para a comunicação entre o Módulo Neural e os recursos reais de rede. Este módulo deve oferecer meios de passar ao Módulo Neural as informações relevantes (estímulos) acerca do *status* da rede e também meios de atuar sobre recursos de rede de acordo com a resposta produzida pelo Módulo Neural, que define a ação correspondente que deve ser desencadeada.

Este módulo se concretiza como um agente, implementado em software ou através de alguma interface específica, que possui meios de obter informações de recursos de rede. Ele também é autorizado a agir sobre estes. Um exemplo de concretização do Módulo de Gerência é a implementação de um agente de gerenciamento baseado em algum protocolo Gerente / Agente, como o SNMP (*Simple Network Management Protocol*) [12] ou o CMIP (*Common Management Information Protocol*) [62].

Portanto, este módulo fornece as informações que alimentam o Módulo Neural, assim como serve como agente modificador dos recursos de rede, realizando as operações sugeridas pelas Políticas de Controle.

5.3 Políticas

O processo de adaptação da arquitetura RENATA para os mais diversos problemas inclui a definição de algumas instruções. Estas informações irão reger o processo de preparação e de operação do protótipo projetado.

As *Políticas de Monitoramento* definem os parâmetros (variáveis) cruciais para o problema proposto. Estas informações são importantes de modo que se decida quais variáveis do ambiente simulado devem ser registradas em *log* de simulação para que a base de conhecimento possa ser gerada. Estes parâmetros devem ser escolhidos de maneira a representar o mais fielmente o problema que se deseja combater.

Por exemplo, para o problema de Controle de Admissão de Conexões (CAC) (Capítulo 3), os parâmetros mais relevantes a serem monitorados são dados sobre as aplicações já operantes no nó, informações acerca da nova conexão que deseja se estabelecer e informações sobre capacidades dos enlaces de dados e do *buffer* do comutador. Já para o problema de Policiamento de Tráfego, as informações mais relevantes que devem ser observadas são os limites de utilização de recursos das conexões e o acompanhamento do fluxo de células nos enlaces de entrada do comutador.

As Políticas de Monitoramento servem também para a definição de fatores como a topologia de rede neural a ser utilizada e o algoritmo de aprendizagem que melhor se aplica.

As *Políticas de Controle* são regras que definem o comportamento da arquitetura RENATA na segunda fase de sua implementação, isto é, na fase operacional. Estas regras ditam quais atitudes o protótipo deve tomar de acordo com uma determinada resposta obtida da rede neural acerca de um problema.

Por exemplo, no caso do problema de Controle de Admissão de Conexões (CAC), a resposta da rede neural treinada deverá ser interpretada acerca da ação que o protótipo deve tomar. Neste caso, a decisão pode ser aceitar a nova conexão, quando os valores da resposta indicam que os requisitos para isto são satisfeitos; ou rejeitar a nova conexão, caso contrário.

Portanto, as Políticas de Controle definem o comportamento de agentes e sistemas baseados

na arquitetura RENATA de acordo com a resposta obtida a partir da rede neural.

5.4 Vantagens e Desvantagens

Dentre as vantagens do uso da arquitetura RENATA estão:

- Abstração de situações do mundo real

  O uso de simulação torna desnecessária a observação de situações reais de tráfego, que nem sempre exibem um comportamento interessante para a geração de um banco de conhecimento, ao passo que em um ambiente simulado, torna-se mais flexível a obtenção de situações críticas raras relevantes para a geração de um bom *baseline* acerca do problema abordado.

- Processamento não baseado em métodos analíticos

  Em geral, métodos analíticos utilizam parâmetros numéricos potencialmente imprecisos, o que torna o resultado deturpado em relação à situação que se deseja inferir. O uso de redes neurais treinadas com base em situações simuladas é capaz de garantir a precisão de parâmetros de entrada, uma vez que tais parâmetros serviram de base para a produção da fonte de conhecimento.

- Custo computacional de processamento reduzido

  Em problemas mais complexos, a ordem de complexidade dos métodos de solução usuais pode ter um custo computacional incompatível com os requisitos de tempo-real e com a alta velocidade das redes ATM. As redes neurais artificiais apresentam para tais problemas um custo computacional compatível com estes requisitos, devido a características intrínsecas a esta tecnologia.

- Maior facilidade de trabalhar com grande número de variáveis

  O número de variáveis de um problema pode representar uma limitação para métodos usuais. No caso de redes neurais, este fator não representa um acréscimo tão considerável

quanto em métodos analíticos, visto que se trata apenas de neurônios adicionais na camada de entrada.

- Possibilidade de implementação em *hardware*

  De acordo com [10] e [44], redes neurais do tipo mais comum (*feedforward*) são de fácil implementação em *hardware*, através da tecnologia VLSI.

- Possibilidade de integração com mecanismos de gerência convencionais

  A rede neural treinada, produto da fase de implementação da arquitetura RENATA, pode ser facilmente integrada a mecanismos de gerência convencionais, como o *Tivoli NetView* ou o *HP Openview* [25].

Verifica-se, também, algumas desvantagens do uso desta arquitetura, dentre as quais estão:

- Necessidade de extensa base de conhecimentos

  Dependendo do problema abordado e do número de variáveis envolvidas, a base de conhecimento necessária para a geração de um banco de exemplos de treinamento da rede neural pode demandar certa dificuldade de obtenção.

- Treinamento *off-line*

  O treinamento da rede neural que compõe a arquitetura RENATA é realizado de maneira *off-line*. Assim, a medida que novos casos até então desconhecidos pela experiência da rede neural ocorrem, a rede neural começa a se tornar ineficiente, sendo necessária a repetição do processo de treinamento com o novo ambiente.

- Custo computacional considerável para simulação e treinamento

  Dependendo da quantidade de conhecimento necessária para a resolução de um problema, o tempo necessário para geração da base de exemplos (simulação) e para o treinamento da rede neural pode tornar esta primeira fase muito onerosa.

- RNA não atinge resultados exatos

  Para problemas que necessitem de resultados exatos, a utilização de redes neurais não é a mais indicada. Apesar das redes neurais em geral conseguirem atingir resultados bem próximos do desejado, a filosofia desta tecnologia praticamente impossibilita a obtenção de resultados exatos.

Capítulo 6

Cenário de Experimentação

"The nice thing about standards
is that there are so many of them
to choose from."
– Andrew S. Tannenbaum

6.1 Introdução

A arquitetura RENATA é composta por diversas ferramentas integradas, dispostas como na sua arquitetura funcional (Figura 5.3). Algumas destas ferramentas foram adotadas dentre opções já implementadas e disponíveis em domínio público ou em modo "*Free Software*".

Outras ferramentas foram implementadas para concluir algumas funcionalidades da arquitetura e para realizar a integração entre os módulos que a compõem. Todas as ferramentas que compõem a arquitetura RENATA são descritas nas seções a seguir.

6.2 Simulador de Redes ATM

Um simulador de redes ATM foi escolhido a fim de compor parte do Módulo de Treinamento da arquitetura RENATA. Esta escolha levou em conta principalmente a flexibilidade de modelagem de aplicações e topologias e a facilidade de produção de *logs* acerca do processo simulado.

Assim, a ferramenta escolhida para este fim foi o simulador de redes ATM NIST. Esta ferramenta foi desenvolvida no NIST (*National Institute of Standards and Technology*) com o intuito de oferecer um ambiente de testes para estudo e análise de desempenho de redes ATM e HFC (*Hybrid Fiber Coax*). O simulador oferece ao usuário um ambiente interativo de modelagem baseado em interface gráfica e foi desenvolvido utilizando linguagem de programação C e o sistema *XWindows* sobre plataformas UNIX. Esta ferramenta, baseada em um simulador de redes desenvolvido no MIT (*Massachusetts Institute of Technology*) [23], oferece suporte a técnicas de simulação de eventos discretos e interface gráfica.

Este simulador permite que o usuário crie diferentes topologias de rede e atribua valores aos parâmetros de seus componentes, podendo salvar ou carregar configurações simuladas. Enquanto a simulação está sendo processada, várias medidas instantâneas de desempenho podem ser exibidas na forma texto/gráfico ou podem ser salvas em arquivos de *log* para análises posteriores.

O simulador ATM/HFC é uma ferramenta para analisar o comportamento de redes ATM e HFC sem o custo de construção de uma rede real. Existem duas utilizações principais para este simulador: como uma ferramenta para planejamento e dimensionamento de redes ATM ou como uma ferramenta de análise de desempenho de protocolos para redes ATM e HFC. Como um software de planejamento e dimensionamento, o projetista de redes pode simular várias configurações de rede e cargas de tráfego para obter estatísticas da utilização dos enlaces e o *throughput* dos circuitos virtuais. Experimentos deste tipo podem responder perguntas como: quando vai haver gargalos na rede planejada, qual é o efeito de mudar a capacidade de um enlace, se a adição de uma nova aplicação vai resultar em congestionamento, etc [18].

Como uma ferramenta de análise de protocolos, o pesquisador ou projetista de protocolo pode estudar o efeito total de um protocolo no sistema. Por exemplo, pode-se investigar a

eficácia de vários mecanismos de controle de fluxo para redes ATM e definir questões como mecanismos de alocação de largura de banda, *overhead* de protocolo, utilização de largura de banda, etc. Pode-se ainda estudar o desempenho de protocolos *Multiple Access* em redes HFC e a interoperabilidade de redes HFC com serviços ATM. Para que os experimentos possam ser conduzidos, uma investigação preliminar deve mudar ou adicionar código para implementar o protocolo a ser estudado. O simulador é projetado de maneira tal que componentes simulados podem ser facilmente modificados, adicionados ou removidos. Os eventos da rede podem, então, ser registrados célula a célula para análise posteriores.

A rede a ser simulada consiste de vários *componentes* enviando mensagens entre si. Os componentes disponíveis incluem Comutadores ATM, Equipamentos Terminais (B - TE's - *Broadband Terminal Equipments*), Rede HFC e Aplicações ATM. Comutadores e B-TE's são interconectados através de enlaces físicos, que também são considerados componentes. As aplicações ATM são entidades lógicas presas a um B-TE (*host*). As aplicações devem ser consideradas como geradores de tráfego capazes de emular fontes de tráfego constante ou variável. As aplicações ATM são conectadas entre si através de uma *rota*, que utiliza uma lista de componentes adjacentes para formar uma conexão virtual fim-a-fim. O componente HFC pode substituir o B-TE e simular um conjunto de *hosts* incluídos em um canal HFC compartilhado, sujeito a colisões.

Todos os componentes são caracterizados por um ou mais parâmetros. Estes parâmetros são divididos em duas categorias: de entrada e de saída. Ambos os tipos são listados em uma *janela informativa* que aparece próximo a cada componente quando o usuário desejar. Todos os parâmetros de entrada devem ser especificados pelo usuário no momento da criação do componente ou podem ser modificados um a um posteriormente. A atividade da rede pode ser observada através de janelas de metragem que exibem valores de parâmetros selecionados. Existem vários tipos de janelas de metragem disponíveis, que podem ser posicionadas em qualquer lugar da tela. As informações de parâmetros podem também ser armazenadas em arquivos especiais de *log*.

O usuário dispõe de várias aplicações cujos comportamentos determinam o tipo de tráfego

gerado para transmissão ao longo da rede. O usuário pode controlar os parâmetros associados a estes componentes, definindo rotas e especificando detalhes acerca do registro e/ou da apresentação das informações de desempenho. A interface ao usuário do simulador é apresentada na Figura 6.1. A tela exibe simultaneamente a configuração da rede, o painel de controle para a simulação em questão e informações de parâmetros.

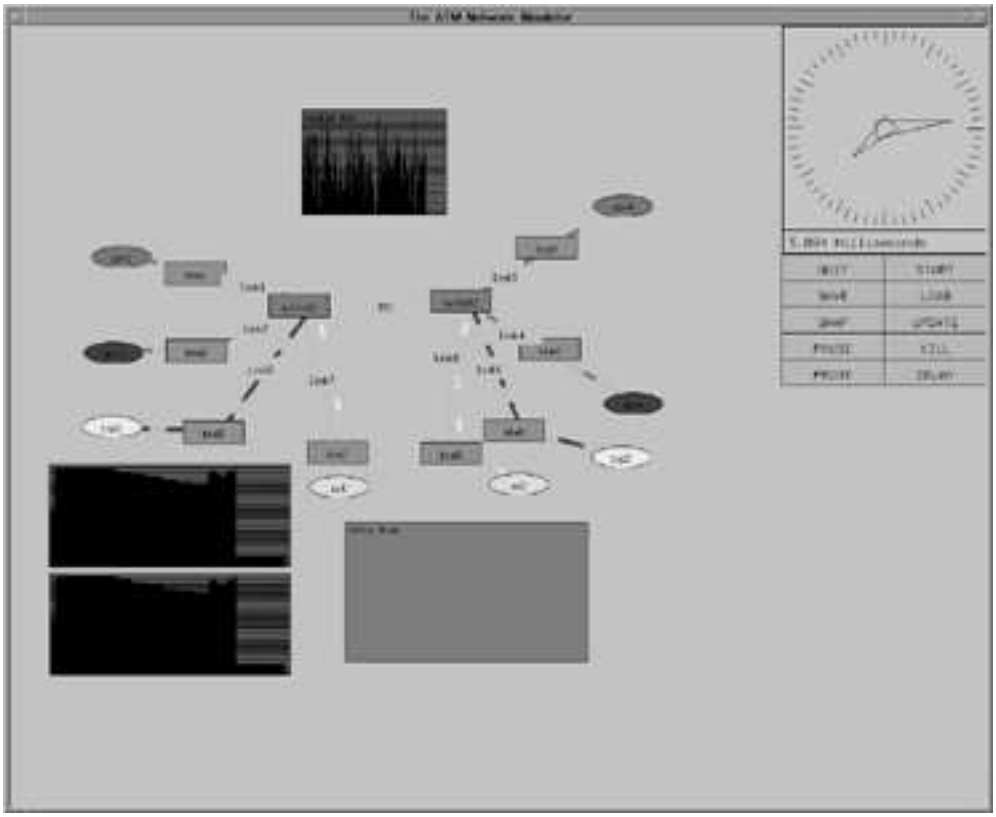


Figura 6.1: Tela do Simulador ATM NIST

Os componentes disponíveis no simulador ATM são brevemente descritos a seguir [18]:

- Comutador

  Este componente é usado para comutar ou rotear células através de vários enlaces de canal virtual. Quando um comutador aceita uma célula a partir de um enlace físico, sua tabela de rotas é consultada para determinar a qual enlace de saída a célula deve ser enviada. Se o enlace de saída estiver ocupado, o comutador deverá enfileirar as células destinadas a este

enlace e não pode mandá-las até que haja *slots* livres para a transmissão. O usuário pode especificar, dentre outras coisas, atraso de processamento de células, tamanho máximo da fila de saída e seus limites máximos. Os parâmetros que podem ser monitorados de um comutador incluem número de células recebidas, número de células na fila de saída, número de células descartadas e *status* dos indicadores de congestionamento.

- Equipamento Terminal de Faixa Larga (B-TE)

  Este componente simula um nó de uma rede RDSI-FL, *e.g.*, microcomputador, estação de trabalho, terminal de voz, etc. Um B-TE possui uma ou mais aplicações em um lado e um enlace físico no outro lado. Células que chegam no lado das aplicações são repassadas para o enlace físico. Se o enlace está ocupado, as células são enfileiradas. O usuário pode especificar o tamanho máximo da fila de saída. Os parâmetros que podem ser monitorados incluem o número de células na fila de saída e o número de células descartadas.

- Rede HFC (*Hybrid Fiber Coax*)

  O HFC é uma rede de TV a cabo (CATV) que utiliza tecnologias de cabo coaxial e fibra óptica para oferecer serviços digitais de alta velocidade para assinantes do serviço. Redes CATV são caracterizadas por uma topologia em árvore. Na raiz da árvore a central CATV difunde dados para os clientes e controla o tráfego nos canais no sentido dos usuários para a central. Neste simulador, um componente HFC simula uma rede HFC com topologia de tronco simples. Assim como um B-TE, o componente HFC é conectado a uma ou mais estações de usuários (ou aplicações) de um lado e a um enlace físico do outro lado.

- Aplicações ATM

  Este componente emula o comportamento de uma aplicação ATM em um ponto terminal de um enlace. Esta aplicação pode ser considerada como um gerador (fonte) de tráfego constante ou variável. No caso de aplicações do tipo CBR, o usuário deve especificar sua taxa de transmissão. Para aplicações do tipo VBR, o usuário deve especificar alguns parâmetros que definam seu comportamento, como tamanho de rajada, intervalo entre rajadas, etc. Para o tráfego de mais baixa prioridade, o usuário pode criar aplicações do tipo ABR. Para todos os tipos de aplicações, o usuário deve especificar o momento de início e o número de megabytes a ser enviado. Outros tipos de aplicações que podem ser simulados incluem TCP/IP, VBR MPEG e VBR auto-similar.

- Enlace Físico

  Este componente simula uma mídia física (cabo elétrico ou fibra óptica) no qual células são transmitidas. O usuário deve especificar a capacidade do enlace (velocidade) a partir de uma lista de vários tipos padrão. O usuário também especifica o comprimento do enlace. O parâmetro de saída reportado pelo simulador é a taxa de transmissão agregada instantânea no enlace em termos de bits por segundo (*Mbps*).

As aplicações simuladas podem produzir tráfego em três diferentes níveis de prioridade:

- Alta prioridade: fontes dos tipos CBR e VBR;
- Média prioridade: fontes do tipo ABR;
- Baixa prioridade: fontes do tipo UBR.

Para fontes de tráfego CBR, VBR e ABR, existem três tipos específicos de geradores de tráfego:

- Tráfego Constante

  Tipo de tráfego utilizado em aplicações do tipo CBR. Este tipo é especificado apenas por sua taxa de transmissão ao longo do tempo simulado.

- VBR - Poisson

  Este tipo fonte é caracterizado pelo tráfego ON–OFF. Os períodos ativo (ON) e inativo (OFF) são descritos a partir de uma distribuição exponencial. O usuário deve especificar neste caso o tamanho médio da rajada (tamanho médio do período ativo t^{on}), o intervalo médio entre rajadas (tamanho médio do período inativo t^{off}) e a taxa de transmissão da fonte em períodos ativos (*PCR*).

- VBR - Batch

  Para esta fonte de tráfego o usuário deve especificar o número médio de células que devem ser transmitidas durante uma rajada e o intervalo médio entre rajadas.

Para todos os tipos de tráfego, o usuário deve especificar o instante de início de operação e o número de *Mbits* que deve ser transmitido por esta fonte.

Outro tipo de aplicação ATM que pode ser simulada é a TCP/IP. Este tipo pode se utilizar tanto de serviço ABR quanto de serviço UBR. Outros dois tipos de aplicações também suportados são tráfego MPEG e tráfego auto-similar, apresentando ambos serviço VBR.

6.3 Simulador de Redes Neurais

A arquitetura RENATA necessita de uma ferramenta para suporte de redes neurais. Este programa deve oferecer meios para o projeto, treinamento e validação da rede neural que compõe o Módulo Neural da arquitetura. Esta ferramenta deve ser flexível o suficiente para que se possa modelar redes neurais para os mais diversos tipos de problemas que podem ser mapeados pela arquitetura RENATA.

O SNNS (*Stuttgart Neural Network Simulator*) é um simulador de redes neurais desenvolvido no Instituto de Sistemas Distribuídos e Paralelos de Alto Desempenho (*Institut für Parallele und Verteilte Höchstleistungsrechner*), na Universidade de Stuttgart, em 1989. O objetivo desta ferramenta é oferecer um ambiente de simulação flexível e eficiente para pesquisa e aplicações em redes neurais.

O simulador SNNS consiste de quatro componentes básicos, que estão ilustrados na Figura 6.2: o Kernel do simulador, interface gráfica ao usuário (GUI - *Graphical User Interface*), ferramenta `batchman` de simulação em lote e compilador de rede `snns2c`. O Kernel do simulador interage com as estruturas de dados internas da rede neural, realizando as operações necessárias para sua simulação. A interface gráfica localizada no topo do Kernel dá a representação gráfica das redes neurais e controla o Kernel durante a execução da simulação. Adicionalmente, a interface do usuário pode ser usada diretamente para criar, manipular e visualizar redes neurais de diversas formas. Assim, redes neurais mais complexas podem ser criadas de forma rápida e fácil.

A ferramenta `batchman` tem como objetivo realizar o treinamento em lote de diversas redes neurais sem que seja necessária a interação com o usuário. Este processo é realizado através da composição de um *script* que dá ao Kernel do simulador as diretivas sobre a escolha de topologias de rede, algoritmos de treinamento e funções de inicialização.

O módulo `snns2c` transforma o código da rede neural treinada (*i.e.* topologia e pesos) em um *stub* em linguagem C para que a funcionalidade desta rede treinada possa ser adicionada a sistemas de software. O código em linguagem C gerado é bastante simples (complexidade de $O(n^2)$) e contém todas as informações necessárias para que a rede neural treinada possa ser colocada em operação.

Um conceito de projeto importante para este simulador foi permitir que o usuário selecione apenas os aspectos de representação visual da rede neural que o interessa. Isto inclui o potencial de subdividir diversos aspectos da rede neural em várias janelas, assim como suprimir informações não solicitadas [56].

Dentre os tipos de arquiteturas de redes e procedimentos de aprendizagem contidos no simulador SNNS estão:

- *Backpropagation* (BP) para redes do tipo *feedforward*;
- *Counterpropagation*
- *Quickprop*

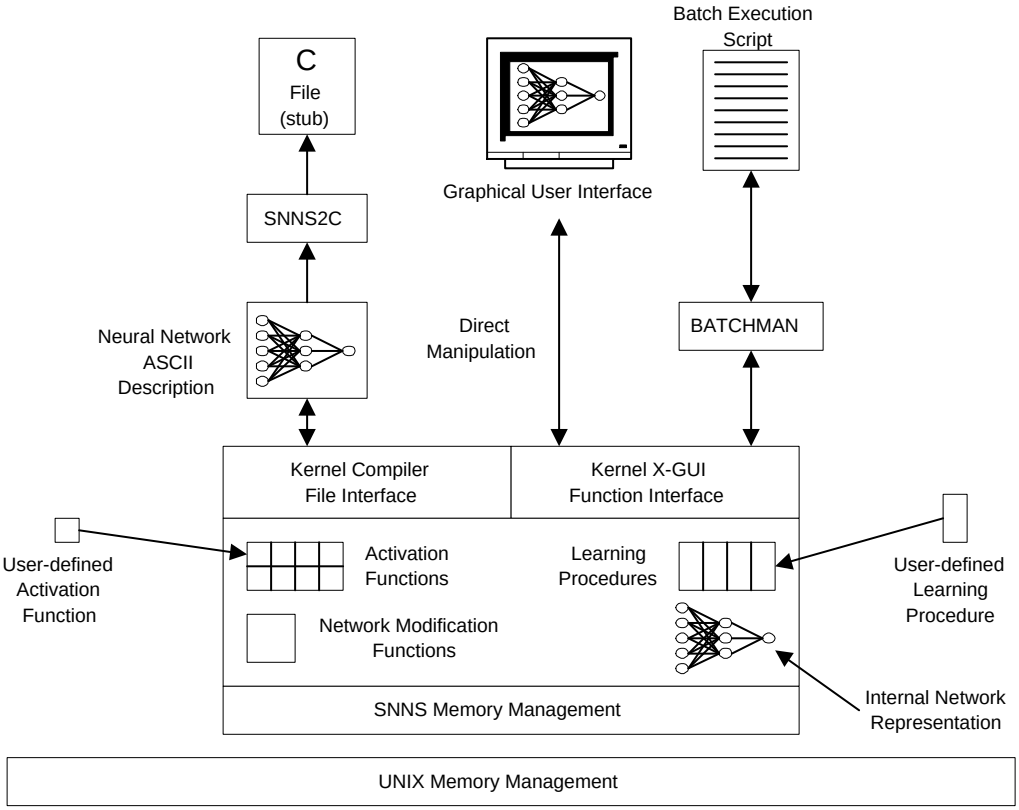


Figura 6.2: Arquitetura do SNNS

- *Backpercolation 1*
- *RProp*
- *Generalized Radial Basis Function* (RBF)
- ART1
- ART2
- ARTMAP
- *Cascade Correlation*
- *Recurrent Cascade Correlation*

- *Self-organizing Maps*
- *Jordan Networks*

A Figura 6.3 mostra o ambiente gráfico do simulador SNNS.

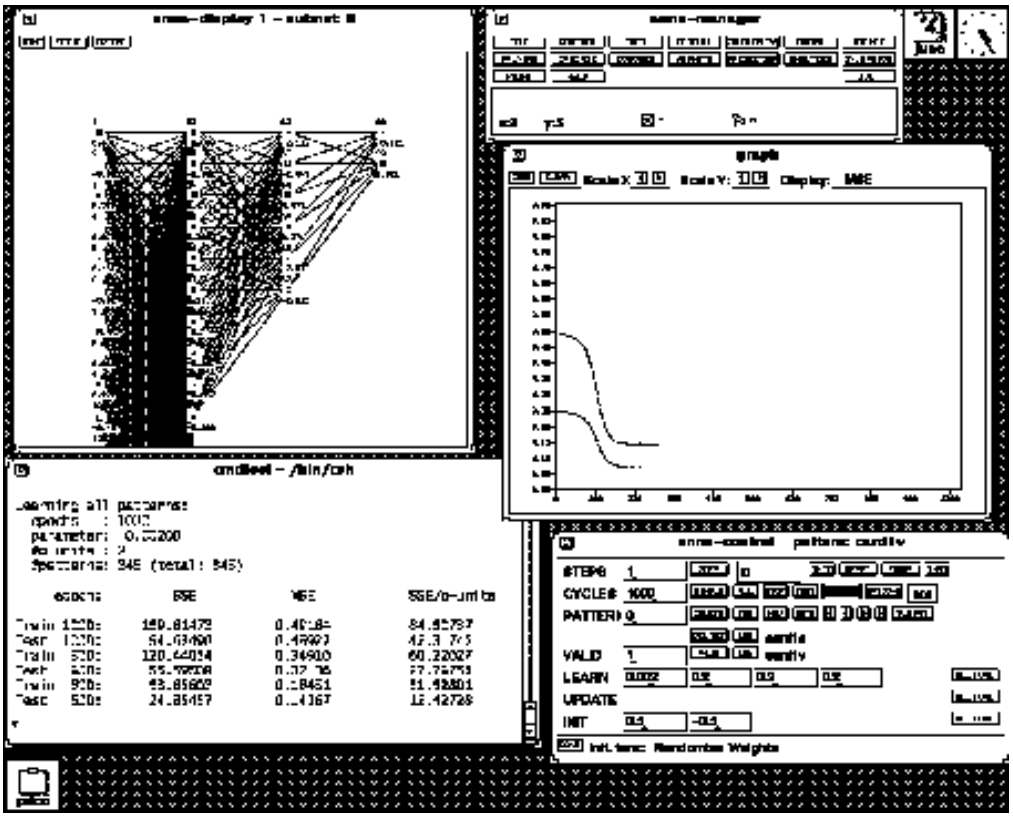


Figura 6.3: Tela do Simulador de Redes ATM SNNS

6.4 Ferramentas Implementadas

Algumas ferramentas complementares às citadas nas seções anteriores foram implementadas com o intuito de concretizar a arquitetura física da RENATA. A principal função destas ferramentas coadjuvantes é integrar as ferramentas que compõem a arquitetura funcional RENATA, automatizando e compatibilizando o processo de treinamento e operacionalização desta arquitetura, especificamente para o problema abordado neste trabalho.

Estas ferramentas incluem um Gerador de Perturbações, um Módulo de Seleção e Preparação de Dados (MSPD), um Agregador de Configurações e um Gerador de Dados Estatísticos. Cada ferramenta será brevemente descrita nas subseções a seguir.

6.4.1 Gerador de Perturbações

O Simulador de Redes ATM NIST é capaz de simular configurações formadas por componentes físicos, como comutadores, enlaces e B-TE's (*hosts*) e por componentes lógicos, representados pelas aplicações que são comutadas e trafegam pela topologia física simulada. Portanto, como o simulador ATM é capaz apenas de simular configurações com número estático aplicações, torna-se necessária uma ferramenta que seja capaz de gerar diversas configurações de forma a representar uma boa parte das possibilidades de composição de configurações que uma rede real é capaz de suportar, portanto dentro de sua área de atuação. Esta variedade é importante para que se possa gerar uma base de conhecimento com representatividade suficiente de toda a diversidade possível de configurações. A generalização para os demais casos é, então, responsabilidade da rede neural treinada a partir desta base de conhecimento.

A ferramenta `gera_conf` foi implementada com o intuito de gerar configurações com uma mesma topologia física, mas variando o número de aplicações aleatoriamente dentro de um intervalo especificado. Portanto, a função deste programa é gerar `N_CONF` configurações, onde cada uma é composta por uma quantidade entre `MIN_CONF` e `MAX_CONF` aplicações. Cada aplicação é caracterizada como VBR do tipo ON–OFF, tendo valores de t^{on} entre `MIN_TON` e `MAX_TON`, valores de t^{off} entre `MIN_TOFF` e `MAX_TOFF` e valores de PCR no intervalo entre `MIN_PCR` e `MAX_PCR`. Estes valores são randomicamente selecionados pela ferramenta `gera_conf`. Estas configurações são geradas na forma de arquivos de configuração compatíveis com o Simulador de Redes ATM NIST. Juntamente com este arquivo de configuração, um outro arquivo é produzido com informações acerca das características da configuração gerada. As configurações geradas são parametrizadas também por caracterizações gerais como as capacidades de enlaces de dados, capacidades de *buffers* de comutadores, etc.

6.4.2 MSPD

O Módulo de Seleção e Preparação de Dados (MSPD), explicitado no Capítulo 5, recolhe do arquivo de *log* de cada configuração simulada os dados brutos necessários para a criação de uma base de conhecimento. Estes dados são coletados dos *logs* de simulação, produzindo-se um arquivo contendo um ou mais exemplos relativos a cada configuração simulada. Estes dados são informações brutas para a obtenção da base de conhecimento que serve de entrada para treinamento e validação de uma rede neural.

A seleção dos parâmetros colhidos do *log* de simulação é definida nas Políticas de Monitoramento, citadas anteriormente. Alguns exemplos de ações do MSPD são coletar as taxas de transmissão agregadas instantâneas ao longo do tempo simulado, acompanhar a ocupação do *buffer* do comutador, verificar o número de células descartadas em cada nó, etc.

6.4.3 Agregador de Configurações

A ferramenta MSPD tem como responsabilidade criar, a partir de cada configuração simulada, um vetor de informações solicitado pelas Políticas de Monitoramento. A ferramenta `merge` pode ser considerada como um finalizador do trabalho do MSPD. Enquanto a ferramenta citada na seção anterior apenas colhe (seleciona) informações a partir de *logs* de simulação, gerando vetores de dados *brutos*, a ferramenta `merge` é responsável pela normalização dos dados brutos produzidos pelo MSPD e pela agregação de todos os vetores de informações normalizadas de todas as configurações em um único arquivo. Este arquivo representa a base de conhecimento completa, que pode ser utilizada para treinamento ou para validação da rede neural.

A Figura 6.4 ilustra o funcionamento das ferramentas `MSPD` e `merge`.

6.4.4 Gerador de Dados Estatísticos

O Gerador de Dados Estatísticos foi implementado com a finalidade de extrair informações acerca do processo de geração de exemplos e simulação da rede neural que compõe o processo de implementação da arquitetura RENATA.

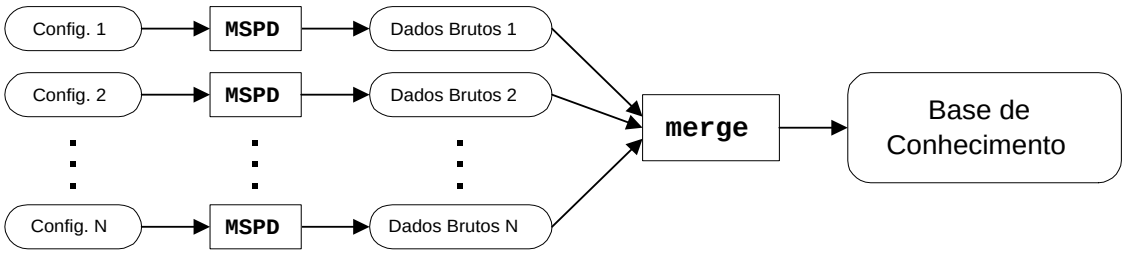


Figura 6.4: Funcionamento das Ferramentas `MSPD` e `merge`

As informações coletadas por esta ferramenta incluem o histograma sobre a distribuição de parâmetros que compõem os bancos de exemplos. Tais histogramas podem conter, por exemplo, informações sobre a distribuição do número de aplicações geradas nas configurações simuladas.

Outra função desta ferramenta é gerar um relatório final sobre o processo de treinamento de uma rede neural. Este relatório é utilizado para avaliar a eficácia dos resultados obtidos a partir da rede neural treinada em relação aos resultados desejados. É através deste relatório que é feita a medida da eficácia média de uma rede neural acerca de uma determinada amostragem de configurações. Por exemplo, este relatório pode fornecer meios para que se possa calcular a relação entre o número de aplicações em cada configuração e o erro obtido pela rede neural.

Capítulo 7

Prototipação

"Imagination is more important than knowledge.
Knowledge is limited. Imagination encircles the world."
– Albert Einstein

7.1 Introdução

A adaptação do Módulo de Treinamento da arquitetura RENATA ao problema abordado neste trabalho consiste em modelar uma rede neural capaz de realizar a estimativa da capacidade requerida do tráfego agregado em um comutador ATM. Os parâmetros utilizados como entrada para esta rede neural devem descrever o comportamento do tráfego agregado nos enlaces de entrada do comutador. As informações escolhidas para esta representação têm que captar fatores relevantes que possam ajudar a distinguir estados semelhantes em alguns aspectos, mas que diferem em alguma nuance de comportamento.

O banco de exemplos que servirá de base para o treinamento e validação da rede neural projetada é composto por um conjunto de valores de entrada (*inputs*) e um valor de saída (*output*). Os *inputs* representam as variáveis que descrevem o tráfego agregado de fontes VBR ON–OFF,

enquanto o *output* representa a capacidade requerida para este padrão de tráfego apresentado. Foram escolhidas algumas informações que podem ser obtidas a partir da simulação do comutador ATM de modo a representar o comportamento do tráfego agregado, levando em conta fatores como a explosividade do tráfego e o seu patamar de operação.

Na abordagem apresentada neste trabalho, o descritor de tráfego das aplicações é utilizado apenas para a composição dos *outputs* dos exemplos, sendo os *inputs* obtidos a partir da simulação das aplicações descritas. Desta forma, este tipo de estimativa não é afetado pela imprecisão dos descritores de tráfego, pois a classificação é gerada a partir do comportamento simulado destas aplicações. Assim sendo, o mapeamento entre os parâmetros de *input*, que caracterizam o comportamento do tráfego agregado, e a estimativa da capacidade requerida, obtida através de um método analítico, será realizada por uma Rede Neural Artificial. A escolha deste meio se deve à inexistência de um método analítico para esta função, e também por causa do requisito de tempo-real para esta classificação.

O processo de implementação do Módulo de Treinamento da arquitetura RENATA adaptada ao problema proposto consiste de quatro fases:

- Delimitação do Escopo de Atuação;
- Definição das Políticas de Monitoramento;
- Geração da Base de Conhecimentos;
- Projeto da Rede Neural.

Cada fase será descrita nas seções a seguir.

7.2 Escopo de Atuação

O primeiro passo para a implementação de uma rede neural é delimitar o seu escopo de atuação. Esta delimitação inclui a definição de intervalos de valores para variáveis e algumas generalizações que são assumidas de modo a facilitar a implementação. É sobre este escopo de atuação que

a base de conhecimentos é gerada. Portanto, a rede neural treinada a partir desta base é capaz de responder satisfatoriamente a situações dentro destes limites. Desta forma, mesmo a generalização realizada pela rede neural deve obter bons resultados apenas dentro deste escopo.

Algumas características gerais do sistema foram fixadas tendo como intuito a simplificação do problema e definição de seu escopo de atuação. Portanto, assume-se:

- O mesmo parâmetro de Qualidade de Serviço CLP (*Cell Loss Probability*) para todas as aplicações como $\varepsilon = 10^{-5}$;
- A existência de apenas conexões do tipo VBR ON-OFF;
- As fontes de tráfego têm tamanhos de estados (ON e OFF) distribuídos exponencialmente;
- Os intervalos entre chegadas de células de uma fonte de tráfego seguem a Distribuição de Poisson.

Foram definidos dois escopos de atuação distintos para experimentação (Escopos 1 e 2). Cada escopo serviu de base para a geração de uma amostra de configurações distintas acerca de uma mesma topologia ATM (Amostras 1 e 2). A simulação destas configurações, através do simulador de redes ATM do NIST [18], gerou para cada amostra um banco de exemplos (Bancos 1 e 2). Cada banco de exemplos foi subdividido em duas partes: o banco de treinamento e o banco de validação. Assim, foram gerados 4 bancos de exemplos: 2 bancos de treinamento (Amostras 1 e 2) e 2 bancos de validação (Amostras 1 e 2).

Os parâmetros que descrevem cada escopo de atuação definem limites para a quantidade e os descritores das aplicações a serem geradas randomicamente. Os valores definidos para cada escopo são descritos na Tabela 7.1.

Configurações são compostas randomicamente dentro dos limites de seu escopo de atuação e representam uma situação em que fontes de tráfego atingem um comutador através de seus

	PCR ($Mbps$)		t^{on} (ms)		t^{off} ($Mbps$)		N. Aplic.	
Escopo	Min	Max	Min	Max	Min	Max	Min	Max
1	0,1	3,00	0,01	2,5	0,01	2,5	5	100
2	0,1	2,75	0,01	2,5	0,01	2,5	20	80

Tabela 7.1: Definição de Escopos de Atuação

enlaces de entrada. Todas as fontes de tráfego simuladas são destinadas a um único enlace de saída. Assim, uma configuração do escopo 2 é formada por, no máximo, $N_{max} = 100$ aplicações (fontes de tráfego) do tipo VBR ON–OFF, cada qual com valores de *PCR*, t^{on} e t^{off} randomicamente gerados dentro dos intervalos supracitados.

Portanto, a topologia modelada para a extração de informações nos dois escopos de atuação se resume a um comutador que recebe um determinado número máximo de fontes de tráfego VBR ON–OFF através de enlaces de entrada que totalizam $L_{in} = 155,52\ Mbps$ e são comutadas com destino a um único enlace de saída de capacidade $L_{out} = 51,84\ Mbps$, servido por um *buffer* de capacidade ξ. Esta topologia é ilustrada na Figura 7.1.

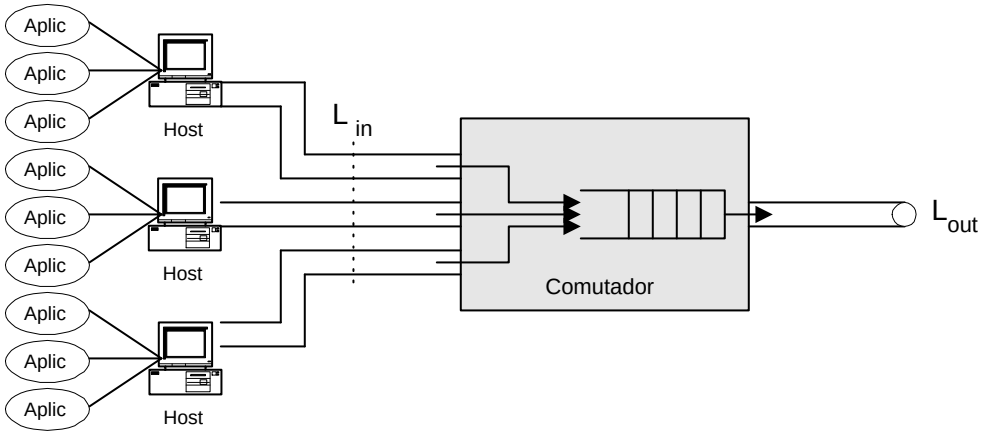


Figura 7.1: Topologia da Simulação

Cada rede neural projetada foi observada em situações de tamanhos de *buffer* variados. Estes valores de ξ foram definidos no intervalo de 50 a 1000 células.

7.3 Políticas de Monitoramento

O comportamento do tráfego agregado de fontes multiplexadas com destino a um mesmo enlace de saída, *i.e.* que compartilham a mesma fila de saída, pode ser diferenciado a partir da observação dos padrões de tráfego nos enlaces de entrada do comutador. Esta observação deve recair sobre dois principais fatores:

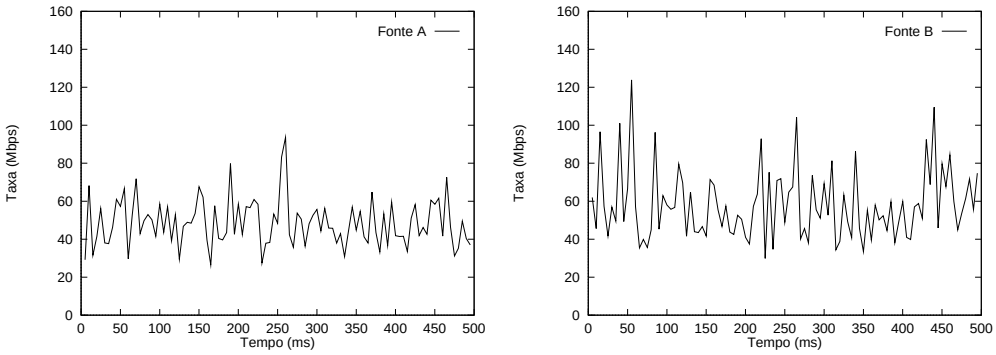
- Patamar de operação

  Este fator informa a qual nível de operação a rede se encontra com relação a taxa média agregada, taxa de pico agregada, etc. Esta informação deve definir o referencial de operação do comutador ao método de estimativa da capacidade requerida.

- Variabilidade de tráfego (explosividade)

  Esta medida representa a explosividade do tráfego agregado, isto é, quanto o tráfego varia de acordo com o tempo. Quanto maior for a explosividade de uma fonte de tráfego, mais distante é sua capacidade requerida de sua taxa média.

Estas informações representam dados estatísticos acerca do padrão que se deseja avaliar. Portanto, pode haver variação destes valores para uma mesma situação, dependendo do intervalo observado. Portanto, o método de estimativa da capacidade requerida baseada nestas informações deve ser capaz de generalizar para casos semelhantes a um determinado caso conhecido.



(a) Configuração A (b) Configuração B

Figura 7.2: Comparação entre Situações de Tráfego

Por exemplo, a Figura 7.2 apresenta duas situações de tráfego agregado, onde suas taxas instantâneas agregadas são expressas ao longo do tempo. A configuração A é gerada por 53 aplicações VBR ON–OFF que totalizam taxa de pico agregada de $\sum PCR = 80,8\ Mbps$. A situação de tráfego B é obtida a partir de 58 aplicações VBR ON–OFF que totalizam a mesma taxa de pico agregada $\sum PCR = 80,8\ Mbps$. Entretanto, a natureza geral das aplicações na configuração B apresenta uma explosividade maior do que na configuração A, *i.e.*, a variabilidade da configuração B é maior do que na configuração A, apesar de estarem praticamente no mesmo patamar de operação, por apresentarem a mesma taxa de pico agregada e número de aplicações aproximadas. Assim, a capacidade requerida na configuração A, calculada de acordo com o método EB para um comutador de tamanho de *buffer* $\xi = 1000$ células, é de $CR_A = 37,93\ Mbps$, enquanto a capacidade requerida da configuração B é de $CR_B = 44,55\ Mbps$. Isto mostra que o aspecto da variabilidade é tão importante quanto o patamar de operação para a capacidade requerida. Portanto, o mecanismo de estimativa desta capacidade requerida deve levar em conta estas duas nuances do comportamento do tráfego agregado.

O aspecto do patamar de operação de uma situação de tráfego pode ser distinguido se houver uma maneira de se estimar uma média sobre a taxa de transmissão agregada ao longo de um determinado intervalo de tempo. Outra variável que pode ajudar a definir este patamar é o somatório das taxas de pico de todas as aplicações envolvidas na configuração, o que define

o teto de operação. É possível também extrair uma idéia da variabilidade do tráfego agregado através do cálculo do *desvio padrão* de sua taxa agregada ao longo de um determinado intervalo de tempo. Por exemplo, no caso da Figura 7.2, as médias das taxas de transmissão agregadas das configurações A e B calculadas durante um intervalo de 45 *ms* são $SBR_A = 51,37\ Mbps$ e $SBR_B = 56,55\ Mbps$, respectivamente. O desvio padrão das taxas de transmissão agregadas destas configurações A e B calculadas no mesmo intervalo são $\sigma_A = 14,62\ Mbps$ e $\sigma_B = 17,46\ Mbps$, respectivamente. Através destes valores, o mecanismo de estimativa da capacidade requerida pode estimar as duas nuances do comportamento do tráfego agregado citadas anteriormente.

Duas variáveis foram escolhidas para oferecer um reforço de especificação às configurações. São elas a taxa agregada de pico ($\sum PCR$) e o número de aplicações envolvidas na configuração. A primeira variável reforça a idéia do patamar de operação, visto que quanto maior o somatório das taxas de pico das aplicações, maior é o patamar de operação esperado. Enquanto isto, a outra variável reforça a idéia da variabilidade da taxa de transmissão da configuração, visto que, de modo geral, quanto maior o número de aplicações envolvidas na configuração, menor será a sua variabilidade.

Para oferecer uma maior robustez à informação fornecida como base de conhecimento, escolheu-se coletar médias e desvios padrão em diferentes bases de tempo, sendo então o intervalo escolhido observado em diferentes prazos.

Portanto, cabe ao MSPD (Módulo de Seleção e Preparação de Dados) colher informações brutas a partir do *log* de simulação produzido de cada configuração. O MSPD deve, então, receber como entrada os arquivos de *log* produzidos pelo simulador de redes ATM e selecionar randomicamente intervalos ao longo do tempo de simulação para a coleta dos dados que, quando processados, comporão os vetores de entrada correspondentes. Uma configuração pode gerar mais de um exemplo. No caso desta experimentação, escolheu-se extrair apenas um exemplo de cada configuração.

7.4 Geração da Base de Conhecimento

Para a composição do banco de exemplos de treinamento e de validação para os escopos definidos, foram geradas para cada escopo 4 000 variedades de configuração sobre uma mesma topologia ATM. Todas as configurações geradas foram simuladas utilizando como ferramenta o simulador NIST [18]. Cada configuração de rede é simulada por um período virtual de 1 *s*. O tempo médio de processamento de simulação para extração dos *logs* de 1 *s* de operação em 4000 configurações é de, em média, 20 horas em um supercomputador IBM SP-2 com 4 nós (CENAPAD-NE).

Em seguida, os *logs* produzidos pelas configurações simuladas foram alimentados ao MSPD. Em primeiro estágio, o MSPD seleciona um intervalo entre o tempo total simulado, denominado *história* e extrai deste período as informações necessárias, que mostram a evolução da taxa de transmissão agregada instantânea ao longo deste intervalo. Estes dados são processados e é criado um vetor com as informações brutas de cada configuração.

Os valores que compõem este vetor incluem:

- Número de fontes de tráfego multiplexadas;
- Somatório das taxas de pico destas aplicações;
- Série temporal da taxa de transmissão agregada média e de seu desvio padrão ao longo do intervalo definido.

Os valores acima representam as informações brutas para entrada da rede neural. Entretanto, a sua saída desejada não é obtida a partir do *log* de simulação. O valor da capacidade requerida para a referida configuração é obtido da aplicação do método analítico EB sobre os valores de descritores de tráfego que serviram de base para a geração do ambiente simulado. Desta forma, tem-se a segurança de que as medidas de tráfego agregado colhidas nos *logs* de simulação realmente representam o comportamento gerado a partir destes descritores. Portanto, a principal função da rede neural é mapear estes dois universos: o descritor e seu comportamento gerado através de simulação. A Figura 7.3 ilustra o processo de geração do banco de exemplos.

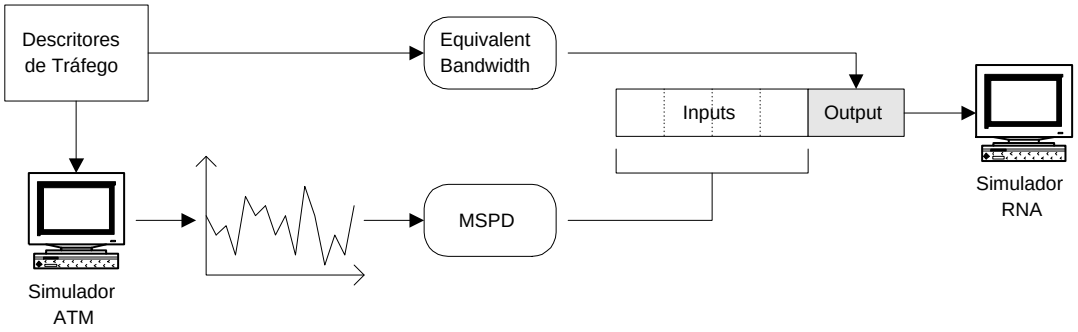


Figura 7.3: Esquema de Geração do Banco de Exemplos

Na segunda fase de operação do MSPD, os valores brutos produzidos são normalizados e dispostos em um arquivo de exemplos compatível com o simulador de redes neurais SNNS.

7.5 Projeto da Rede Neural

Uma das fases mais relevantes do projeto de uma rede neural é a definição das variáveis dos vetores de entrada e de saída da rede neural. Para cada problema, deve existir um conjunto de valores que represente o mais fielmente possível o estado real que se deseja classificar ou inferir sobre. Portanto, para que uma rede neural possa interpolar uma função o mais confiável possível, deve-se definir suas variáveis e representá-las no vetor de entrada da RN, com os respectivos valores desejados de saída [24].

O tipo de rede neural escolhida para este experimento é a *feed-forward*, utilizando para o processo de treinamento o algoritmo *Backpropagation Momentum* [70], que é uma otimização do método *Backpropagation* que inclui mecanismos que desconsideram mínimos locais. A topologia escolhida consiste de 3 camadas de neurônios (entrada, intermediária e saída). O número de neurônios na entrada varia em função do parâmetro história ($2 \times (h+1)$). A camada de saída é fixa em 1 neurônio, representando o valor da Capacidade Requerida normalizada. Fez-se variar o número de neurônios na camada intermediária em busca de uma maior precisão.

Para o problema proposto, considera-se um comutador caracterizado por sua capacidade máxima de conexões N_{max} e por sua capacidade de buffer ξ (Figura 7.4). Em um determinado momento, este equipamento multiplexa N fontes de tráfego advindas de enlaces de entrada, os quais totalizam uma capacidade de L_{in}. Todo o tráfego destas fontes deverão ser roteadas para um mesmo enlace de saída de capacidade L_{out}. Tais aplicações (fontes) são caracterizadas pelo tráfego VBR ON-OFF. O processo de chegada de células destas aplicações segue o Modelo Estocástico de Poisson. Conseqüentemente, os tamanhos de cada estado (ON e OFF) seguem uma distribuição exponencial. Cada fonte de tráfego n é caracterizada por sua taxa de transmissão de pico P_n e pelos seus tamanhos médios dos períodos ativo (t^{on}) e inativo (t^{off}), com $n \leq N \leq N_{max}$, sendo que o valor máximo para a taxa de transmissão de pico de aplicações é representada por P_{max}.

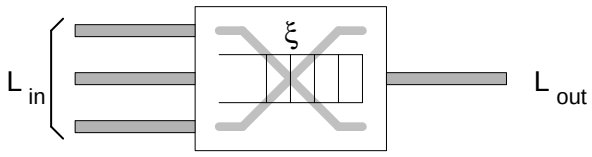


Figura 7.4: Parâmetros do Comutador

A rede neural proposta utiliza informações obtidas a partir de leituras sobre a taxa de transmissão do tráfego agregado em momentos anteriores. A partir destas informações, a rede neural deverá determinar a capacidade requerida agregada no instante presente, denotado por T_0.

Ao longo do tempo, define-se *Pontos de Medição* (*MP′s*) e *Pontos de Checagem* (CP's). Os MP's marcam os instantes onde as leituras acerca da taxa de transmissão agregada são realizadas. Estes pontos distam entre si em $\Delta\tau$. Os CP's são os MP's onde todas as informações são totalizadas. Os CP's são eqüidistantes entre si em ω intervalos $\Delta\tau$, isto é, entre dois CP's existem $(\omega - 1)$ MP's. Os valores coletados nos MP's são totalizados e processados no CP subseqüente. A distribuições de MP's e CP's ao longo do tempo é ilustrada na Figura 7.5.

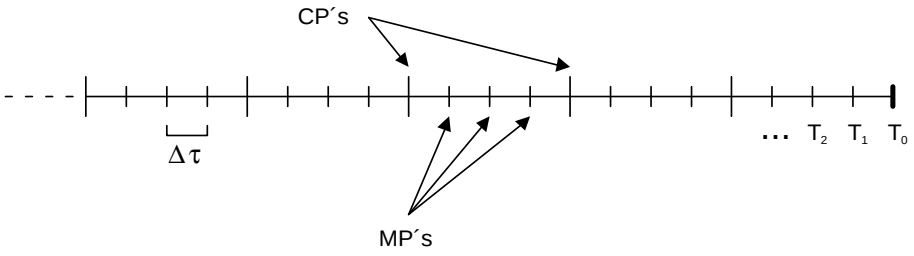


Figura 7.5: Diagrama de Pontos de Checagem e Medição

Denomina-se o instante presente como T_0, e os instantes anteriores, eqüidistantes entre si, como T_i, com $i = 0, 1, \cdots$. Assim:

$$T_i - T_{i+1} = \Delta\tau \quad \text{para qualquer } i \geq 0$$

Define-se *história* como sendo a medida do tempo durante o qual o sistema é observado para que a rede neural possa coletar as informações suficientes para compor um vetor de entrada. Este período pode ser caracterizado pela quantidade de CP's que o compõe. Portanto, se o intervalo entre dois CP's é de $\Delta\tau$, e o vetor de entrada da RN contém informações de h CP's, então o tráfego deverá ser observado e medido por um período de $h \times \omega \times \Delta T$ para que seja composto um vetor de entrada para a rede neural.

Seja R_t^j a taxa de transferência instantânea da aplicação j no momento T_t. Portanto, sendo C_δ^j o número de bits transmitidos pela fonte de tráfego j durante o intervalo δ, temos:

$$R_t^j = \lim_{\delta \to 0} \frac{C_\delta^j}{\delta}$$

Então, a *taxa de transmissão agregada instantânea* que chega ao comutador no momento T_t é expressa como:

$$R_t = \sum_{j=1}^{N} R_t^j$$

N é o número de aplicações que utilizam o comutador para atingir um único enlace de saída.

Em cada MP, mede-se a taxa de transferência agregada instantânea e em cada CP os últimos ω valores medidos – incluindo o valor medido no próprio CP – são totalizados e processados.

Portanto, em um instante T_m onde $m \ mod \ \omega = 0$, isto é, o momento T_m é um CP, define-se σ_t como sendo o *desvio padrão* da taxa de transmissão agregada R_t, com t variando entre $[0; m]$.

$$\sigma_m = \sqrt{\frac{\sum_{t=0}^{m} (\bar{R_m} - R_t)^2}{m-1}}$$

onde $\bar{R_m}$ é a média das taxas de transmissão agregadas instantâneas nos momentos entre $[0; m]$.

$$\bar{R_m} = \frac{\sum_{t=0}^{m} R_t}{m}$$

Baseado nestes valores, define-se as seguintes funções:

$$f(t) = \frac{2 \times \bar{R_t}}{\sum P_{max}} \qquad\qquad g(t) = \frac{4 \times \sigma_t}{\sum P_{max}}$$

Onde $\sum P_{max} = P_{max} \times N_{max}$

Tais valores e funções foram definidos com o intuito de normalizar o vetor de dados brutos para que dados compatíveis com a topologia e com o algoritmo de aprendizagem da rede neural projetada.

Assim sendo, os valores selecionados para compor o vetor I de entrada da rede neural são:

$$I = \left(\frac{N}{N_{max}}, \frac{\sum P_j}{\sum P_{max}}, \overbrace{f(1\times\omega), g(1\times\omega), \cdots, f(h\times\omega), g(h\times\omega)}^{h\times} \right)$$

O processo de normalização foi aplicado aos valores brutos, tendo como objetivo colocar os valores que compõem o banco de exemplos para treinamento e validação da rede neural no intervalo $[-1;1]$. Esta medida tem como objetivo possibilitar e/ou facilitar a convergência no processo de treinamento da rede neural. Os fatores de normalização para os parâmetros que compõem o banco de exemplos para o problema proposto são descritos na Tabela 7.2.

Parâmetro	Variável	Fator de Normalização
Número de Conexões	N	N_{max}
Somatório das Taxas de Pico	$\sum PCR$	$\sum P_{max}$

Tabela 7.2: Fatores de Normalização

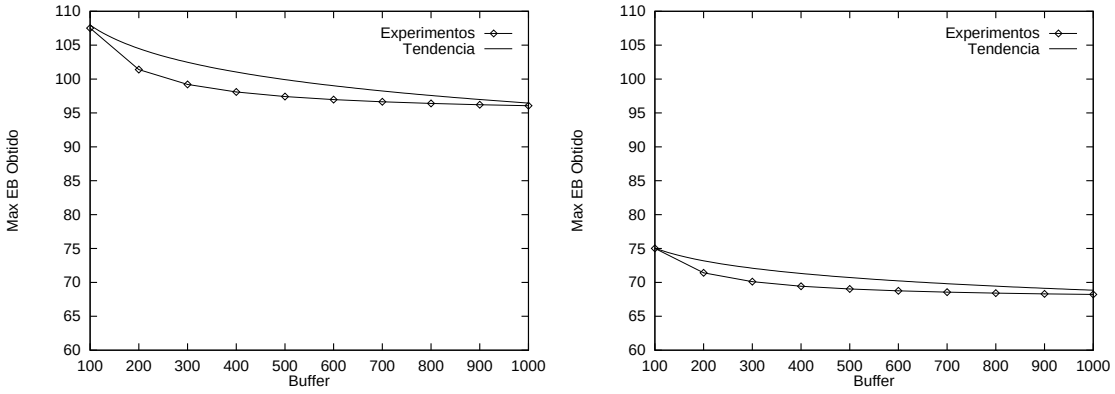
O valor da classificação, representado pelo vetor de saída com uma posição, é obtido a partir da aplicação do método EB (Equação 4.1) nos descritores de tráfego utilizados para a simulação. Portanto, o vetor de saída O é definido como:

$$O = \left(\frac{\sum_{j=1}^{N} EB_j}{\gamma} \right)$$

onde γ é o fator de normalização para a capacidade requerida agregada, dado por:

$$\gamma = \begin{cases} -5,0 \times ln(\xi) + 131,0 & \text{(Amostra 1)} \\ -2,7 \times ln(\xi) + 87,5 & \text{(Amostra 2)} \end{cases}$$

As funções de normalização definidas têm como objetivo colocar os valores das capacidades requeridas obtidas em intervalo [0;1]. Estas funções foram obtidas a partir de regressão logarítmica aplicada a valores obtidos de um número razoável de experimentações em cada amostra. A Figura 7.6 mostra os valores obtidos por cada amostra experimentada e as curvas de tendência para estes valores.



(a) Amostra 1 (b) Amostra 2

Figura 7.6: Regressão Logarítmica para Normalização do *Output*

A Figura 7.7 ilustra a rede neural projetada.

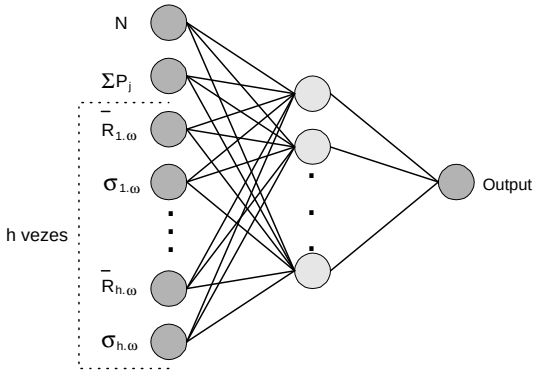


Figura 7.7: Parâmetros de Entrada da Rede Neural

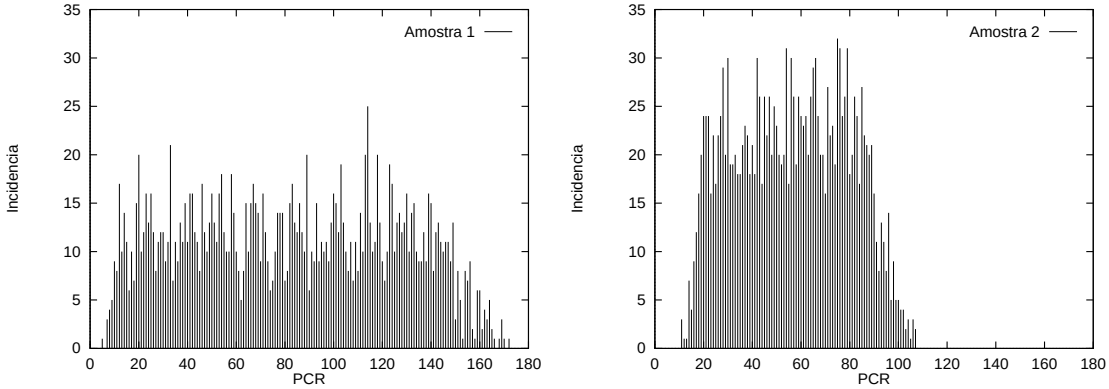
Capítulo 8

Análise dos Resultados

"When people agree with me
I aways feel that I must be wrong."
– Oscar Wilde

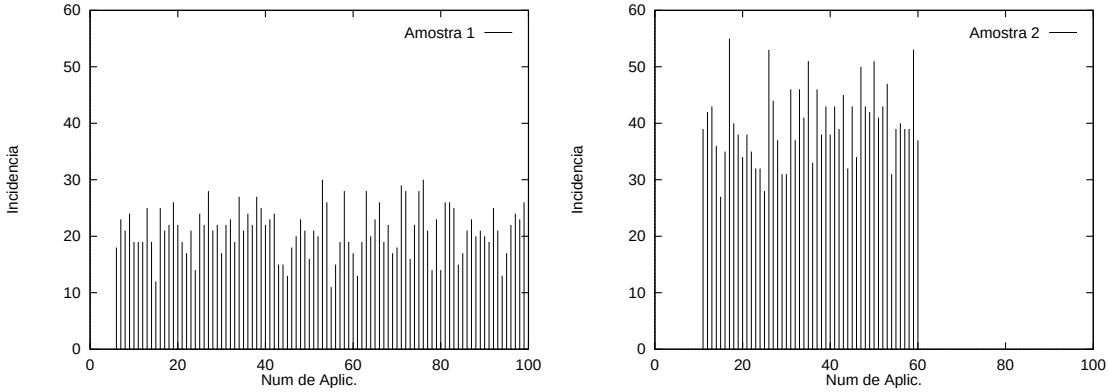
Os bancos de exemplos gerados a partir dos escopos de atuação definidos no capítulo anterior serviram de base para a experimentação descrita neste trabalho. Assim, foram gerados dois bancos de exemplos distintos, cada um gerado a partir de um escopo definido. Cada banco de exemplos foi gerado a partir de um conjunto de 4 000 configurações simuladas. Estas configurações foram geradas randomicamente a partir dos limites descritos em cada escopo de atuação. A incidência de casos em cada uma das duas amostras geradas está ilistrada na Figura 8.1 e 8.2. As Figuras 8.1(a) e 8.1(b) mostram a incidência quanto ao somatório das taxas de pico das aplicações ($\sum PCR$), enquanto as Figuras 8.2(a) e 8.2(b) mostram a incidência levando-se em conta o número de aplicações envolvidas ($N.Aplic$) em cada configuração.

Cada banco de exemplos foi subdividido em duas partes, sendo uma utilizada para o treinamento da rede neural (*banco de treinamento*) e outra utilizada para a validação da rede neural (*banco de validação*). Cada subconjunto é composto por 2 000 exemplos. Esta organização está ilustrada na Figura 8.3.



(a) $\sum PCR$ para Amostra 1 (b) $\sum PCR$ para Amostra 2

Figura 8.1: Histograma das Amostras - $\sum PCR$



(a) *N.Aplic* para Amostra 1 (b) *N.Aplic* para Amostra 2

Figura 8.2: Histogramas das Amostras - *N.Aplic*

Adicionalmente a esta primeira divisão, uma outra subdivisão lógica foi realizada nos bancos de exemplos, com a finalidade de testar a eficácia de uma rede neural treinada a partir de diferentes seleções de exemplos.

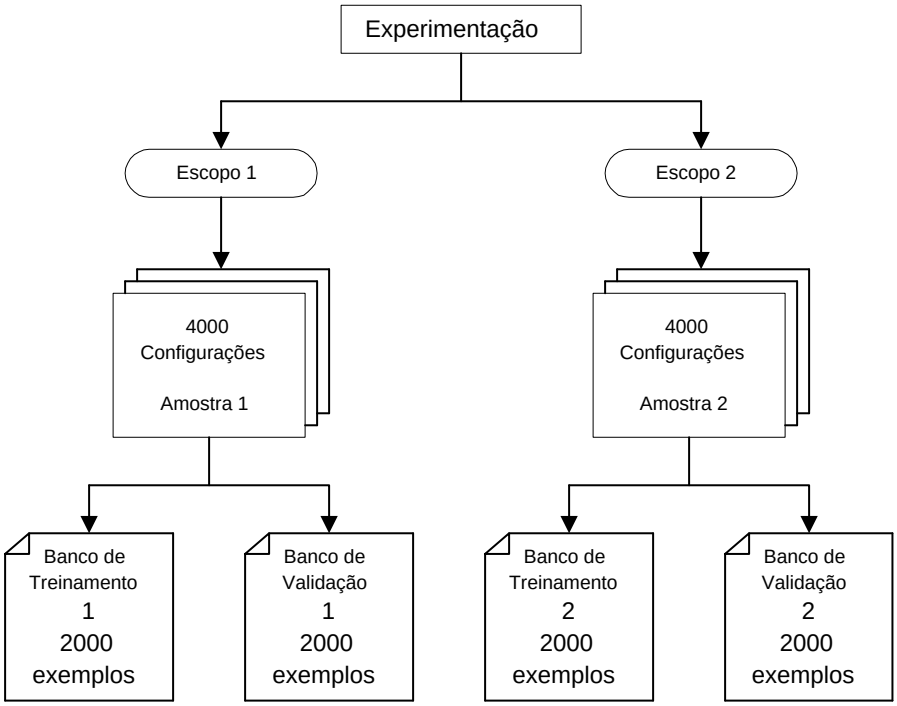


Figura 8.3: Divisão dos Bancos de Exemplos

As situações geradas em cada base de dados são as seguintes:

- Situação 1:

  Seleção de todos os exemplos gerados;

- Situação 2

  Seleção dos exemplos cujo somatório das taxas de pico das aplicações ($\sum PCR$) seja superior ou igual à capacidade do enlace de saída (L_{out});

- Situação 3

  Seleção dos exemplos cujo número de aplicações seja superior a 20;

- Situação 4

  Seleção dos bancos de exemplos cujas capacidades requeridas estejam distantes do enlace

de saída (L_{out}) em 40% de L_{out}, isto é, configurações cujas capacidades requeridas sejam maiores do que $0,4 \times L_{out}$ e menores do que $1,4 \times L_{out}$.

Os números resultantes de exemplos de cada situação após estes filtros estão mostrados na Tabela 8.1.

		Situação 1	Situação 2	Situação 3	Situação 4
Amostra 1	Treinamento	2000	1407	1685	999
	Validação	2000	1433	1701	1033
Amostra 2	Treinamento	2000	1018	1458	811
	Validação	2000	1282	1733	997

Tabela 8.1: Distribuição das Situações Geradas

Todos os resultados dos experimentos realizados correspondem aos valores obtidos no processo de validação da rede neural, isto é, o teste é feito sempre a partir de configurações que não foram apresentadas à rede neural no processo de treinamento.

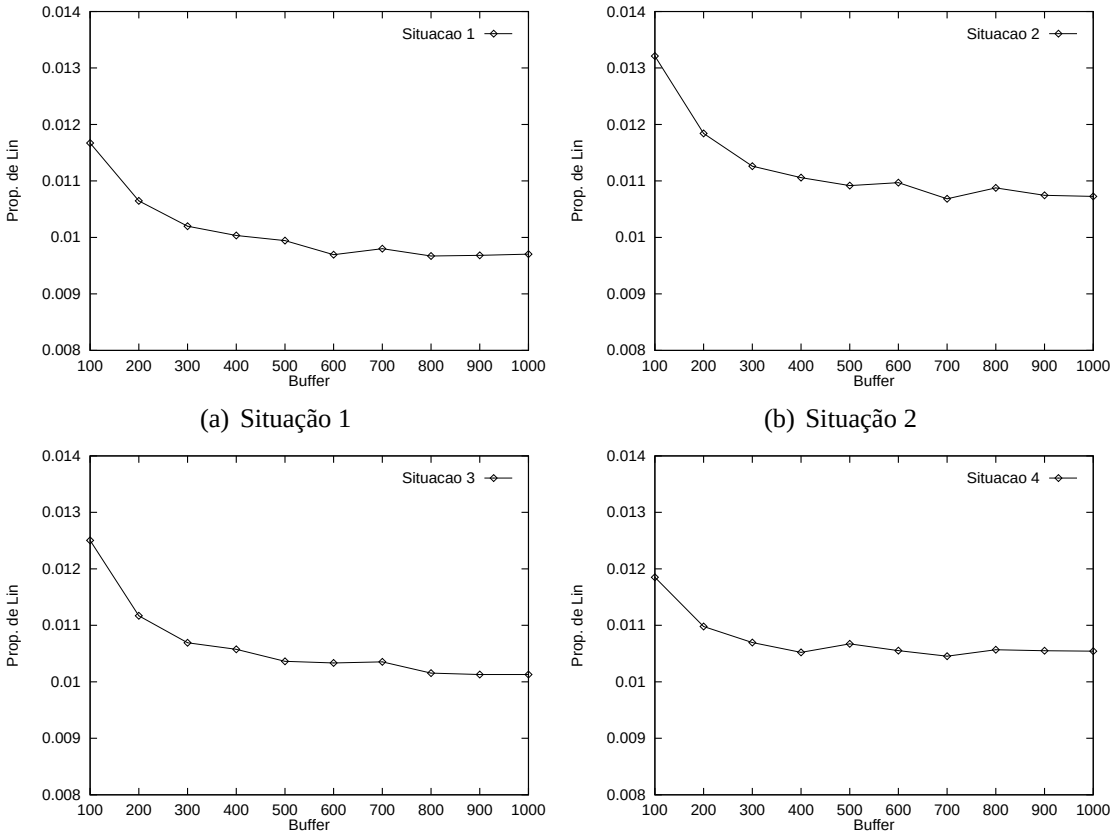
O primeiro experimento realizado mostra a taxa de erro obtida no treinamento de uma rede neural com as diferentes situações definidas. A acurácia dos resultados neste experimento foi expressa em função da diferença média entre o valor da capacidade requerida desejada e obtida pela rede neural, proporcionalmente à capacidade total dos enlaces de entrada ($L_{in} = 155,52\ Mbps$). Assim, definiu-se uma função denominada *Erro Absoluto* como sendo:

$$E_{abs} = \frac{\sum |EB_{des} - EB_{obt}|}{L_{in} \times N}$$

Os resultados dos experimentos são apresentados nas figuras a seguir. As Figuras 8.4(a), 8.4(b), 8.4(c) e 8.4(d) mostram o E_{abs} obtido no treinamento das quatro situações da Amostra 1 em função do tamanho do *buffer* utilizado no comutador. Observou-se que em todos os casos a taxa de erro decresce a medida que o tamanho do *buffer* aumenta.

Da mesma maneira, as Figuras 8.5(a), 8.5(b), 8.5(c) e 8.5(d) mostram o E_{abs} obtido para as mesmas situações definidas na Amostra 2.

As Figuras 8.6(a) e 8.6(b) mostram a comparação dos valores do E_{abs} entre as situações definidas nas Amostras 1 e 2, respectivamente.



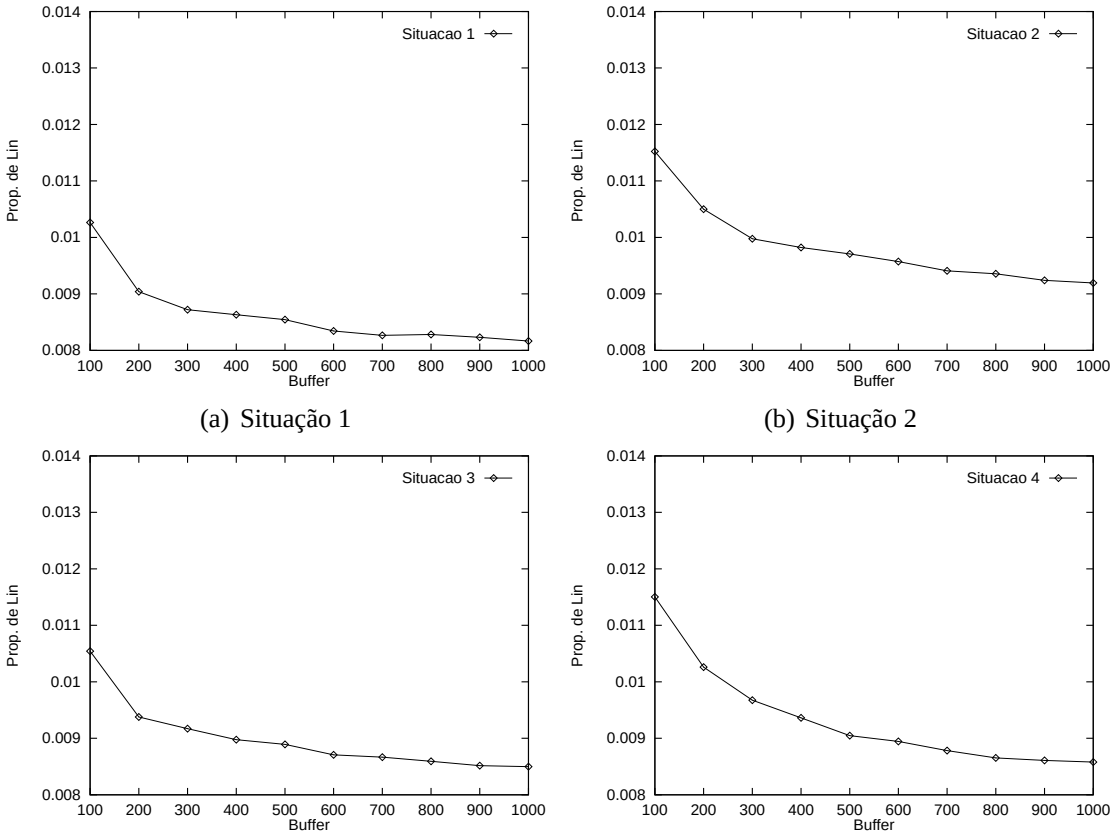
(a) Situação 1 (b) Situação 2

(c) Situação 3 (d) Situação 4

Figura 8.4: Valores de E_{abs} para Amostra 1

Observa-se que a Situação 1 em ambas as amostras apresentam margem de erro menor do que as outras. Isto se deve principalmente à disponibilidade de uma variedade de exemplos mais ampla, que propicia o aprendizado da rede neural. A Situação 3 apresenta a segunda melhor precisão, enquanto a situação 2 apresenta a maior margem de erro. Observa-se também que no pior caso, o Erro Absoluto não passa de 0,014, representando, em outras palavras, que a diferença entre o valor obtido e o valor desejado é em média inferior a 1,2% da capacidade dos enlaces de entrada (L_{in}). Se tomarmos como base o valor de L_{in} definido para esta experimentação ($L_{in} = 155,52\ Mbps$), a diferença média entre os valores de EB desejado e obtido é, no pior caso, de 2,17 Mbps. Considerando que podem haver até 100 aplicações comutadas em um determinado momento, tem-se que o erro médio para cada aplicação não passa de 0,02 Mbps.

Notou-se também que a precisão de uma rede neural treinada a partir da Amostra 2 é geral-



(a) Situação 1 (b) Situação 2

(c) Situação 3 (d) Situação 4

Figura 8.5: Valores de E_{abs} para Amostra 2

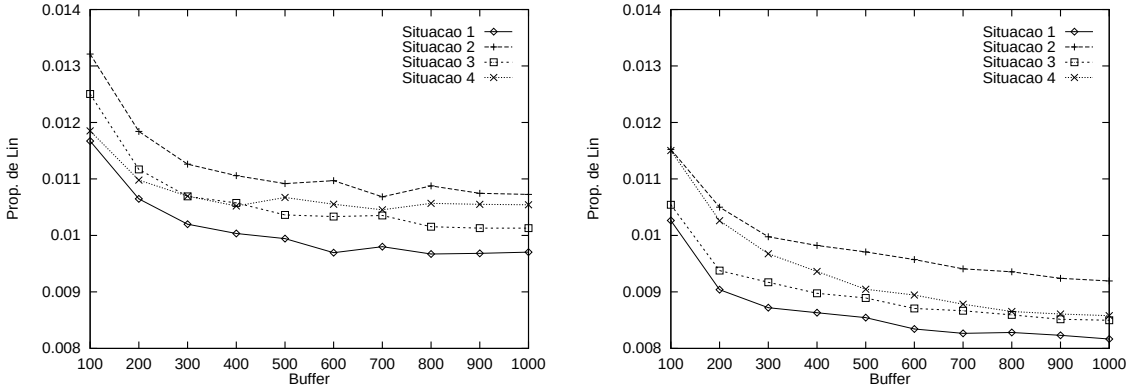
mente mais precisa do que se tivesse sido treinada a partir da Amostra 1.

Um outro experimento testou a influência do número de aplicações comutadas na precisão obtida pela rede neural treinada a partir da Amostra 1. Os resultados apresentados na Figura 8.7 mostram a precisão da rede neural expressa através do *Erro Médio Quadrado* (MSE - *Mean Square Error*) em função do número de aplicações comutadas na configuração. A medida do MSE é obtido da seguinte maneira:

$$MSE = \frac{\sum_{i=1}^{N}(o_{des} - o_{obt})^2}{N}$$

onde o_{des} e o_{obt} representam as saídas desejada e obtida da rede neural.

Neste experimento, observa-se que a taxa de erro média cresce à medida que o número de aplicações envolvidas na configuração cresce. Isto se deve ao fato de que quanto maior o número



(a) Amostra 1 (b) Amostra 2

Figura 8.6: Comparação entre Amostras 1 e 2

de aplicações envolvidas, maior é a sua explosividade, o que torna mais difícil a classificação.

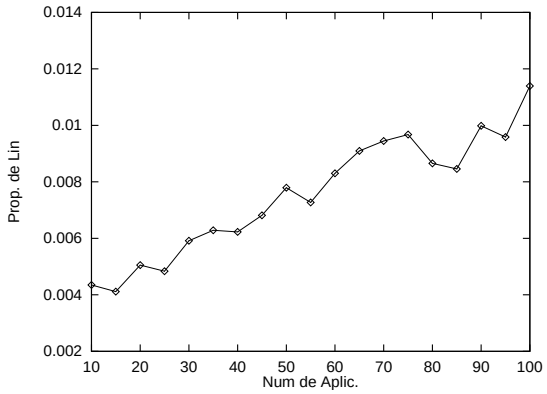
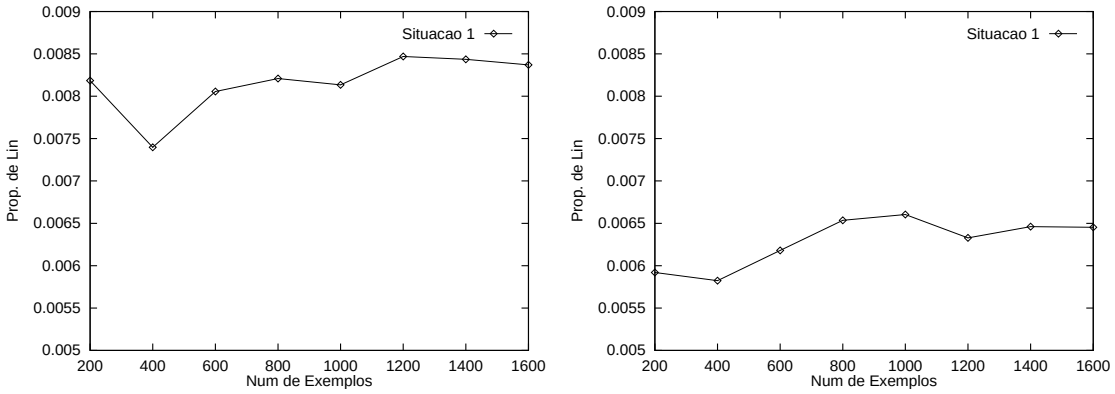


Figura 8.7: Número de Aplicações $\times$ MSE

Foi experimentada também a influência do tamanho do banco de treinamento para a precisão da rede neural. Os resultados obtidos para este experimento aplicado à Amostra 1 são exibidos na Figura 8.8(a), que mostra valores de E_{abs} em função do número de exemplos utilizados no banco de treinamento da rede neural. Observou-se que a precisão da rede neural oscila não-monotonicamente em função do número de exemplos de treinamento utilizado. Este mesmo comportamento é observado nos resultados da experimentação sobre a Amostra 2, mostrados na Figura 8.8(b).

Foi testada também a influência do número de neurônios ocultos na camada intermediária da rede neural *feedforward* na precisão dos resultados obtidos pelo treinamento. Estes resulta-



(a) Amostra 1 (b) Amostra 2

Figura 8.8: Número de Exemplos de Treinamento $\times$ E_{abs}

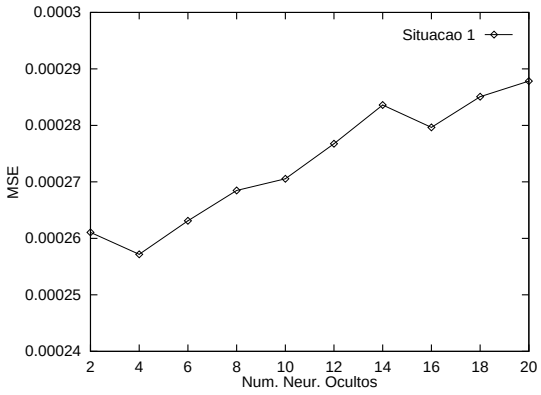


Figura 8.9: Número de Neurônios Ocultos $\times$ MSE

dos são exibidos na Figura 8.9. Notou-se que o valor de MSE varia não-monotonicamente de acordo com o número de neurônios na camada oculta. Portanto, para os demais experimentos realizados, foi escolhido utilizar 4 neurônios na camada intermediária.

Outro experimento testou a evolução da precisão da rede neural em relação ao parâmetro *história*. Os resultados, exibidos na Figura 8.10, mostram que quanto maior a história, *i.e.*, quanto maior o intervalo de tempo observado, maior é a precisão da rede neural treinada. Entretanto, há dois fatores a se considerar. O primeiro é que quanto maior a história, maior é o número de neurônios na camada intermediárie, e, consequentemente, mais complexa se torna a rede neural para treinamento e operação. O outro fator relevante é que quanto maior a história, maior é o tempo de observação necessário para que a rede neural necessita para chegar

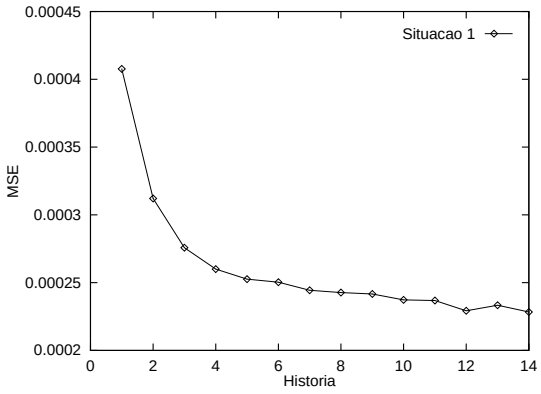
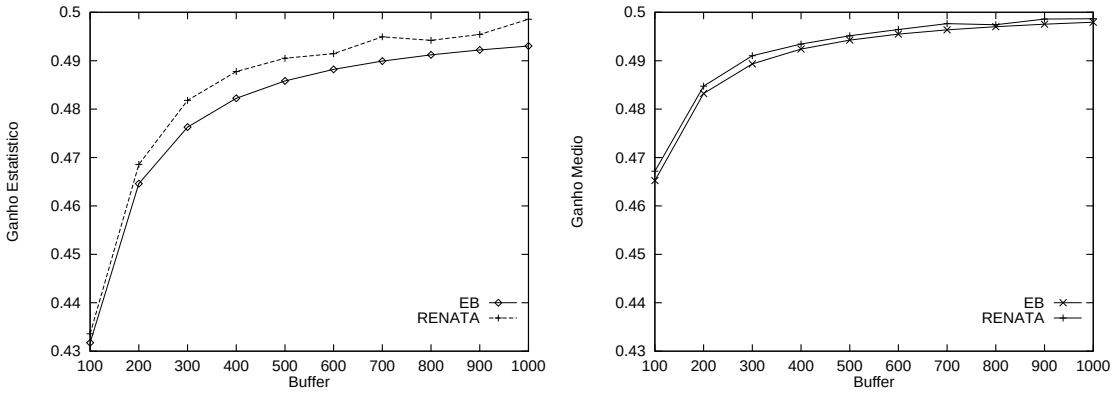


Figura 8.10: História $\times$ MSE

a algum resultado. Estes dois fatores devem ser observados de modo a se chegar a um valor que represente um meio termo entre precisão e desempenho. Para os demais experimentos realizados neste trabalho, foi escolhido utilizar *história* com valor 9. A rede neural projetada apresenta, portanto, $2 \times (h+1) = 2 \times 10 = 20$ neurônios na camada de entrada.

O último experimento realizado comparou o ganho estatístico obtido a partir do valor médio da capacidade requerida calculada pela rede neural projetada e treinada com as Amostras 1 e 2 com o ganho estatístico médio obtido da aplicação do método analítico EB às mesmas amostras. Os resultados são mostrados nas Figuras 8.11(a) e 8.11(b). Nota-se que a rede neural treinada calculou em todos as situações um ganho estatístico médio bem semelhante ao obtido da aplicação do método EB. Observa-se, também, que na Amostra 2, os resultados foram ainda mais aproximados do que na Amostra 1. Isto se deve ao fato da Amostra 2 ser gerada a partir de aplicações em número suficientemente grande para que seja mais fácil coletar as informações estatísticas necessárias que sejam realmente condizentes com a situação real do comportamento do tráfego. Por exemplo, em configurações onde existem apenas um número reduzido de aplicações, como entre 5 e 10, as informações estatísticas coletadas acerca do comportamento agregado (média e desvio padrão) não conseguem capturar precisamente que tipo de tráfego e capacidade equivalente que deve ser alocada.



(a) Amostra 1 (b) Amostra 2

Figura 8.11: Ganho Estatístico Médio Calculado $\times$ E_{abs}

Em geral, as redes neurais treinadas nos experimentos realizados apresentaram precisão aceitável na maioria dos casos. O principal problema que se observa é que para se ter uma classificação mais precisa, torna-se necessário um maior tempo de observação, o que nem sempre é viável em aplicações reais. Entretanto, para aplicações com maior exigência quanto a requisito de tempo-real, mesmo os resultados obtidos com pequenos períodos de observação podem ser considerados aceitáveis, visto que mesmo o método analítico utilizado (EB) é baseado em

informações estatísticas e, consequentemente, apresentam uma aproximação para um problema que não possui valor exato, por causa do comportamento estocástico do tráfego ATM.

Capítulo 9

Conclusões

"In the world, nothing is
certain but death and taxes."
– Benjamin Franklin

A quantidade de largura de banda a ser alocada a uma aplicação com o intuito do garantir sua Qualidade de Serviço e das demais aplicações é o conceito de Capacidade Requerida, um parâmetro extremamente importante para o Gerenciamento de Recursos em redes ATM.

A estimativa da Capacidade Requerida é, portanto, de grande importância para que haja a utilização mais racional de recursos por aplicações. A utilização de Redes Neurais na estimativa da Capacidade Requerida em comutadores ATM é a base da validação do primeiro módulo (*Módulo de Treinamento*) da arquitetura RENATA, uma arquitetura proposta neste trabalho, representando uma solução para a gerência pró-ativa de redes ATM.

Assim, a grande vantagem da nova proposta de gerenciamento idealizada na arquitetura RENATA consiste na possibilidade de antecipar situações reais de interesse dos mecanismos de gerenciamento através do uso de simulação para a geração de *baselines*. A utilização de Redes Neurais nesta arquitetura ajuda sobremaneira, baseado em resultados simulados, na tomada de decisão acerca de situações reais. Generalização, tempo-real e precisão de estimativa são

características importantes inerentes às Redes Neurais, sem as quais os métodos de gerência pró-ativa podem se tornar ineficazes.

Neste contexto proativo que caracteriza a ação de gerenciamento da RENATA, tem-se que a estimativa da Capacidade Requerida em comutadores ATM exige uma abordagem específica baseada nesta arquitetura. Esta abordagem resultou na experimentação objeto deste trabalho, permitindo ilustrar de forma concreta a importância da idéia proposta na arquitetura RENATA. Percebe-se, pelos resultados obtidos descritos no Capítulo 8, que as Redes Neurais são capazes de atingir, com grau de precisão aceitável, um valor estimado da capacidade requerida em um comutador ATM, utilizando como base o comportamento real de seu tráfego agregado. Assim, os descritores de tráfego tornam-se parcialmente dispensáveis, a medida que apenas a taxa de transmissão de pico (PCR) é de interesse da abordagem proposta, sendo todas as demais informações obtidas através de medições de tráfego.

Por tratar-se de uma arquitetura original recém-concebida, a RENATA apresenta-se mais como um ambiente de estudo e pesquisa da potencialidades das Redes Neurais dentre as soluções inteligentes na complexa tarefa de gerenciamento de redes ATM. Neste sentido, muito trabalho resta a ser feito na direção de se investigar que classes de gerenciamento são mais eficientemente tratadas por uma solução baseada em Redes Neurais, como proposto na RENATA. Este é, possivelmente, um dos trabalhos futuros na continuidade da pesquisa sobre a RENATA.

No que se refere especificamente à validação realizada, visualiza-se as seguintes etapas: escolha de melhores fatores de normalização, generalização de seu campo de atuação e comparação desta abordagem com métodos analíticos de estimativa da Capacidade Requerida. Estes trabalhos estão sendo desenvolvidos no *Département Réseaux et Services de Telecommunications* do INT (*Institut National des Télécommunications*), na França, no contexto do projeto Neuraltel (*Neural Networks on Telecommunications*). Este projeto, que pertence ao Programa ALFA (*Amérique Latine - Formation Académique*) da Comunidade Européia, se interessa pelo uso de Redes Neurais em Telecomunicações, tendo como parceiros a UFC (Universidade Federal do Ceará) e o CEFET-CE (Centro Federal de Educação Tecnológica do Ceará), além do próprio INT.

Bibliografia

[1] E. S. Artola and L. M. R. Tarouco. Um Sistema Especialista para Gerência Pró-Ativa Remota. In *XIV Brazilian Symposium on Computer Networks (SBRC 96)*, pages 118–139, Fortaleza, Brazil, May 1996.

[2] The ATM Forum, af-lane-0021.000. *LAN Emulation over ATM 1.0*, Jan 1995.

[3] The ATM Forum, af-lane-0057.000. *LANE Servers Management Specification v1.0*, Mar 1996.

[4] The ATM Forum, af-lane-0084.000. *LANE v2.0 LUNI Interface*, Jul 1997.

[5] The ATM Forum, af-lane-0093.000. *LAN Emulation Client Management Specification Version 2.0*, Oct 1998.

[6] The ATM Forum, af-saa-0048.000. *Native ATM Services: Semantic Descriptions*, Feb 1996.

[7] The ATM Forum, af-tm-0056.000. *Traffic Management Specification Version 4.0*, April 1996.

[8] The ATM Forum, af-uni-0010.002. *ATM User-Network Interface Specification Version 3.1*, 1994.

[9] J. P. Bigus. *Data Mining with Neural Networks*. McGraw-Hill, 1996.

[10] P. K. Campbell, A. Christiansen, M. Dale, H. L. Ferrá, A. Kowalczyk, and J. Szymanski. Experiments with Simple Neural Networks for Real-Time Control. *IEEE J. on Selected Areas in Communications*, 15(2):165–178, Feb 1997.

[11] A. C. Carvalho, M. C. Moruzzi, and E. D. Peterson. An Integrated Boolean Neural Network for Pattern Classification. *Pattern Recognition Letters*, 15:807–813, August 1994.

[12] J. Case, M. Fedor, M. Schoffstall, and J. Davin. A Simple Network Management Protocol. IETF RFC 1157, May 1990.

[13] R. Cole, D. Shur, and C. Villamizar. IP over ATM: A Framework Document. IETF RFC 1932, Apr 1996.

[14] Z. Dziong. *ATM Network Resource Management*. McGraw-Hill, 1997.

[15] A. Elwalid and D. Mitra. Effective Bandwidth of General Markovian Traffic Sources and Admission Control of High-Speed Networks. *IEEE/ACM Transactions on Networking*, 1(3):329–343, Jun 1993.

[16] R. H. Filho, S. Q. R. Teixeira, and M. Oliveira. Implementando Conhecimento em um Sistema para Apoio à Gerência de Redes. In *2o. Seminário Franco-Brasileiro em Sistemas Informáticos Distribuídos (SFBSID '97)*, Fortaleza, Brazil, 1997.

[17] O. Gällmo, E. Nordström, M. Gustafsson, and L. Asplund. Neural Networks for Preventive Traffic Control in Broadband ATM Networks. In *Proceedings of the World Congress on Neural Networks (WCNN-93)*, volume I, pages 295–299, Portland, Oregon, USA, Jul 1993.

[18] N. Golmie, F. Mouveaux, L. Hester, Y. Saintllan, A. Koenig, and D. Su. *The NIST ATM/HFC Network Simulator: Operation and Programming Guide - Version 4.0*. NIST - National Institute of Standards and Technology - U.S. Department of Commerce, Dec 1998.

[19] D. Griffin, editor. *Integrated Communications Management of Broadband Networks*. Crete University Press, Heraklio, Greece, 1996.

[20] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice-Hall, 1994.

[21] D. O. Hebb. *The Organization of Behavior: A Neuropsychological Theory*. Wiley, New York, 1949.

[22] S. R. Hedberg. AI's Impact in Telecommunications: Today and Tomorrow. *IEEE Expert Intelligent Systems & Their Applications*, 11(1), February 1996.

[23] A. Heybey. *The Network Simulator*. Laboratory of Computer Science, Massachusetts Institute of Technology, Oct 1989.

[24] A. Hiramatsu. *Handbook of Neural Computing*, chapter ATM Network Control by Neural Network. Institute of Phisics Publishing and Oxford University Publishing, 1997.

[25] HP. *HP Openview Extensible SNMP Agent 4.0 - Administrator's Guide*, 1998.

[26] ITU-T. *Digital Hierarchy Bit Rates*. Especificação G.702, Nov 1988.

[27] ITU-T. *ISDN - User-Network Interfaces - Reference Configurations*. Especificação I.411, 1988.

[28] ITU-T. *ISDN User-Network Interfaces – Reference Configurations*. Especificação I.412, 1988.

[29] ITU-T. *Broadband Aspects of ISDN*. Especificação I.413, 1990.

[30] ITU-T. *B-ISDN ATM Adaptation Layer (AAL) Specification*. Especificação I.363, 1991.

[31] ITU-T. *Frame Mode Bearer Services*. Especificação I.233, 1992.

[32] ITU-T. *B-ISDN ATM Adaptation Layer (AAL) Functional Description*. ITU-T Recommendation I.362, Nov 1993.

[33] ITU-T. *B-ISDN Service Aspects*. Especificação I.211, 1993.

[34] ITU-T. *ISDN Protocol Reference Model*. ITU-T Recommendation I.320, 1993.

[35] ITU-T. *B-ISDN Operation and Maintenance Principles and Functions*. Especificação I.610, Jan 1994.

[36] ITU-T. *B-ISDN Signaling ATM Adaptation Layer (SAAL) Overview Description*. Especificação Q.2100, Jul 1994.

[37] ITU-T. *Digital Subscriber Signalling System No. 2 (DSS 2) User-Network Interface (UNI) Layer 3*. Especificação Q.2931, Feb 1995.

[38] ITU-T. *Traffic Control and Congestion Control in B-ISDN*. Especificação I.731, Aug 1996.

[39] ITU-T. *The International Public Telecommunication Numbering Plan*. Especificação E.164, May 1997.

[40] R. Jain. Congestion Control and Traffic Management in ATM Networks: Recent Advances and A Survey. In *Computer Networks and ISDN Systems*, February 1995.

[41] P. Joos and W. Verbiest. A Statistical Bandwidth Allocation and Usage Monitoring Algorithm for ATM Networks. In *ICC '89*, pages 415–422, 1989.

[42] M. Laubach. Classical IP and ARP over ATM. IETF RFC 1577, 1994.

[43] C. S. Lindsey and T. Lindblad. Survey of Neural Network Hardware. In *Invited paper, SPIE 1995 Symposium on Aerospace Defense Sensing and Control and Dual Use Photonics*, Apr 1995.

[44] Y.-C. Liu and C. Dougligeris. Rate Regulation with Feedback Controller in ATM Networks – A Neural Network Approach. *IEEE Journal on Selected Areas in Communications*, 15(2):200–208, Feb 1997.

[45] T. Magedanz, K. Rothermel, and S. Krause. Intelligent Agents: An Emerging Technology for Next Generation Telecommunications ? In *INFOCOM '96*, March 1996.

[46] W. S. McCulloch and W. Pitts. A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.

[47] D. Minoli and T. Golway. *Planning & Managing ATM Networks*. Ed. Manning, Feb 1997.

[48] D. Minoli and A. Schmidt. *Client/Server Applications on ATM Networks*. Manning, 1996.

[49] M. L. Minsky and S. A. Papert. *Perceptrons*. MIT Press, Cambridge, MA, 1969.

[50] A. S. Nascimento. Desenvolvendo Agentes Inteligentes para a Gerência Pró-Ativa de Redes ATM. Master's thesis, Universidade Federal do Ceará, Mar 1999.

[51] J. E. Neves, L. B. de Almeida, and M. J. Leitão. ATM Call Control By Neural Networks. In *The International Workshop on Applications of Neural Networks to Telecommunications*, pages 210–217, 1993.

[52] A. Nigrin. *Neural Network for Pattern Recognition*. The MIT Press, Cambridge, MA, 1993.

[53] E. Nordström. A Hybrid Admission Control Scheme for Broadband ATM Traffic. In *Proceedings of the International Workshop on Applications of Neural Networks to Telecommunications*, pages 77–84. Lawrence Erlbaum, 1993.

[54] E. Nordström and J. Carlström. A Reinforcement Learning Scheme for Adaptive Link Allocation in ATM Networks. In *Proceedings of the International Workshop on Applications of Neural Networks to Telecommunications 2 (IWANNT'95)*, pages 300–307. Lawrence Erlbaum, 1995.

[55] E. Nordström, O. Gällmo, L. Asplund, M. Gustafsson, and B. Eriksson. Neural Networks for Admission Control in an ATM Network. In L. Niklasson and M. Bodén, editors,

Connectionism in a Broad Perspective: Selected Papers from the Swedish Conference on Connectionism - 1992, pages 239–250. Ellis Horwood, 1994.

[56] U. of Stuttgart. *SNNS - Stuttgart Neural Network Simulator - User Manual, Version 4.1*, 1995.

[57] A. L. I. Oliveira and J. A. S. Monteiro. Modelos de Tráfego para a Multiplexação Estatística do Tráfego de Dados em Redes ATM. In *Brazilian Symposium on Computer Networks (SBRC '98)*, pages 703–722, Rio de Janeiro, RJ, May 1998.

[58] M. Oliveira, M. Franklin, A. Nascimento, and M. V. conce los. *Introdução à Gerência de Redes ATM*. Editora CEFET-CE, 2nd. edition, 1998.

[59] R. O. Onvural. *Asynchronous Transfer Mode Network - Performance Issues*. Artech House Publishers, 2nd. edition, 1995.

[60] V. Peris. ATM Network Simulation. Technical Report CSHCN TR 92-1, Center for Satellite & Hybrid Communication Networks, 1992.

[61] J. M. Pitts and J. A. Schormans. *Introduction to ATM Design and Performance*. John Wiley & Sons, 1996.

[62] M. T. Rose. *The Open Book: A Practical Perspective on OSI*. Prentice-Hall, 1990.

[63] F. Rosemblatt. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review*, 65:386–408, 1958.

[64] D. E. Rumelhart and J. L. McClelland. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1. MIT Press, Cambridge, MA, 1986.

[65] A. Schuhknecht and G. Dreo. Preventing Rather Repairing - A New Approach in ATM Network Management. In *INET '95*, 1995.

[66] G. M. Shepherd and C. Koch. Introduction to Synaptic Circuits. In G. M. Shepherd, editor, *The Synaptic Organization of the Brain*, pages 3–31. Oxford University Press, New York, 1990.

[67] L. F. G. Soares, G. Lemos, and S. Colcher. *Redes de Computadores: Das LANs, MANs e WANs às Redes ATM*. Ed. Campus, Rio de Janeiro, 2nd. edition, 1997.

[68] R. A. M. Sprenkels. Management of ATM Networks. Master's thesis, University of Twente, June 1996.

[69] W. Stallings. *High-Speed Networls: TCP/IP and ATM Principles*. Prentice Hall, 1998.

[70] University of Stuttgart - Institute for Parallel and Distributed High Performance Systems (IPVR). *The Stuttgart Neural Network Simulator - Version 4.0*, 1995.

[71] G. C. Vasconcelos, E. C. de Barros Carvalho Filho, T. B. Ludermir, and D. L. Borges. Redes Neurais: Filosofia, Teoria, Modelagem e Aplicações. Minicurso - JAI - SBC '96, 1996.

[72] M. V. Vasconcelos and M. Oliveira. Utilizando Banco de Dados Ativos no Suporte ao Gerenciamento Pró-Ativo de Redes ATM. In *Simpósio Franco-Brasileiro de Sistemas Informáticos Distribuídos*, Fortaleza, Brazil, November 1997.

[73] P. J. Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavior Sciences*. PhD thesis, Harvard University, Cambridge, MA, 1974.

[74] B. Widrow and J. M. E. Hoff. Adaptive Switching Circuits. In *IRE WESCON Convention Record*, pages 96–104, 1960.

[75] J. Witters, A. B. Nielsen, J. Kroeze, H. Pettersen, and T. Renger. Results of Eperiments on Traffic Control Using Real Applications. Technical report, EXPLOIT, 1994.

[76] A. J. Wong. Recognition of General Patterns Using Neural Networks. *Biological Cybernetics, Springer-Verlag*, 58:361–372, 1996.

Apêndice A

Glossário de Acrônimos

AAL	*ATM Adaptation Layer*
ABR	*Avaliable Bit Rate*
ATM	*Asynchronous Transfer Mode*
bps	*bits por segundo*
BECN	*Backward Explicit Congestion Notification*
B-ISDN	*Broadband Integrated Service Digital Network*
B-NT	*Broadband Network Terminal*
B-TE	*Broadband Terminal Equipment*
CAC	*Connection Admission Control*
CATV	*Community Antenna Television*
CBR	*Constant Bit Rate*
CCITT	*Consultative Committee for International Telephone and Telegraph*
CDV	*Cell Delay Variation*
CLP	*Cell Loss Priority*
CLR	*Cell Loss Ratio*
CMIP	*Commom Management Information Protocol*
CRC	*Cyclic Redundancy Checks*
EFCI	*Explicit Forward Congestion Indication*
HEC	*Header Error Check*
IETF	*Internet Engineering Task Force*
ISDN	*Integrated Services Digital Network*

ISO	*International Organization for Standardization*
ITU-T	*International Telecommunication Union Telecommunication Standardization Sector*
IWU	*Interworking Unit*
MAC	*Media Access Control*
MAN	*Metropolitan Area Network*
MPEG	*Moving Picture Experts Group*
NIST	*National Institute of Standards and Technology*
NNI	*Network Node Interface*
NT	*Network Terminal*
PDU	*Protocol Data Unit*
PT	*Payload Type*
RDSI	*Rede Digital de Serviços Integrados*
RDSI-FE	*Rede Digital de Serviços Integrado de Faixa Estreita*
RDSI-FL	*Rede Digital de Serviços Integrado de Faixa Larga*
RFC	*Request For Comments*
QoS	*Quality of Service*
RENPAC	*Rede Nacional de Pacotes*
SAAL	*Signalling AAL*
SAR	*Segmentation and Reassembly*
SDH	*Synchronous Digital Hierarchy*
SDU	*Service Data Unit*
SNMP	*Simple Network Management Protocol*
SONET	*Synchronous Optical Network*
STM	*Synchronous Transfer Mode*
TA	*Terminal Adapter*
TCP	*Transport Control Protocol*
TDM	*Time Division Multiplexing*
TE	*Terminal Equipment*
UNI	*User-Network Interface*
UME	*UNI Management Entity*
VBR	*Variable Bit Rate*
VCC	*Virtual Channel Connection*

VCL *Virtual Channel Link*

VCI *Virtual Circuit Identifier*

VPC *Virtual Path Connection*

VPI *Virtual Path Identifier*

VPL *Virtual Path Link*

WAN *Wide Area Network*

Apêndice B

O Algoritmo da Capacidade Equivalente

Um modelo com estados ativos e inativosé utilizado para caracterizar cada fonte. É assumido que a duração de cada estado é distribuída exponencialmente e independentemente um do outro. Faça-se:

R = Taxa de transmissão de pico da conexão

b = Duração média do período ativo

ρ = Utilização da fonte
(*i.e.* probabilidade da fonte estar em período ativo)

μ = Taxa de transmissão de saída do período ativo
$\mu = 1/b$

λ = Taxa de transmissão de saída do período inativo
$\lambda = \rho/(b(1-\rho))$

c = Velocidade do *link*

X = Tamanho do *buffer*

Faça-se $P_i(t,x)$ denotar a probabilidade da fonte estar no estado i e o conteúdo do *buffer* ser x no instante t, definindo-se $i = 1$ se a fonte está ativa e $i = 0$ se a fonte estiver inativa. Então, a ocupação do único *buffer* considerado é descrita pelas seguintes equações:

$$P_0(t+dt,x) = (1-\lambda.dt)P_0(t,x+c.dt) + \mu.dt.P_1(t,x+(c-R).dt) \qquad \text{(B.1)}$$

$$P_1(t+dt,x) = \lambda.dt.P_0(t,x+c.dt) + (1-\mu.dt)P_1(t,x+(c-R).dt) \tag{B.2}$$

Após algumas manipulações, as equações B.1 e B.2 resultam nas seguintes equações diferenciais:

$$\frac{\partial P_0(t,x)}{\partial t} - c\frac{\partial P_0(t,x)}{\partial x} = -\lambda P_0(t,x) + \mu P_1(t,x) \tag{B.3}$$

$$\frac{\partial P_1(t,x)}{\partial t} - (c-R)\frac{\partial P_0(t,x)}{\partial x} = \lambda P_0(t,x) + \mu P_1(t,x) \tag{B.4}$$

O comportamento estacionário do sistema é, então, caracterizado pelo vetor $F(x) = [F_0(x), F_1(x)]^T$, onde:

$$F_i(x) = \lim_{t\to\infty} P_i(t,x) \qquad i = 0,1 \tag{B.5}$$

que é a solução do seguinte sistema de equações:

$$\begin{bmatrix} -c & 0 \\ 0 & -(c-R) \end{bmatrix} F'(x) = \begin{bmatrix} -\lambda & \mu \\ \lambda & -\mu \end{bmatrix} F(x) \tag{B.6}$$

onde $F'(x)$ é a derivada de F(x) em relação a x. A solução deste sistema, $F(x)$, é:

$$F(x) = \alpha_0 \begin{bmatrix} \mu \\ \lambda \end{bmatrix} + \alpha_1 \begin{bmatrix} (R-c)/c \\ 1 \end{bmatrix} exp\{\frac{-x(c-\rho R)}{b(1-\rho)(R-c)c}\} \tag{B.7}$$

onde as constantes α_0 e α_1 devem ser obtidos a partir das condições de limitação. Assumindo $R > c$, o *buffer* não pode estar vazio. Portanto, $F_1(0) = 0$. Similarmente, observando-se que o *buffer* não pode estar cheio em período inativo, tem-se $F_0(X^-) = 1-\rho$. Utilizando estas duas condições de limitação e a equação B.7, tem-se:

$$\alpha_0 = b(1-\rho)^2/\Delta \qquad \text{e} \qquad \alpha_1 = -\rho(1-\rho)c/\Delta \tag{B.8}$$

onde

$$\Delta = (1-\rho)c - \rho(R-c)exp\{-X(c-\rho R)/[b(1-\rho)(R-c)c]\} \tag{B.9}$$

Portanto, a distribuição do tamanho da fila $Pr\{Q < x\} = F_0(x) + F_1(x)$ é tido como:

$$Pr\{Q<x\} = \begin{cases} \frac{c(1-\rho)}{\Delta} - \frac{\rho(1-\rho)R}{\Delta} exp\{\frac{-x(c-\rho R)}{b(1-\rho)(R-c)c}\}, & \text{se } x < X; \\ 1, & \text{caso contrário.} \end{cases} \quad \text{(B.10)}$$

A probabilidade de extrapolação do estado estacionário p é igual à probabilidade de uma fonte estar ativa e o *buffer* estar cheio, o que pode ser obtido a partir da identidade $\pi_1 = p + F_1(X^-)$, onde $\pi_1 = \rho$ é a probabilidade da fonte estar ativa. Portanto, tem-se:

$$p = \frac{\rho(c-\rho R)exp\{-X(c-\rho R)/[b(1-\rho)(R-c)c]\}}{(1-\rho)c - \rho(R-c)exp\{-X(c-\rho R)/[b(1-\rho)(R-c)c]\}} \quad \text{(B.11)}$$

Utilizando esta estruturação, a quantidade de largura de banda requerida por uma conexão é obtida isoladamente como a resposta à seguinte pergunta: Se a conexão com parâmetros (R,m,b) é a entrada do *link* com capacidade de *buffer* X, qual deve ser a taxa de transmissão para esta conexão para atingir uma probabilidade de estouro de *buffer* ε? Para encontrar a probabilidade condicional do *buffer* estar estourando, é assumido que $\varepsilon = \rho p$. Então, com $\delta = (c-\rho R)/[b(1-\rho)(R-c)c]$, a equação B.11 se torna:

$$(c-\rho R)exp\{-\delta X\} = \varepsilon(1-\rho)c - \rho(R-c)exp\{-\delta X\} \quad \text{(B.12)}$$

Utilizando o fato da probabilidade de estouro dever ser menor ou igual a ε, [B.12], e a identidade $\varepsilon = \rho p$, tem-se:

$$e^{-\delta X} \leq \frac{\varepsilon(1-\rho)c}{(c-\rho R) + \varepsilon\rho(R-c)} \quad \text{(B.13)}$$

onde c representa a capacidade equivalebte da conexão. Como [B.13] inclui tanto funções racionais quanto exponenciais, a derivação de uma solução explícita para c não é possível. Além disto, uma solução numérica da equação é dispendiosa quanto a tempo de processamento, considerando a necessidade de tempo-real para realizar o processamento. Como solução, um patamar superior para o valor de c é obtido através da substituição de $(1-\rho)c/[(c-\rho R) + \varepsilon\rho(R-c)] = 1$ em [B.13], que será reduzida à seguinte equação quadrática de c:

$$\alpha b(1-\rho)c^2 + [X - \alpha b(1-\rho)R]c - X\rho R = 0$$

onde $\alpha = ln(1/\varepsilon)$. Portanto, a capacidade equivalente é:

$$c = R\frac{y - X + \sqrt{(y-X)^2 + 4X\rho y}}{2y} \quad \text{(B.14)}$$

com $y = \alpha b(1-\rho)R$. Desta forma, a quantidade total de largura de banda de n conexões multiplexadas é igual ao somatório das capacidades equivalentes individuais das conexões c_i, isto é:

$$C = \sum_{i=1}^{n} c_i$$

Entretanto, C pode superestimar significativamente o valor da capacidade equivalente das n conxões multiplexadas para o tráfego agregado, já que a interação entre conexões não é levada em consideração. Para captar este efeito da multiplexação, a Aproximação Gaussiana [59] é utilizada conjuntamente à Capacidade Equivalente. Em partcular, o total de largura de banda requerida para o tráfego agregado C para as n conexões é dada por:

$$C = min\{m + \alpha'\sigma, \sum_{i=1}^{n} c_i\} \tag{B.15}$$

com os valores de média m_i e variância σ_i^2 da taxa de transmissão de cada conexão com o modelo de fluxo acima sendo respectivamente iguais a $R_i b_i$ e $m_i(R_i - m_i)$, e:

$$m = \sum_{i=1}^{n} m_i, \qquad \sigma^2 = \sum_{i=1}^{n} \sigma_i^2$$

Apêndice C

Parâmetros de Configuração do Módulo de Treinamento

C.1 Amostra 1

```
/* Capacidade do Link de Saida (Mbps) */
#define LINK_OUT 51.84

/* Capacidade do Link de Entrada (Mbps) */
#define LINK_IN 155.52

/* Capacidade minima do buffer */
#define MIN_BUFFER 100

/* Capacidade maxima do buffer */
#define MAX_BUFFER 1000

/* Passo de incremento da capacidade do buffer */
#define PASSO_BUFFER 100
```

```
/* Numero de configuracoes de rede geradas */
#define NUM_CONF 4000

/* Numero minimo de aplicacoes geradas */
#define MIN_APLIC 5

/* Numero maximo de aplicacoes geradas */
#define MAX_APLIC 100

/* Numero de exemplos gerados a partir de cada configuracao */
#define EXEMPCONF 1

/* Numero maximo de tuplas no banco de exemplos de treinamento */
#define MAXEXEMP 70000

/* Quantidade de intervalos de tempo observados */
#define HISTORIA 9

/* Intervalo Delta tau entre cada ponto de medida (MP) (em microseg) */
#define INTERVALO 100

/* Numero de MP's que compoem um CP */
#define NUM_MPS 50

/* Valores minimo e maximo do PCR (em Mbps) */
#define MIN_PCR 0.10
#define MAX_PCR 3.00
```

```
#define GRAN_PCR 0.10

/* Valores minimo e maximo do t_on (em ms) */
#define MIN_TON 0.01
#define MAX_TON 2.50
#define GRAN_TON 0.01

/* Valores minimo e maximo do t_off (em ms) */
#define MIN_TOFF 0.01
#define MAX_TOFF 2.50
#define GRAN_TOFF 0.01

/* Valor maximo do CLR (Cell Loss Ratio) */
#define EPSILON 0.00001

/* Definicao da granularidade do log (em 10 ns - time-tick) */
#define LOGGING 1

/* Tempo total da simulacao (em microseg) */
#define TEMPO 1000000

/* Apagar o arquivo de Log apos o processamento: Sim (1) / Nao (0) */
#define APAGA 0

/* Compactar o arquivo de Log apos o processamento: Sim (1) / Nao (0) */
#define COMPACTA 1

/* Numero de entradas iniciais do vetor */
```

```
#define XENTRADAS 2  /* Numero de Aplicacoes e Som. de PCR */

/* Identificacao dos Parametros no arquivo de log do simulador ATM */

/* Numero de parametros observados */
#define NUMPARAM 1

/* Parametro: Taxa de Transmissao instantanea no Link de Saida */
#define PAR_RATE 1

/* Parametro: Percentagem de Celulas Perdidas */
#define PAR_PERC 2

/* Parametro: Quantidade de Celulas no buffer */
#define PAR_C_BUFF 3

/* Parametro: Quantidade de Celulas Descartadas */
#define PAR_C_DROP 4
```

C.2 Amostra 2

```
/* Capacidade do Link de Saida (Mbps) */
#define LINK_OUT 51.84

/* Capacidade do Link de Entrada (Mbps) */
#define LINK_IN 155.52

/* Capacidade minima do buffer */
#define MIN_BUFFER 100

/* Capacidade maxima do buffer */
#define MAX_BUFFER 1000

/* Passo de incremento da capacidade do buffer */
#define PASSO_BUFFER 100

/* Numero de configuracoes de rede geradas */
#define NUM_CONF 4000

/* Numero minimo de aplicacoes geradas */
#define MIN_APLIC 20

/* Numero maximo de aplicacoes geradas */
#define MAX_APLIC 80

/* Numero de exemplos gerados a partir de cada configuracao */
#define EXEMPCONF 1
```

```
/* Numero maximo de tuplas no banco de exemplos de treinamento */
#define MAXEXEMP 70000

/* Quantidade de intervalos de tempo observados */
#define HISTORIA 9

/* Intervalo Delta tau entre cada ponto de medida (MP) (em microseg) */
#define INTERVALO 100

/* Numero de MP's que compoem um CP */
#define NUM_MPS 50

/* Valores minimo e maximo do PCR (em Mbps) */
#define MIN_PCR 0.10
#define MAX_PCR 2.75
#define GRAN_PCR 0.10

/* Valores minimo e maximo do t_on (em ms) */
#define MIN_TON 0.01
#define MAX_TON 2.50
#define GRAN_TON 0.01

/* Valores minimo e maximo do t_off (em ms) */
#define MIN_TOFF 0.01
#define MAX_TOFF 2.50
#define GRAN_TOFF 0.01

/* Valor maximo do CLR (Cell Loss Ratio) */
```

```
#define EPSILON 0.00001

/* Definicao da granularidade do log (em 10 ns - time-tick) */
#define LOGGING 1

/* Tempo total da simulacao (em microseg) */
#define TEMPO 1000000

/* Apagar o arquivo de Log apos o processamento: Sim (1) / Nao (0) */
#define APAGA 0

/* Compactar o arquivo de Log apos o processamento: Sim (1) / Nao (0) */
#define COMPACTA 1

/* Numero de entradas iniciais do vetor */
#define XENTRADAS 2  /* Numero de Aplicacoes e Som. de PCR */

/* Identificacao dos Parametros no arquivo de log do simulador ATM */

/* Numero de parametros observados */
#define NUMPARAM 1

/* Parametro: Taxa de Transmissao instantanea no Link de Saida */
#define PAR_RATE 1

/* Parametro: Percentagem de Celulas Perdidas */
#define PAR_PERC 2
```

```
/* Parametro: Quantidade de Celulas no buffer */
#define PAR_C_BUFF 3

/* Parametro: Quantidade de Celulas Descartadas */
#define PAR_C_DROP 4
```