

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
DEPARTAMENTO DE TELEMÁTICA

Dimensionamento de Enlaces em Redes com Integração de Serviços

Autor: **Renê Souza da Mata**

Orientador: **Prof. Dr. Shusaburo Motoyama**

Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos pré-requisitos necessários a obtenção do título de Mestre em Engenharia Elétrica.

Área de Concentração: **AG - Telecomunicações e Telemática**

Banca Examinadora:

Prof. Dr. Shusaburo Motoyama (DT/FEEC/UNICAMP)
Prof. Dr. Ivanil Sebastião Bonatti (DT/FEEC/UNICAMP)
Prof. Dr. Akebo Yamakami (DT/FEEC/UNICAMP)
Dra. Eunice Luvizotto M. Pissolato (Fundação CPQD)

Campinas - SP - Brasil

25/04/2002

Resumo

Neste trabalho é tratado o dimensionamento da capacidade de enlaces em redes com integração de serviços (ex.: voz, vídeo e dados). Diferentes tipos de serviços demandam níveis de QoS (Quality of Service) distintos - tais como atraso, jitter e taxa de perda de pacotes - que devem ser providos pela rede. Neste sentido, são propostos dois métodos para o cálculo da capacidade dos enlaces, sendo que em ambos o atraso médio dos pacotes e a taxa de perda de pacotes são os parâmetros de QoS utilizados. No primeiro método o escalonador encaminha os pacotes armazenados num *buffer* limitado do tipo FIFO, sem distinção alguma entre os pacotes. O segundo método consiste num esquema de prioridade entre os pacotes, isto é, os pacotes são colocados num *buffer* com capacidade limitada, porém o escalonador os encaminha conforme prioridades pré-definidas. Em adição, foi desenvolvido um software como ferramenta de auxílio à tarefa de dimensionamento. A interatividade e a automatização de cálculos complexos, originados da metodologia proposta, são as principais características deste software.

Palavras-chave: Dimensionamento, enlaces, capacidade, largura de banda, redes com integração de serviços, RDSI, QoS.

Abstract

The subject of this work is the dimensioning of link capacity for networks with integrated services (e.g., voice, video and data). Different types of services demand distinct levels of QoS (Quality of Service) - such as delay, jitter and packet loss ratio - which must be provided by the network. In this sense, two methods for link capacity calculation are proposed, both of them using packet average delay and packet loss ratio QoS parameters. For the first method, the scheduler forwards the packets stored in a size-limited FIFO *buffer*, with no distinction among the packets. The second method consists of a priority scheme among the packets, i.e., the packets are placed into a size-limited *buffer* and the scheduler forwards them according to a previously defined priority. Additionally, a software package was developed as an aiding tool for the dimensioning. The main features of this software are interactivity and automation of the complex calculus originated by the proposed methodology.

Key-words: Dimensioning, links, capacity, band width, integrated services digital networks, ISDN, QoS.

*Aos meus pais e aos meus irmãos, pelas
condições que me proporcionaram;*

À Eliana, pelo amor, carinho e paciência;

*À Livia Silveira de Freitas (em memória),
minha querida amiga.*

DEDICO ESTE TRABALHO

“Mestre não é quem sempre ensina, mas quem de repente aprende”.

(Guimarães Rosa)

Agradecimentos

Agradeço a Deus, em primeiro lugar, pela força e luz recebidas em todos os momentos decisivos, ou não, para a realização deste trabalho.

Agradeço também ao professor Motoyama pela confiança, apoio e orientação durante todas as fases do curso de mestrado, especialmente pela orientação no trabalho de tese em todas as suas etapas.

Expresso minha gratidão também ao professor Ivanil, que gentilmente cedeu-me alguns minutos com importantes comentários.

Aos meus amigos José Renato, Dean e Eduardo, que viveram comigo durante todo o período de elaboração deste trabalho, sou muito grato.

Enfim, agradeço ao apoio financeiro da CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior).

Índice

Resumo.....	ii
Abstract	ii
Agradecimentos.....	v
Capítulo 1 - Introdução	1
Capítulo 2 - Técnicas para a Integração de Serviços.....	4
2.1 - Introdução.....	4
2.2 - Redes ATM.....	5
2.3 - Arquiteturas de Serviços Integrados (IntServ).....	6
2.4 - Arquitetura de Serviços Diferenciados (DiffServ).....	8
2.5 - Multi-Protocol Label Switching (MPLS)	10
2.6 - VoIP (Voz sobre IP)	13
2.6.1 - VoIP através da Internet.....	14
2.7 - Qualidade de Serviço.....	15
2.7.1 - Fatores que Influenciam a Qualidade do Sinal de Voz	15
2.7.1.1- Largura de Banda	15
2.7.1.2 - Atraso de Pacote.....	16
2.7.1.3 - Perda de Pacotes.....	18
2.7.2 - Mecanismos para Prover QoS ao Tráfego de Voz em Redes IP	19
2.7.2.1 - Enfileiramento, Priorização e Disciplina de Escalonamento.....	20
2.7.2.1.1 - First-In, First-Out - FIFO.....	20
2.7.2.1.2 - Fila com Prioridade (Priority Queueing - PQ)	21
2.7.2.1.3 - Custom Queueing - CQ	22
2.7.2.1.4 - Weighted Round Robin - WRR	23
2.7.2.1.5 - Weighted Fair Queueing - WFQ.....	23
2.8 - Dimensionamento de Enlaces em Redes de Pacotes.....	25
2.9 - Conclusões	27

Capítulo 3 - Modelos de Dimensionamento de Enlaces	28
3.1 - Introdução	28
3.2 - Modelo de Dimensionamento FIFO (sem prioridade)	29
3.2.1 - Modelo de Filas M/M/1/N.....	30
3.2.2 - Metodologia para o Cálculo da Capacidade do Enlace.....	32
3.2.2.1 - Determinação de ρ a partir de PB.....	32
3.2.2.2 - Determinação de ρ a partir de $E\{T\}$	33
3.2.2.3 - Cálculo de ρ a partir de $fPB(\rho)$ e $fT(\rho)$	34
3.3 - Modelo de Dimensionamento com Prioridades	37
3.3.1 - Modelo de Filas com Prioridades e sem Preempção	38
3.3.2 - Metodologia para o Cálculo da Capacidade do Enlace.....	41
3.3.2.1 - Modelo para o Cálculo da Capacidade do Enlace.....	41
3.3.2.2 - Modelo Equivalente para o Dimensionamento do Buffer	43
3.4 - Conclusões	46
Capítulo 4 - Software de Dimensionamento de Enlaces	47
4.1 - Introdução	47
4.2 - Descrição do Software Desenvolvido	48
4.3 - Exemplos de Dimensionamento	52
4.3.1 - Dimensionamento do Enlace 1	53
4.3.2 - Dimensionamento do Enlace 2	54
4.3.3 - Simulações com os Resultados do Dimensionamento	55
4.3.3.1 - Simulações com os resultados do Enlace 1	55
4.3.3.2 - Simulações com os resultados do Enlace 2.....	56
4.3.3.3 - Validação dos modelos de dimensionamento propostos	57
4.4 - Considerações sobre os Resultados do Dimensionamento	58
4.4.1 - Oscilações na intensidade do tráfego de entrada do enlace.....	59
4.4.2 - Tráfego de entrada com outra distribuição estatística.....	59
4.5 - Conclusões	62
Capítulo 5 - Conclusões	63
Referências Bibliográficas.....	66
Apêndice I - Raízes de Polinômios do 3o Grau.....	68
Apêndice II - Relatórios das Simulações Feitas Através do Arena	69

Capítulo 1

Introdução

Várias técnicas estão sendo propostas na literatura para transportar integradamente, em uma única rede, os serviços de voz, vídeo e dados. Uma das técnicas bastante utilizada em redes backbone é o ATM (*Asynchronous Transfer Mode*). O ATM foi desenvolvido para implementar a RDSI-FL (Rede Digital de Serviços Integrados de Faixa Larga) com uma concepção completamente nova e com a preocupação de garantir QoS (*Quality of Service*) para cada classe de serviço independentemente [1]. No entanto, devido a sua complexidade, o desenvolvimento de aplicativos (*softwares*) é bastante lento e custoso.

Outras técnicas para a integração, em desenvolvimento, são baseadas sobre plataformas com a arquitetura do IP (*Internet Protocol*). O IP é a parte essencial da rede Internet e associado ao TCP (*Transmission Control Protocol*) e UDP (*User Datagram Protocol*) formam uma plataforma com larga escala de aplicações. Graças ao extraordinário crescimento da Internet, a rede IP tornou-se uma das maiores redes de dados do mundo. Contudo, a rede IP atual não garante a QoS dos sinais transportados. A filosofia a partir de uma arquitetura IP é que se deve utilizar eficientemente a largura de banda disponível na rede. Esta por sua vez trata a demanda do tráfego, proveniente das diversas aplicações, na base do melhor esforço.

Grandes esforços estão sendo feitos para o provimento de QoS em redes IP. Uma das técnicas para suprir esta QoS é denominada IntServ (*Integrated Service*), especificada pelo IETF (*Internet Engineering Task Force*) [2]. Esta técnica requer um protocolo de sinalização fim-a-fim, o RSVP (*Resource ReSerVation Protocol*) [3], e um mecanismo de alocação de largura de banda no roteador ou

comutador de pacotes. Com a integração desses dois mecanismos consegue-se um serviço orientado à conexão na Internet [4], e a QoS para um tráfego específico é garantida. Alguns problemas ligados à escalabilidade da rede tem sido apontados. O primeiro problema é consequência da utilização de protocolo de sinalização, o qual deve interagir com todos os roteadores ao longo de um caminho sobre informações do estado da rede. O segundo está relacionado ao algoritmo de gerenciamento dos *buffers* (*packet scheduling algorithm*), pois quando o número de conexões ativas na rede aumenta, o tempo de processamento nos roteadores também aumenta.

Com o objetivo de resolver os problemas de escalabilidade existentes no IntServ o IETF propôs outra arquitetura, a DiffServ (*Differentiated Services*) [5]. Esta nova proposta é representada principalmente por um mecanismo de gerenciamento de largura de banda para redes IP. A existência de um byte presente no cabeçalho de cada pacote IP, o ToS (*Type of Service*), torna possível a marcação da prioridade ou nível de serviço que cada pacote requer. Os fluxos provenientes das aplicações (sub-redes LANs, Frame Relays ou RDSI) interligados aos roteadores de borda (*Edge Routers*) são agregados em classes de tráfego pré-definidas (mesmo ToS). Isto possibilita minimizar o processamento gerado nos roteadores da rede, pois as informações são tratadas não como um fluxo simples (IntServ), mas como um agregado de fluxos.

Ainda no contexto de provimento de QoS em redes IP, mais recentemente vem sendo propostas as tecnologias MPLS (*Multi Protocol Label Switch*) e IP sobre WDM. O MPLS é uma técnica avançada de encaminhamento de pacotes, através de rótulos inseridos nestes, por roteadores do domínio MPLS. Os pacotes são mapeados em FECs (*Forward Equivalence Classes*), oferecendo desta forma fluxos diferenciados com diferentes requisitos de QoS [6]. Com a utilização de técnicas fotônicas o WDM permite a criação de caminhos ópticos fim-a-fim para o encaminhamento de tráfegos IP. Esta tecnologia, além de garantir largura de banda, provê também garantia de QoS, segurança e tolerância a falhas [7]. As duas propostas fornecem caminhos físicos fim-a-fim de excelente qualidade para a conexão dos roteadores IP. Desta forma, a QoS e a largura de banda entre os roteadores de borda é garantida.

Independentemente da tecnologia utilizada, ainda existe um fundamental limite entre elas: uma capacidade física (e/ou lógica) do caminho deve ser determinada a priori. O modelo de dimensionamento tem um papel essencial na determinação da capacidade apropriada dos enlaces da rede.

Este modelo deve levar em conta, ao mesmo tempo, a minimização de custos pré-definidos e a garantia de QoS mínima a ser provida. Como em muitas outras situações de dimensionamento, o interesse principal deve ter o contrapeso apropriado entre estes dois fatores. Dessa forma, o desenvolvimento de modelos de dimensionamento dos recursos de redes, para se obter desempenho máximo, é muito complexo.

Por outro lado, como o projeto de redes é uma tarefa essencial nas telecomunicações, as técnicas de otimização, com literatura ampla, têm-se por muito tempo agido como ferramentas poderosas para as finalidades de planejamento. No entanto, poucos trabalhos de pesquisa foram feitos sobre o dimensionamento da capacidade de enlaces em redes, levando-se em conta algumas restrições de garantia de QoS.

Neste trabalho são propostos dois modelos de dimensionamento da capacidade de enlaces em redes de comutação de pacotes. Ambos consideram como parâmetros principais de restrição de QoS o atraso médio e a taxa de perda dos pacotes, os quais são previamente definidos. No primeiro modelo o servidor trata todos os pacotes de forma indiscriminada, os pacotes aguardam pelo atendimento numa fila FIFO com capacidade limitada. No segundo modelo é associada uma classe de prioridade para cada pacote. Estes pacotes, inicialmente armazenados num *buffer* com capacidade limitada, são atendidos conforme uma classe de prioridade previamente definida. Adicionalmente, com a finalidade de automatizar o cálculo da capacidade dos enlaces é desenvolvido um software como ferramenta de auxílio ao usuário.

O complemento deste trabalho está organizado em outros quatro capítulos: No capítulo 2 são apresentadas as principais técnicas para o provimento de QoS em redes com integração de serviços e algumas metodologias utilizadas para o dimensionamento de redes. Os modelos propostos para o dimensionamento de enlaces, sujeitos às restrições de QoS pré-definidas (atraso médio e taxa de perda dos pacotes) são apresentados no capítulo 3. No capítulo 4 é apresentado o software desenvolvido, seguido por alguns exemplos de dimensionamento de enlaces, feitos através deste software. Ainda no capítulo 4, são mostradas validações e algumas considerações sobre a sensibilidade dos modelos propostos. Finalmente, as principais conclusões e sugestões para trabalhos futuros são apresentadas no capítulo 5.

Capítulo 2

Técnicas para a Integração de Serviços

2.1 - Introdução

Tradicionalmente, sistemas de comunicação foram desenvolvidos para o transporte de tipos específicos de informações - o sistema telefônico para o tráfego de voz, as redes de comutação de pacotes para dados textuais, vídeo e televisão em redes de radiodifusão ou a cabo. Essas redes foram claramente projetadas para aplicações específicas adaptando-se mal a outros tipos de serviço. O ideal de uma única rede capaz de atender a todos esses serviços, de forma a obter uma economia devido ao compartilhamento dos recursos, motivou o conceito de redes com integração de serviços.

Neste capítulo, primeiramente, são apresentadas cinco das principais técnicas para implementação de redes que possam dar suporte à demanda de diferentes fontes de tráfego - a ATM, a IntServ (*Integrated Service*), a DiffServ (*Differentiated Services*), a MPLS (*Multi Protocol Label Switch*) e a VoIP (*Voice over IP*). Em seguida, são mostrados os principais requisitos de QoS (*Quality of Service*) e as principais técnicas, utilizadas pelas tecnologias acima, para o provimento destes. E, ao final, são mostradas algumas técnicas utilizadas para realização de dimensionamento de enlaces em redes de pacotes.

2.2 - Redes ATM [8]

Uma rede ATM é composta por um conjunto de comutadores ATM interligados através de interfaces ou *links* (enlaces) ATM. Os comutadores ATM suportam dois tipos de interfaces: UNI (*User-Network Interface*) e NNI (*Network-Network Interface*). A UNI conecta estações finais (ou roteadores, etc) a um comutador ATM, enquanto a NNI serve para conectar dois comutadores ATM. A Fig. 2.1 mostra a estrutura geral de uma rede ATM. Existem dois tipos de redes ATM: as redes privadas e as redes públicas. Tradicionalmente, esses conceitos estão relacionados com o uso de circuitos compartilhados (rede pública) ou dedicados (rede privada).

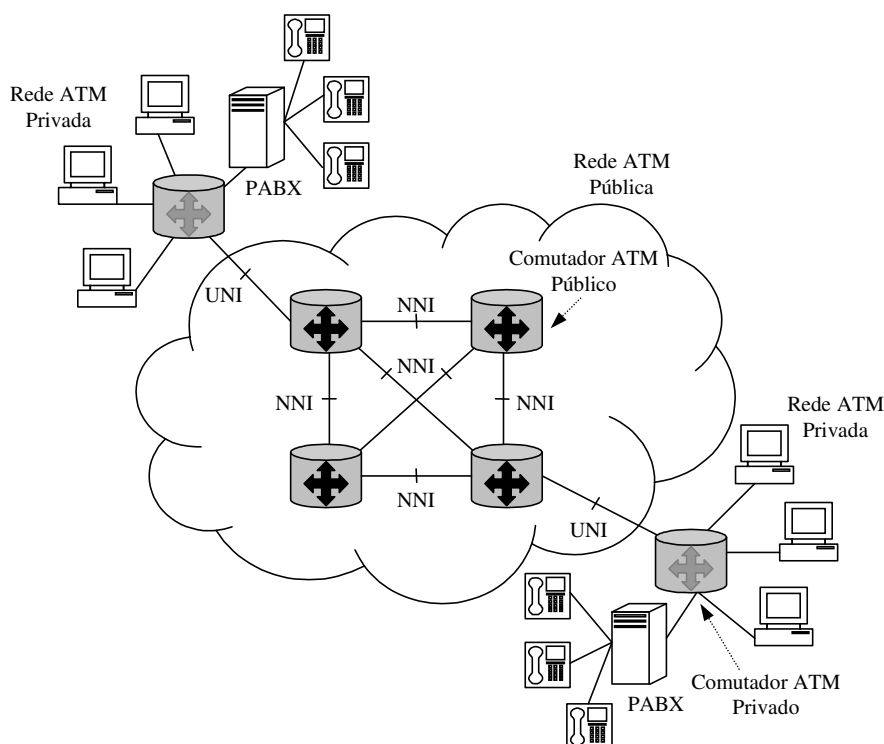


Figura 2.1 - Estrutura geral de uma rede ATM.

A idéia básica da tecnologia ATM é transmitir todas as informações em pequenos pacotes de tamanho fixo, chamados de células. As células têm 53 bytes, dos quais 5 bytes formam o cabeçalho e 48 bytes formam a carga (dados). Toda a informação (voz, vídeo, dados, etc.) é transportada pela rede através destas células ATM.

2.3 - Arquiteturas de Serviços Integrados (IntServ) [2]

A arquitetura de serviços integrados propõe um conjunto de extensões ao tradicional modelo de melhor esforço das redes IP, com o objetivo de oferecer qualidade de serviço (QoS) fim-a-fim para as aplicações, tanto através da comunicação um para um (“unicast”) quanto da comunicação multiponto (“multicast”).

A filosofia deste modelo é baseada em fluxos e supõe que seja necessário que os roteadores reservem recursos de modo a fornecer alguma garantia de QoS para os fluxos dos usuários. As reservas destes recursos são feitas de acordo com as requisições dos terminais de usuário. Um fluxo de dados identifica um conjunto de pacotes que receberá tratamento especial. Esses fluxos são definidos por sessão, as quais são identificadas por uma especificação genérica, contendo o endereço IP, o protocolo de transporte utilizado e o número da porta do destino, juntamente com uma lista de terminais transmissores para aquela sessão. Cada terminal transmissor é identificado pelo seu endereço IP e o número da porta, enquanto o protocolo de transporte que ele utiliza deve ser o mesmo definido na identificação da sessão.

A arquitetura supõe também que, para realizar essas reservas, algum mecanismo explícito de sinalização que leve as informações aos roteadores seja utilizado. O protocolo mais conhecido para este propósito, apesar da arquitetura poder acomodar outros, é o RSVP (*Resource Reservation Protocol*), o qual é independente da arquitetura IntServ.

O RSVP é um protocolo de sinalização que permite aos terminais requisitarem níveis de QoS específicos para suas aplicações. Ele também é utilizado pelos roteadores para entregar requisições de QoS para outros roteadores ao longo do caminho de um fluxo. As requisições do RSVP resultam, quando possível, em reservas de recursos na rede, de modo que esta possa prover o nível de QoS solicitado. O RSVP não é um protocolo de roteamento, ele utiliza os protocolos de roteamento da rede para determinar a rota a ser seguida entre origem e destino, e pode operar nos modos unicast ou multicast [3].

A abordagem do RSVP é baseada no terminal receptor, de modo que são estes terminais que escolhem o nível de recursos a serem reservados. A rede responde explicitamente, admitindo ou rejeitando essas requisições. As aplicações que possuem necessidades quantificadas de recursos podem expressar tais

necessidades através de parâmetros, como definido nas especificações dos respectivos serviços oferecidos pela arquitetura IntServ.

No processo de sinalização, ilustrado na Fig. 2.2, o transmissor (Terminal A) envia uma mensagem PATH para o receptor (Terminal B), especificando as características do tráfego. Após receber a mensagem PATH, o receptor responde com uma mensagem RESV, requisitando os recursos desejados. Cada nó intermediário ao longo do caminho pode rejeitar ou aceitar a requisição. Se a requisição for rejeitada, o roteador irá enviar uma mensagem de erro, RESVERR, para o receptor e o processo de sinalização é finalizado. Se a requisição é aceita, a largura de banda e o espaço necessário em *buffer* são alocados para o fluxo e as informações relacionadas ao fluxo são instaladas no roteador.

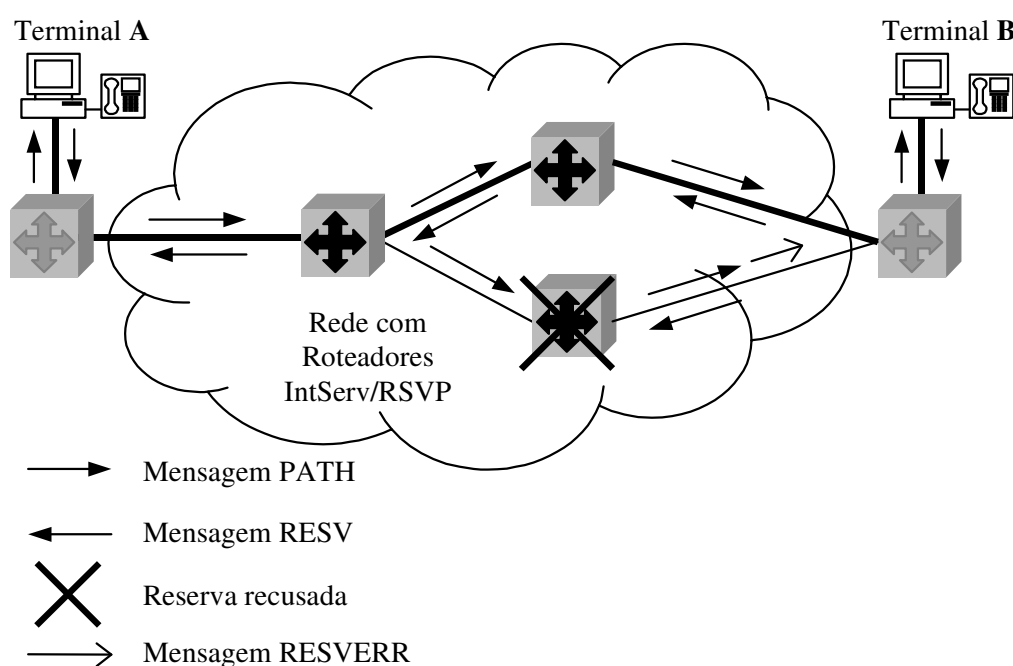


Figura 2.2 - Processo de sinalização RSVP numa arquitetura IntServ.

A arquitetura IntServ/RSVP representa uma mudança fundamental no modelo da Internet, o qual é baseado na premissa de que todas as informações de estado relacionadas ao fluxo devem ficar nas extremidades, nos sistemas dos usuários finais. No entanto, esta arquitetura é considerada excessivamente complexa e pouco escalável. Os principais problemas existentes com ela são a quantidade de informações de estado em cada roteador (que cresce proporcionalmente ao número de fluxos) e a carga excessiva sobre os roteadores. No primeiro caso, para cada novo fluxo que solicita reservas, informações de estado PATH

e RESV devem ser criadas em cada roteador ao longo do caminho, o que exige capacidade extra de armazenamento e processamento nos roteadores. No segundo caso, como cada roteador ao longo do caminho pode aceitar ou negar a solicitação de reservas, todos eles devem ser aptos a processar RSVP, realizar controle de admissão, classificar microfluxos e escalonar pacotes, o que acaba por sobrecarregar os mesmos.

2.4 - Arquitetura de Serviços Diferenciados (DiffServ) [5]

Em função dos problemas existentes com a arquitetura IntServ/RSVP, principalmente no que diz respeito a escalabilidade e à complexidade de implementação, uma outra proposta ganhou força, a arquitetura de serviços diferenciados. Em contraste com a orientação por fluxos do RSVP, as redes de serviços diferenciados (DiffServ) foram projetadas para oferecer classes de serviços agregados, onde os fluxos individuais são agrupados e tratados pela rede de acordo com a classe de serviços em que se encontram.

Para poder identificar a classe de um pacote, as redes DiffServ utilizam o campo DS (*Differentiated Service*), conhecido anteriormente como tipo de serviço (ToS - *Type of Service*), e situado dentro do cabeçalho dos pacotes IP. Desta forma, um novo tráfego que chega a uma rede DiffServ, nos roteadores de borda, é primeiro classificado, em seguida passa por um tipo de “filtro de admissão”, com o intuito de moldá-lo de acordo com a política de controle associada com aquela classificação. Este fluxo de tráfego é então atribuído a um comportamento agregado específico, através da marcação/remarcação apropriada do campo DS. A partir disso, em cada roteador DiffServ de núcleo, conforme mostrado na Fig. 2.3, os pacotes são sujeitos a um tratamento específico, chamado comportamento agregado por nó (PHB - *Per-Hop Behavior*), o qual dependerá do valor do campo DS dos pacotes [9].

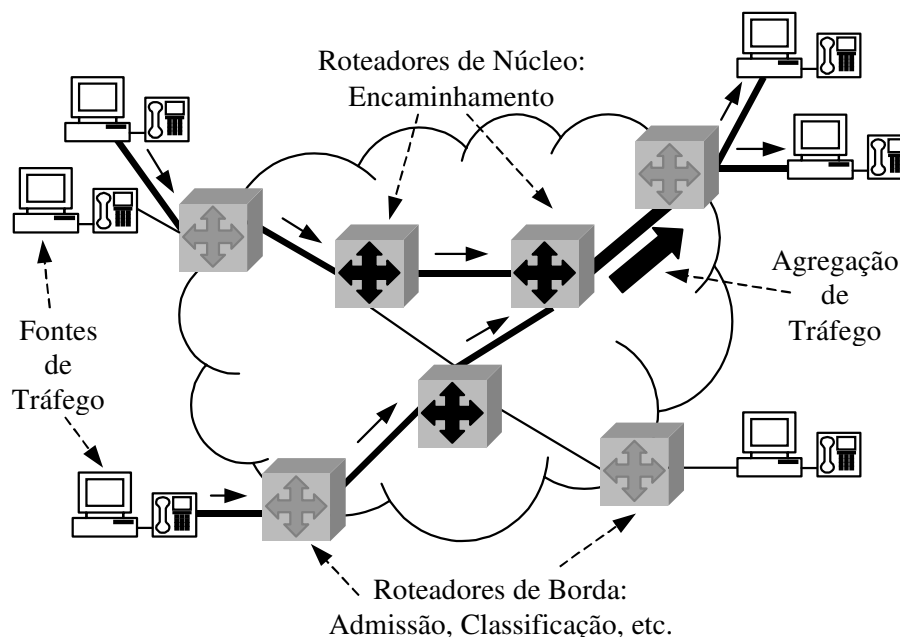


Figura 2.3 - A arquitetura de serviços diferenciados.

Para um consumidor poder usufruir os serviços diferenciados, a idéia é que ele deva definir um contrato com o seu provedor de serviços Internet (ISP), chamado de acordo do nível de serviço (SLA - *Service Level Agreement*). Os SLA's especificam basicamente as classes de serviço suportadas e a quantidade de tráfego permitida por classe. Definido o SLA entre o consumidor e o ISP, os pacotes oriundos da rede do consumidor devem vir com o campo DS marcado de acordo com a classe desejada, podendo esta marcação ser feita tanto pela aplicação do consumidor como pelo roteador de borda da rede. Ao ingressar na rede de serviços diferenciados os pacotes, como dito antes, são classificados, policiados e possivelmente moldados de acordo com o SLA. Quando um pacote passa de um domínio para outro, seu campo DS pode ser remarcado dependendo do que foi determinado pelo SLA entre os domínios.

A arquitetura de serviços diferenciados é significativamente diferente da IntServ. Primeiro, porque na DiffServ existe um número limitado de classes de serviço, indicadas pelo campo DS. Como os serviços são alocados de acordo com as classes, a quantidade de informações de estado é proporcional ao número de classes ao invés do número de fluxos, fazendo com que a arquitetura DiffServ seja mais escalável. Segundo, porque nela a classificação, a marcação, o policiamento e a moldagem são tarefas necessárias apenas nos elementos de borda das redes, tornando a arquitetura mais simples e fácil de implementar.

Apesar de mais escalável, em função de sua reduzida complexidade, a arquitetura DiffServ também possui alguns problemas, os quais são atribuídos justamente ao excesso de sua simplicidade. Entre as principais críticas à arquitetura DiffServ, pode-se destacar: a dificuldade de uma rede em identificar a política de ingresso de uma rede vizinha (quando o tráfego precisa ser passado através desta última), a dificuldade dos elementos de borda em aplicar técnicas de controle de admissão sem informações suficientes e atualizadas sobre a utilização dos recursos no seu respectivo domínio e a dificuldade de redes DiffServ em atender as características heterogêneas dos grupos multiponto.

2.5 - Multi-Protocol Label Switching (MPLS) [6] [10]

O MPLS é um esquema de encaminhamento de pacotes que evoluiu da tecnologia *Tag Switching* desenvolvida pela Cisco. No modelo OSI de 7 camadas ele se situa entre as camadas 2 (enlace) e 3 (rede). A Fig. 2.4 ilustra a arquitetura básica de uma rede implementando a tecnologia MPLS. O núcleo da rede é composto por LSR's (*Label Switched Router*). Um nó de borda é um LER (*Label Edge Router*), o qual pode ser um LER de ingresso - quando os pacotes entram no domínio MPLS - ou um LER de egresso - quando pacotes saem deste domínio.

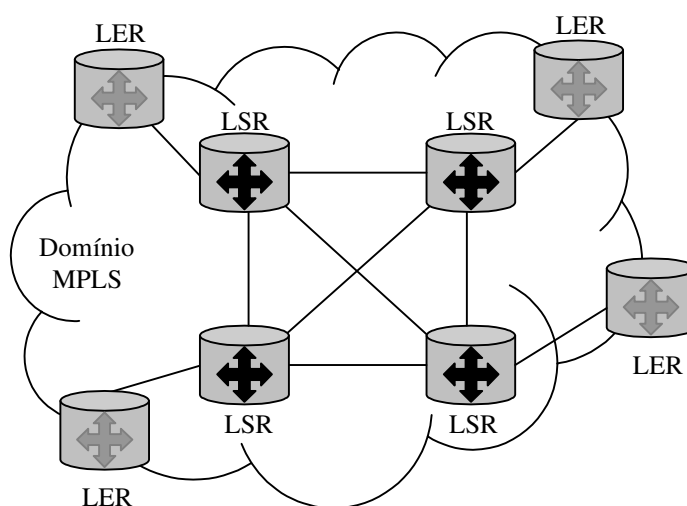


Figura 2.4 - Arquitetura básica de uma rede utilizando a tecnologia MPLS.

A essência do MPLS é a geração de um rótulo curto (o *label*), com comprimento fixo de 20 bits, um campo de Classe de Serviço (CoS - *Class of Service*) de 3 bits, um bit (B) que indica que o label é o último de uma pilha de labels e um campo TTL (*Time To Live*) de 8 bits, presentes num cabeçalho de 32 bits, conforme ilustrado na Fig. 2.5. O cabeçalho MPLS é encapsulado entre o cabeçalho da camada de enlace e o cabeçalho da camada de rede. Um roteador MPLS examina somente o label no encaminhamento do pacote.

Label - 20 bits	CoS - 3 bits	B - 1 bit	TTL - 8 bits
-----------------	--------------	-----------	--------------

Figura 2.5 - Cabeçalho MPLS.

O MPLS precisa de um protocolo para distribuir labels e para formar Caminhos Comutados por Label (LSP's - *Label Switched Paths*), que pode ser o protocolo LDP (*Label Distribution Protocol*) ou uma extensão do RSVP. Um LSP é similar a um circuito virtual ATM, e é unidirecional do transmissor para o receptor.

O rótulo colocado em um pacote identifica uma FEC (*Forwarding Equivalence Class*) atribuída ao pacote. As FEC's representam caminhos diferenciados, em função de classes de serviços, prioridades e QoS. Com os rótulos, os pacotes IP não precisam ser analisados a cada "hop" pertencente à conexão; são somente comutados. O rótulo proporciona uma melhora na eficiência e no controle da rede de roteadores, por permitir aos mesmos que encaminhem os pacotes com o uso de caminhos pré-determinados, de acordo, entre outras coisas, com níveis específicos de QoS. Separa-se o componente de encaminhamento do componente de roteamento. Ao se separarem esses componentes, um único algoritmo de encaminhamento pode ser utilizado para múltiplos serviços e tipos de tráfego. Na Fig. 2.6 é mostrado o processo de comutação de pacotes através de domínios MPLS.

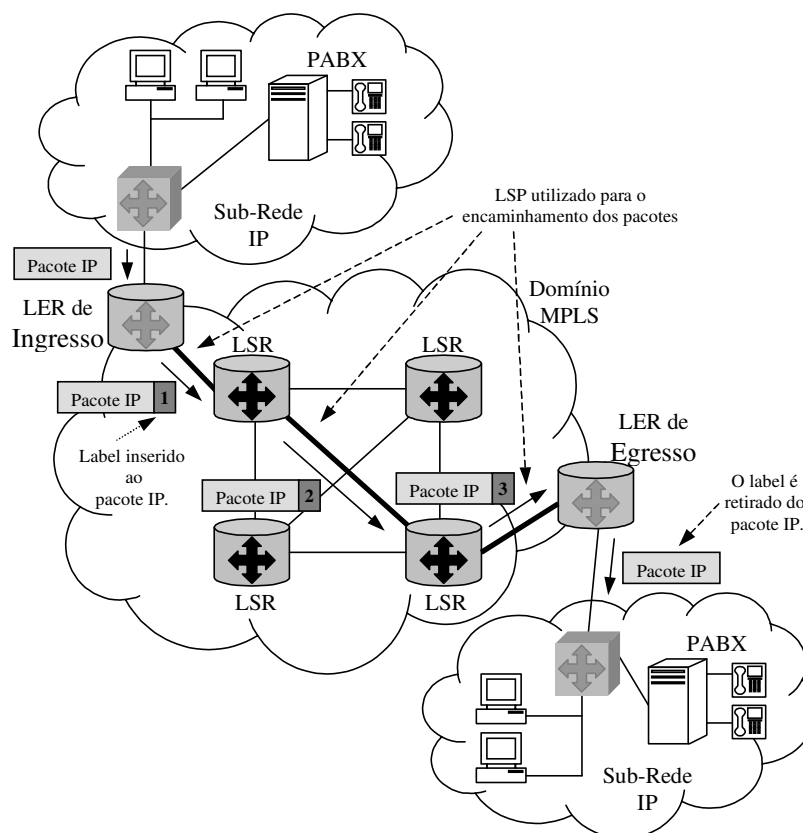


Figura 2.6 - Comutação dos pacotes dentro de um domínio MPLS.

Na Fig 2.6 os pacotes são classificados e roteados pelo LER de ingresso de um domínio MPLS, onde o primeiro cabeçalho MPLS é inserido. Quando os LSR's recebem os pacotes MPLS, o label é utilizado como índice para a consulta da tabela de encaminhamento (LIB - *Label Information Base*). Isto é mais rápido que o processo de roteamento IP normal, resultando em um menor atraso experimentado por cada pacote em cada roteador MPLS. O pacote é processado como especificado pela LIB. O label de entrada é substituído pelo label de saída e o pacote é comutado para o próximo LSR, através de um LSP previamente estabelecido no processo de distribuição de labels. Este processo é repetido até que o pacote MPLS alcance o último roteador do domínio MPLS (LER de egresso), onde o rótulo é retirado e o formato original do pacote é recuperado. Dentro do domínio MPLS, a classificação, encaminhamento e tratamento dos pacotes são definidos pelos campos de label e CoS.

2.6 - VoIP (Voz sobre IP) [11]

Voz sobre IP ou VoIP (*Voice over Internet Protocol*) é uma técnica de transmitir sinais de voz em redes que utilizam o protocolo IP. Os terminais utilizados em VoIP podem ser computadores (PC), telefones digitais (telefones IP) e telefones convencionais analógicos. Para uma primeira abordagem, são definidas três arquiteturas básicas de implementação de VoIP:

- **Arquitetura PC-a-PC:** Nesta arquitetura dois computadores providos de recursos multimídia (ou telefones IP), conectados a uma LAN ou, através da PSTN (Rede Pública de Telefonia), a um provedor de serviços Internet, se comunicam para troca de sinais de voz. Todo o tratamento do sinal de voz (amostragem, compressão e empacotamento) é realizado nos computadores, sendo a chamada de voz estabelecida com base no endereço IP de destino.
- **Arquitetura com gateway:** Nesta arquitetura um telefone convencional é utilizado para gerar e receber a chamada telefônica sobre a Internet. O usuário chamador (telefone de origem) disca para o gateway mais próximo de sua central telefônica local (gateway origem). O gateway de origem solicita ao usuário chamador o número do usuário chamado (telefone de destino). O gateway de origem, após localizar um gateway próximo ao telefone chamado (gateway de destino), inicia uma sessão para transmissão de pacotes de voz. O gateway de destino chama o telefone de destino e, após a chamada ser atendida, a comunicação fim-a-fim tem início, com o sinal de voz sendo enviado através de datagramas IP entre os gateways. A codificação/decodificação e o empacotamento/desempacotamento dos sinais de voz são feitos nos gateways de origem e de destino.
- **Arquitetura híbrida:** Naturalmente, esquemas híbridos com as duas arquiteturas anteriores são desejáveis. Nas estruturas híbridas um usuário de um telefone convencional origina (ou recebe) uma chamada para um usuário de PC (ou telefone IP). Em tais situações, deve haver um serviço de mapeamento ou tradução de endereços IP em números telefônicos. Existem quatro caminhos unidirecionais neste caso: PC-a-PC, gateway-a-gateway, PC-a-gateway, gateway-a-PC. Em todas estas arquiteturas os pontos terminais (PC ou gateways) devem empregar o mesmo esquema de codificação de voz.

2.6.1 - VoIP através da Internet

O protocolo IP oferece um serviço sem orientação à conexão baseado em datagramas. Este tipo de serviço não garante que os datagramas IP sejam entregues a tempo, que eles cheguem ao destino na ordem correta e nem que cheguem ao destino. Os roteadores operam segundo um serviço de melhor esforço. Neste tipo de operação, a rede tenta encaminhar todos os pacotes o mais rápido possível, mas não pode dar qualquer garantia quantitativa sobre a qualidade de serviço.

Os requisitos de QoS para VoIP dificultam sobremaneira, se não impedem, a implementação desta aplicação diretamente sobre a Internet, ao menos em sua versão atual. No entanto, com a utilização de técnicas de provimento de QoS em redes IP (tais como as apresentadas no próximo capítulo) a implementação de VoIP através da Internet pode se tornar possível.

VoIP constitui uma boa alternativa para a implantação de redes multimídia corporativas, ou mesmo para a implementação de *backbones*, com capacidade para integração de tráfego. A Fig. 2.7 ilustra um cenário onde redes corporativas, usuárias de VoIP, podem ser beneficiadas através da interconexão com provedores de serviços de Internet (ISP - *Internet Service Provider*) que garantam um caminho fim-a-fim com QoS mínima.

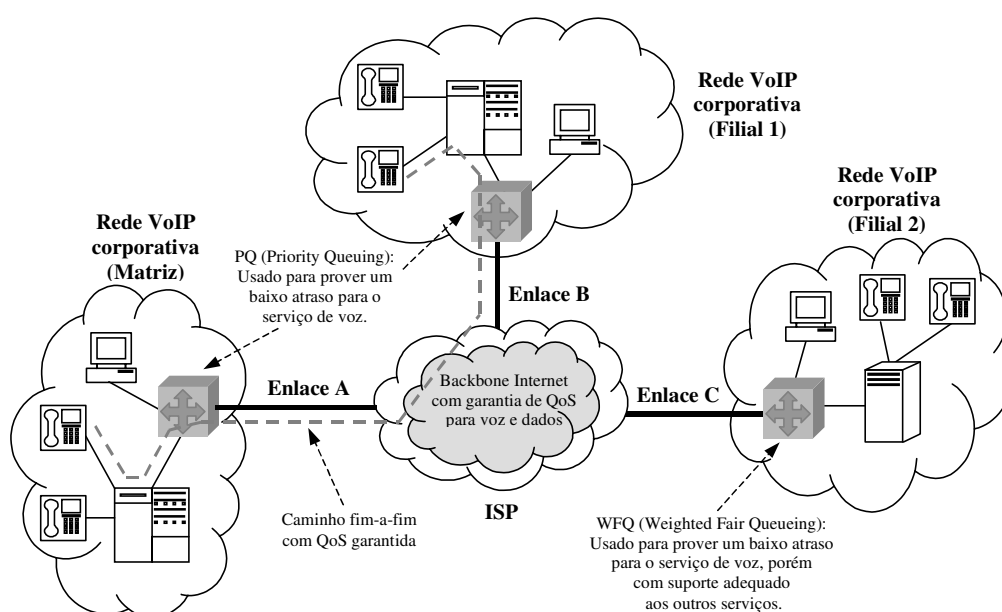


Figura 2.7 - Provendo QoS para redes corporativas com VoIP.

Com relação à Fig. 2.7, no ambiente corporativo o uso da arquitetura IntServ pode ser uma boa alternativa. No *backbone*, devido ao grande número de fluxos, a arquitetura DiffServ, possivelmente associada ao uso de MPLS com roteamento explícito, pode ser uma opção. Em qualquer caso, o uso de algoritmos de escalonamento que priorizem o tráfego de voz (tais como PQ e WFQ, mostrados no item 2.7.2) são indispensáveis.

2.7 - Qualidade de Serviço

A qualidade de serviço (QoS - *Quality of Service*) pode ser definida como a “habilidade da rede para garantir e manter certos níveis de desempenho para cada aplicação, de acordo com as necessidades específicas de cada usuário” [12].

Embora o conceito de QoS usualmente se refira à fidelidade do sinal de voz recebido, ele também pode se aplicar a outros aspectos, tais como: disponibilidade da rede, probabilidade de bloqueio, existência de serviços especiais (conferência, identificação do usuário chamador, etc), escalabilidade e penetração. Neste item é tratada apenas a questão da qualidade no que diz respeito à fidelidade do sinal recebido.

2.7.1 - Fatores que Influenciam a Qualidade do Sinal de Voz

A qualidade de reprodução de voz na rede telefônica é fundamentalmente subjetiva, embora medidas padrões tenham sido desenvolvidas pelo ITU-T. Para a transmissão de voz sobre redes de pacotes existem três fatores principais que impactam a qualidade do serviço: largura de banda, atraso (fim-a-fim) de pacotes e perda de pacotes.

2.7.1.1- Largura de Banda

A largura de banda mínima necessária para a transmissão do sinal de voz é função da técnica de codificação utilizada. A largura de banda disponível na rede e o mecanismo de compartilhamento desta,

entre as diversas aplicações, tem influência direta no atraso sofrido pelo pacote e conseqüentemente na qualidade de serviço resultante.

2.7.1.2 - Atraso de Pacote

O atraso de pacote é formalmente definido como a diferença de tempo entre o instante em que o terminal chamador envia o primeiro bit do pacote e o instante em que o terminal chamado recebe este bit. Seu comportamento é aleatório em função da carga na rede. O principal problema associado a este atraso é conhecido como *jitter* de atraso.

O jitter é a variação no intervalo entre chegadas de pacotes, introduzida pelo comportamento aleatório do atraso na rede. Um método típico de contornar o problema do jitter é adicionar um *buffer* na recepção, o qual acrescenta um atraso previamente determinado, de tal forma que o atraso total experimentado pelo pacote seja igual ao máximo atraso possível na rede. Este método é razoável no ambiente de redes locais ou Intranets corporativas onde o atraso máximo é pequeno. Em redes WAN onde o atraso máximo pode assumir valores inaceitavelmente elevados, este método não é aplicável. A escolha do máximo atraso a ser introduzido pelo *buffer* na recepção é então uma solução de compromisso entre o atraso total admissível e a taxa de perda de pacotes [13].

O atraso dos pacotes possui diversas componentes, algumas de natureza fixa e outras de natureza variável, que são brevemente descritas a seguir:

Atraso de Propagação: É o atraso relacionado com o tempo que o sinal leva para se propagar no meio de transmissão (par metálico, fibra óptica, espaço livre, etc). Este atraso é fixo e depende do tipo do meio e da distância percorrida pelo sinal. Por exemplo, para uma transmissão via rádio, o atraso de propagação é de 3,33 $\mu\text{s/Km}$.

Atraso de Codificação/Decodificação: Este atraso é fixo e é composto basicamente de três parcelas: tamanho do quadro e “*lookahead delay*”, que juntos compõem o chamado atraso algorítmico, e o atraso de processamento. A Fig. 2.8 mostra os valores destes atrasos para três tipos de codificadores comuns para transmissão de voz sobre redes de pacotes [14].

	CODECS		
	G 723.1	G 729	G 729 A
Taxa de bits	5,3 / 6,4 Kbps	8 Kbps	8 Kbps
Tamanho de quadro	30 ms	10 ms	10 ms
Atraso de processamento	30 ms	10 ms	10 ms
Lookahead delay	7,5 ms	5 ms	5 ms
Comprimento do quadro	20/24 bytes	10 bytes	10 bytes
DSP MIPS	16	20	10,5
RAM	2200	3000	2000

Figura 2.8 - Algumas características de alguns Codecs aplicáveis a VoIP.

Na Fig. 2.8, o comprimento do quadro corresponde ao número de bytes em um quadro codificado (excluindo o cabeçalho); o parâmetro DSP MIPS indica a mínima velocidade necessária ao processador DSP para implementar o algoritmo de codificação; e o parâmetro RAM especifica a memória mínima requerida, em palavras de 16 bits.

Atraso de Empacotamento: É o tempo necessário para se gerar um número suficiente de quadros de voz para preencher o payload do pacote IP, por exemplo. Para se evitar valores excessivos para o atraso de empacotamento, pode-se enviar pacotes parcialmente carregados. Deve-se notar, no entanto, que isto reduz a eficiência do sistema. O atraso de empacotamento pode, dependendo da situação, absorver ou se confundir com atrasos de codificação.

Atraso nos Nós da Rede: O principal atraso que os pacotes sofrem dentro da rede é o atraso de enfileiramento nos roteadores (ou comutadores de pacotes). Este atraso é variável e depende do tempo necessário para o roteador tomar a decisão de roteamento, do tempo de transferência do pacote do *buffer* de entrada para o *buffer* de saída (tempo de chaveamento), do tempo de transmissão do pacote no enlace de saída e do fator de utilização do enlace de saída associado (carga). O atraso de enfileiramento é o principal responsável pela aleatoriedade do comportamento do atraso total experimentado pelo pacote, podendo assumir valores inaceitáveis em situações de congestionamento na rede.

Atraso Devido ao Dejitter Buffer: A variação do atraso (*jitter*) é introduzida no sistema basicamente pelo comportamento aleatório do tempo de enfileiramento dos pacotes nos roteadores, e é um

fator de degradação da qualidade do sinal. A compensação desta variação é feita através de *buffers* (*de jitter buffers*) que armazenam os pacotes que chegam com atraso variável, para entregá-los ao decodificador com atrasos constantes. No entanto, se a variação do atraso for muito elevada, o atraso adicional necessário para compensar a variação pode resultar em um atraso total fim-a-fim inaceitável. Por esta razão, define-se um valor máximo de atraso admissível para o *de jitter buffer*.

2.7.1.3 - Perda de Pacotes

As redes IP não garantem a entrega dos pacotes. Devido aos fortes requisitos de atraso impostos pelas aplicações interativas em tempo real, protocolos de transporte confiáveis, como o TCP, não podem ser utilizados. Portanto, a perda de pacotes é inevitável e pode influenciar significativamente a qualidade do serviço de voz. A perda de pacotes é definida como a percentagem de pacotes transmitidos que não chegam ao destino, devido principalmente a:

- **Imperfeições na transmissão:** problemas físicos nos equipamentos de transmissão podem resultar em perdas de pacotes.
- **Atraso excessivo:** se o parâmetro TTL (*Time-To-Live*) definido para o pacote for excedido, o pacote é descartado pela rede.
- **Congestionamento:** o aumento em excesso do tráfego na rede pode resultar no *overflow* dos *buffers* dos roteadores, resultando na perda de pacotes. Além disto, se o protocolo RED (*Random Early Detection*) é utilizado, o roteador irá descartar aleatoriamente pacotes.
- **Overflow do buffer de de jitter:** se o *jitter* na rede for excessivo, poderá ocorrer um *overflow* no *buffer* utilizado para compensar este *jitter*, com conseqüente perda de pacotes.

As perdas de pacotes de voz serão percebidas como “*gaps*” na conversação, degradando a qualidade do serviço. Contudo, uma certa percentagem de perda de pacotes, entre 3 e 5% [13], pode ser compensada por esquemas de recuperação dos Codecs. Por exemplo, o G.723.1 interpola um quadro perdido simulando as características vocais do quadro anterior e reduzindo lentamente o sinal [14]. A taxa máxima tolerável de perda de pacotes de voz é usualmente fixada em 10% [14]. A Fig. 2.9 mostra os limites aceitáveis para o atraso e percentagem de perda de pacotes para VoIP.

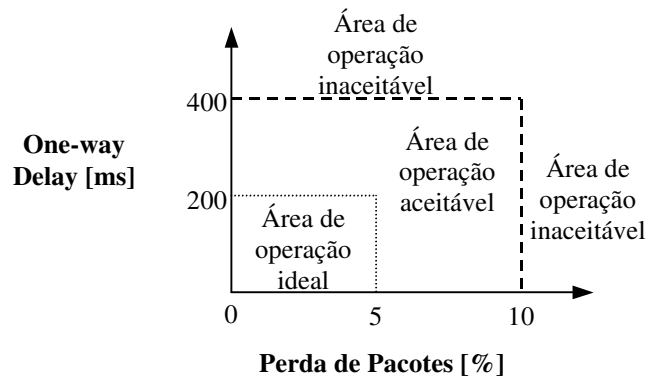


Figura 2.9 - Limites aceitáveis de atraso e perda de pacotes para VoIP.

Embora a perda de pacotes isolados, dentro de determinados limites, seja de menor consequência devido aos esquemas de recuperação utilizados pelos Codecs, perdas de pacotes em *bursts*, como as produzidas pela Internet, podem causar grande degradação no sinal recebido. Técnicas de correção automática de erros no receptor (FEC - *Forward Error Correction*) têm sido propostas para contornar perdas de pacotes em *bursts* de tamanho reduzido, mas ainda precisam ser melhor investigadas [14]. A desvantagem de se utilizar técnicas de FEC é que para recuperar o pacote n (onde n é o número de seqüência do pacote) é necessário que já se tenha recebido ao menos o pacote $n+1$. Assim, o esquema de FEC introduz um atraso adicional no mínimo equivalente ao tempo de um quadro, além do atraso de processamento para o processo de codificação e de decodificação. Estes atrasos adicionais podem fazer com que o quadro seja recuperado tarde demais, ou seja, o atraso total pode exceder o limite aceitável, resultando no descarte do pacote [14].

2.7.2 - Mecanismos para Prover QoS ao Tráfego de Voz em Redes IP

Para se alcançar um nível de QoS adequado para o tráfego de voz sobre uma rede IP pode-se adotar um conjunto de medidas no sentido de: garantir a banda necessária para a transmissão dos pacotes de voz (ex.: protocolos de reserva de recursos), minimizar os atrasos sofridos pelos pacotes na rede e torná-los o mais constante possível (ex.: utilizar mecanismos para priorização dos pacotes de voz, utilizar técnicas de roteamento que privilegiam as rotas de menor atraso, utilizar mecanismos mais eficientes para o encaminhamento dos pacotes nos roteadores), e eliminar ou minimizar o *jitter* de atraso sofrido pelos pacotes (ex.: utilizar *de jitter buffer*).

São exemplos de mecanismos para prover QoS em redes IP transportando tráfego de voz: *Dejitter Buffer* [12], [14]; Classificação ou Identificação do Tráfego e Enfileiramento [15], [16]; Priorização e Disciplina de escalonamento. Neste trabalho é discutido apenas o último mecanismo. Em geral todas estas técnicas para prover QoS estão associadas às seguintes funções [17]:

- Classificação do Tráfego, de modo a poder diferenciar um tipo de outro;
- Priorização dos pacotes de tráfego de voz;
- Policiamento e conformação do tráfego;
- Gerenciamento de congestionamento;
- Fragmentação de grandes pacotes de dados e entrelaçamento destes pacotes com os pacotes de voz;
- Garantia de largura de faixa para o tráfego de voz;
- Compensação, no receptor, da variação do atraso na rede.

2.7.2.1 - Enfileiramento, Priorização e Disciplina de Escalonamento

De modo a absorver situações momentâneas de congestionamento na rede, onde a taxa de chegada de pacotes excede a capacidade do enlace de saída, os nós de uma rede de pacotes (ex.: roteadores em uma rede IP) possuem *buffers* especiais para armazenamento temporário dos pacotes. A disciplina de escalonamento define a forma como o nó da rede irá servir os pacotes armazenados nestes *buffers*. Quando a rede transporta simultaneamente tráfego de voz e dados, deve-se associar níveis distintos de prioridade aos dois tipos de tráfego, com a disciplina de escalonamento priorizando o tráfego de voz, de modo a minimizar o atraso que estes pacotes sofrem em cada nó da rede.

2.7.2.1.1 - First-In, First-Out - FIFO

Esta é a disciplina de escalonamento mais simples. Nenhum conceito de prioridade ou classe de tráfego é utilizado, com todos os pacotes sendo tratados igualmente. Existe uma única fila de saída, os pacotes recebidos são armazenados e enviados na mesma ordem em que chegaram.

Neste tipo de fila, fontes de tráfego mal comportadas podem consumir toda a largura de banda disponível, tráfegos em rajada podem causar atrasos inaceitáveis em tráfegos sensíveis a atraso e pacotes pertencentes a tráfegos de maior importância podem ser perdidos devido a *overflow* no *buffer*, causado possivelmente por tráfegos de menor importância.

2.7.2.1.2 - Fila com Prioridade (Priority Queueing - PQ)

Nesta técnica existem filas distintas para diferentes classes de prioridades. Por exemplo, podemos ter quatro níveis de prioridade (alta, média, normal e baixa), com uma fila associada a cada nível, conforme ilustra a Fig. 2.10.

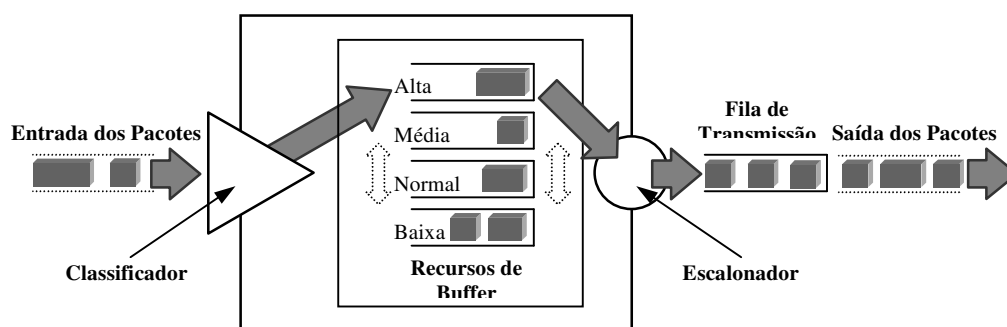


Figura 2.10 - Fila com quatro níveis de prioridade.

Os pacotes, uma vez classificados, são encaminhados para a fila com nível de prioridade correspondente. Durante a transmissão, o algoritmo de escalonamento dá tratamento preferencial absoluto à fila de maior prioridade, em detrimento das filas de menor prioridade. Ou seja, no momento da transmissão, o algoritmo sempre busca um pacote na fila de alta prioridade; caso não haja pacote nesta fila, um pacote da fila de média prioridade é buscado, e assim sucessivamente até se chegar à fila de baixa prioridade.

Um volume de tráfego de maior prioridade pode reter os tráfegos de menor prioridade por um tempo inaceitavelmente elevado; este fenômeno é conhecido como “*starvation*”. No pior caso, devido a um grande volume de tráfego de alta prioridade, por exemplo, o tráfego de baixa prioridade pode nunca ser transmitido. Para se evitar esta situação, pode-se utilizar ferramentas de formatação de tráfego ou CAR (*Committed Access Rate*), de modo a restringir a taxa de chegada de tráfego de alta prioridade [18].

2.7.2.1.3 - Custom Queueing - CQ

Nesta técnica, tem-se uma fila para cada tipo de tráfego especificado. As filas são servidas de forma cíclica (*round-robin*), permitindo-se que se especifique o número de pacotes (ou bytes) a serem transmitidos, de cada fila, a cada ciclo de serviço. Desta forma, pode-se especificar a largura de banda mínima do canal disponível para cada tipo de tráfego e, considerando a existência de pacotes para todos os tráfegos especificados, o valor aproximado do percentual da banda do canal utilizado por cada tipo de tráfego. A largura de banda não utilizada por uma fila (por ausência de tráfego) é naturalmente distribuída para as demais, pelo mecanismo de escalonamento dos pacotes.

Um exemplo de implementação do algoritmo Custom Queueing, feito pela Cisco, é mostrado na Fig. 2.11 e descrito a seguir [18].

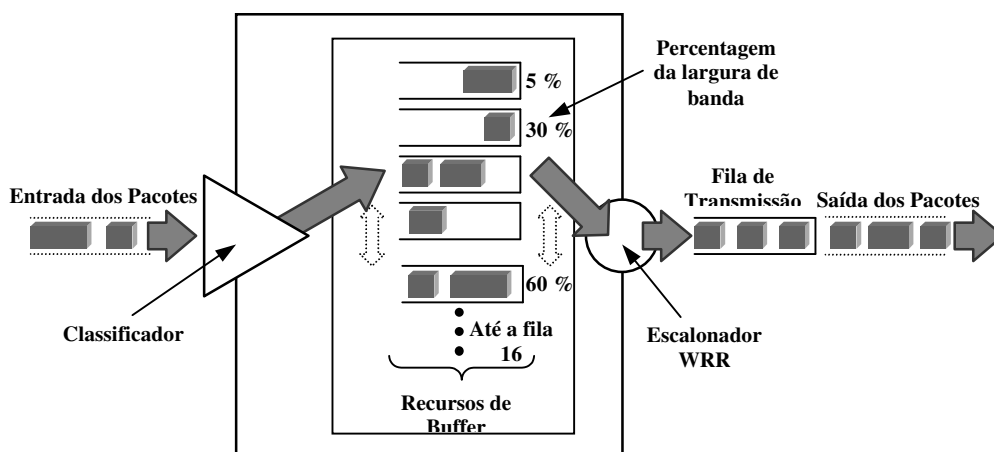


Figura 2.11 - Custom Queueing.

- O sistema mantém 17 filas, sendo uma para tráfego do sistema (fila 0) e 16 configuráveis pelo usuário.
- Associado a cada uma das 16 filas configuráveis existe um contador (programável) que especifica quantos bytes serão transmitidos da fila a cada ciclo de serviço. Pode-se especificar também o número máximo de pacotes em cada fila.
- Um ciclo de serviço é compreendido pelo atendimento sequencial (*round-robin*) de todas as 17 filas existentes.

- A primeira fila atendida é a fila 0, que é utilizada para transpotar pacotes de alta prioridade do sistema, como mensagem de *keepalive* e de sinalização. Todos os pacotes da fila 0 são transmitidos antes que a fila 1 seja atendida.
- Quando uma das filas configuráveis está sendo atendida, os pacotes são enviados até que o número de bytes transmitido exceda o valor do contador de bytes associado, ou a fila fique vazia. Quando o número de bytes transmitidos excede o valor do contador, o pacote corrente é transmitido até o fim.
- As filas são estaticamente configuradas, não respondendo automaticamente a mudanças nas condições da rede.

2.7.2.1.4 - Weighted Round Robin - WRR

Na técnica WRR associa-se um peso a cada classe de tráfego, possivelmente com base no conteúdo do campo ToS do cabeçalho IP. Este peso é utilizado para determinar qual percentual da banda do enlace será alocado para cada classe de tráfego, de acordo com a seguinte fórmula: $P_i = (W_i/S) \times B$. Onde P_i representa a taxa (em bps) alocada para o tráfego de classe i ; W_i representa o peso associado à classe de tráfego i ; S é o somatório dos pesos atribuídos a todas as classes de tráfego; e B é a banda total do enlace [17].

As filas são servidas em ordem decrescente de prioridade (peso). Entretanto, contrário à operação da disciplina PQ, onde uma fila é atendida somente após o atendimento pleno de todas as filas de maior prioridade, na técnica WRR o serviço passa para a próxima fila se a fila corrente ficar vazia ou se o percentual de banda atribuído a ela for ultrapassado. O controle de banda é feito através de um contador de bytes associados a cada fila, que indica o número de bytes que serão transmitidos a cada ciclo de serviço, de forma semelhante à disciplina CQ [17], [19].

2.7.2.1.5 - Weighted Fair Queueing - WFQ

Antes de descrever a operação da disciplina de escalonamento WFQ, é importante descrever o algoritmo denominado BRR (*Bit-by-bit Round Robin*), no qual o WFQ se inspira. No BRR cada fluxo é mantido em uma fila de saída exclusiva e um bit de cada fluxo é enviado pelo enlace de saída a cada ciclo

de serviço. O BRR, portanto divide a banda do canal igualmente entre os N fluxos existentes, atribuindo a cada um a fração de $1/N$ da banda do canal. Esta abordagem faz com que o BRR seja um algoritmo ótimo pelo ponto de vista de equidade da distribuição dos recursos do canal entre os diversos fluxos, mas de implementação inviável, em função do *overhead* resultante de se transmitir um único bit por vez pelo enlace [19].

Uma disciplina de escalonamento denominada FQ (*Fair Queueing*) simula o algoritmo BRR sem a restrição de servir as filas bit-a-bit. No FQ existe uma única fila de saída. Quando um pacote é recebido, o algoritmo FQ calcula o instante de tempo (t_p) em que o último bit deste pacote seria transmitido se o algoritmo BRR estivesse sendo utilizado. Este instante de tempo é tratado como um parâmetro t_p , em ordem crescente, ou seja, pacotes com menor valor de t_p são posicionados na fila à frente de outros pacotes com maior valor de t_p . Os pacotes são transmitidos um a um, integralmente, pelo enlace de saída, e não mais bit-a-bit como na técnica BRR [19].

A técnica WFQ possui o mesmo princípio da técnica FQ, oferecendo a possibilidade de se diferenciar um fluxo do outro, através de um peso atribuído a cada fluxo. Este peso é utilizado no momento de se calcular o valor de t_p associado a cada pacote. Ao simular o BRR, o WFQ supõe, para cada pacote, a transmissão de uma quantidade de bits proporcional ao peso associado ao fluxo (e não um único bit como no caso do FQ) [19]. Portanto, o WFQ não faz uma divisão igual da banda entre os fluxos, mas atribui um maior percentual de banda para fluxos com maior peso.

A Fig. 2.12 ilustra um sistema com duas filas, uma para tráfego de voz e outra para os fluxos de dados; para o tráfego de voz aloca-se um mínimo de 80 kbps, dos 128 kbps disponíveis no enlace.

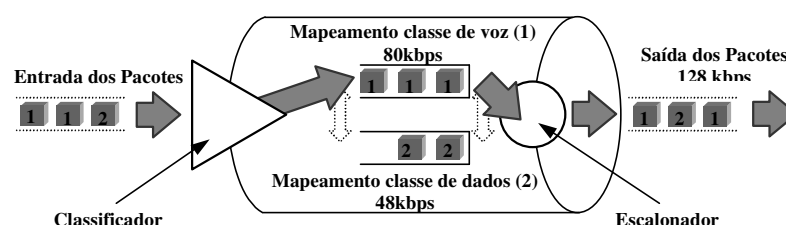


Figura 2.12 - Exemplo de compartilhamento de tráfego de voz e dados utilizando WFQ.

Deve-se observar que, para se garantir a banda mínima para o tráfego de voz, o peso atribuído a este tipo de tráfego deve se ajustar dinamicamente com o perfil de tráfego da rede. Por exemplo, se o número de fluxos de dados cresce, em um dado instante, é preciso que se aumente o peso do tráfego de voz (ou diminua o peso do tráfego de dados) para que se mantenha a distribuição de banda anterior.

2.8 - Dimensionamento de Enlaces em Redes de Pacotes

Trabalhos relacionados ao dimensionamento de enlaces em redes receberam um novo impulso graças à tendência, cada vez maior, das redes de telecomunicações tradicionais migrarem para redes com capacidade para integração de serviços. Em redes com integração de serviços, tipos diferentes de tráfego demandam necessidades distintas de QoS (ex.: atraso médio e probabilidade de perda de pacotes) . Para atender os níveis de QoS requisitados por esses tráfegos, o dimensionamento adequado da capacidade dos enlaces é de primordial importância. Nos parágrafos seguintes, estão mostradas algumas técnicas para a obtenção da capacidade de transmissão de enlaces em redes de comutação de pacotes.

Uma das técnicas mais simples de dimensionamento leva em conta apenas o atraso médio dos pacotes, considerando para tanto o modelo de filas M/M/1. Nesse método é feito o cálculo para a melhor alocação da capacidade, enlace por enlace, no sentido de minimizar o atraso médio, ou seja, é feita a minimização dos atrasos fim-a-fim a partir da distribuição adequada da capacidade de transmissão global da rede entre os enlaces. Esse método pode ser utilizado para se encontrar uma primeira aproximação para a capacidade de transmissão dos enlaces [20].

Considerando que modelos analíticos de tráfego sejam conhecidos, é sempre possível se obter um valor aproximado para a capacidade dos enlaces através de tentativas sucessivas de cálculo, ou seja, a partir de um dado parâmetro de restrição (ex.: atraso médio) tenta-se encontrar uma capacidade que a satisfaça. Uma representação sistemática para isso pode ser dada por:

$$\left| E\{T\} - E'\{T\} \right| \leq \varepsilon \quad (2.1)$$

onde, $E\{T\}$ é o atraso médio requisitado, $E'\{T\}$ é o atraso encontrado a partir de uma capacidade aleatória e ε é a precisão desejada. A capacidade de transmissão, escolhida aleatoriamente, que atender o critério da Eq. 2.1 pode ser escolhida como capacidade para o enlace.

Técnicas através de tentativas e erros são boas opções para o dimensionamento quando expressões analíticas de desempenho de redes são conhecidas, mas nem sempre isso é possível. Neste caso, o dimensionamento pode ser feito recorrendo-se ao auxílio de ferramentas de simulação de eventos discretos. A partir de simulações sucessivas, considerando capacidades de transmissões aleatórias, tenta-se encontrar aproximações para a capacidade do enlace, as quais satisfaçam as requisições de QoS, como por exemplo, a da Eq. 2.1.

As chegadas poissonianas e tempo de serviço exponencial têm servido como excelentes modelos para tráfego de voz comutado por circuito. As chegadas poissonianas e tamanho de pacotes exponenciais têm sido usados extensivamente no estudo do desempenho de redes de comutação de pacotes [21]. Outro exemplo de modelo para tráfego de voz em redes de pacotes é o que considera a chegada dos pacotes conforme um processo de Poisson modulado por Markov (MMPP). Uma expressão analítica, considerando apenas a probabilidade de perda de pacotes, foi desenvolvida com a finalidade de dimensionamento de enlaces em redes de telefonia sobre IP [22]. Esses modelos, bem estabelecidos para tráfegos homogêneos, não são suficientes para a análise de redes com integração de serviços. A integração de voz, vídeo e dados, cada qual com sua própria requisição de QoS, solicita o desenvolvimento de novos modelos de tráfego [21].

O dimensionamento de enlaces em redes com integração de serviços foi bastante estudado considerando-se redes ATM, destacando-se a idéia de banda efetiva nas suas mais variadas formulações. O cálculo da banda efetiva depende de hipóteses assumidas em relação à classe de serviço - se possui ou não um tráfego de pico limitado ou uma duração máxima do surto limitada, etc. Cada uma das fontes de tráfego é, então, servida pela respectiva banda efetiva, sem comprometer o tráfego das demais fontes que compartilham o enlace [21]. A capacidade de transmissão do enlace é dada, então, pelo agregado das bandas efetivas de cada tráfego transportado pelo enlace.

2.9 - Conclusões

Neste capítulo, além de apresentar cinco das principais técnicas para redes com suporte à integração de serviços (ATM, IntServ, DiffServ, MPLS e VoIP), foram discutidos também alguns dos requisitos de QoS (ex.: perda e atraso de pacotes), exigidos pelos diversos tipos de tráfegos, e alguns métodos utilizados para o provimento destes (ex.: disciplina de escalonamento WFQ). No item 2.8, foram mostradas algumas estratégias utilizadas para o dimensionamento da capacidade dos enlaces em redes de comutação de pacotes.

Em redes com integração de serviços, além da complexidade no tratamento de múltiplos serviços, exige-se também um controle de QoS para garantir a integridade de cada um destes serviços. Essa combinação de tratamento múltiplo de serviços e de controle de QoS faz com que os projetos destas redes enfrentem desafios não observados nas redes tradicionais. Por exemplo, o dimensionamento de enlaces em redes de telefonia tradicionais é baseado somente no critério de perda de chamadas, admitindo que todas as chamadas necessitam de largura de banda fixa.

No caso de redes com integração de serviços, o dimensionamento de enlaces é mais complexo, pois deve-se levar em conta a existência de diferentes tipos de serviços, exigindo diferentes larguras de bandas e diferentes níveis de qualidade de serviços. Assim, o dimensionamento de enlaces destas redes, levando-se em conta todos os parâmetros de restrição (largura de banda, QoS, topologia da rede, custos, etc), torna-se bastante complexo.

O critério de dimensionamento de enlaces para redes com integração de serviços proposto neste trabalho, baseia-se somente em dois principais parâmetros de QoS: o atraso médio dos pacotes nos nós da rede, devido ao enfileiramento, e a taxa de perda de pacotes, devido ao *overflow* nos *buffers*. Para a disciplina de atendimento dos pacotes, ou escalonamento, são utilizados dois tipos - FIFO e filas com prioridade.

Capítulo 3

Modelos de Dimensionamento de Enlaces

3.1 - Introdução

Em redes usando comutação de circuitos a garantia de QoS pode ser conseguida facilmente. Há algumas fórmulas matemáticas bem conhecidas, tais como a fórmula de Erlang, para calcular a capacidade da rede. A situação é um tanto diferente em redes por comutação de pacotes. Nestas redes, o comutador/roteador deve utilizar eficientemente a capacidade do enlace. Neste sentido, modelos de dimensionamento têm essencial importância na determinação da capacidade dos enlaces da rede. Estes devem levar em conta, ao mesmo tempo, a minimização de custos pré-definidos e a garantia de QoS mínima a ser provida pela rede.

No projeto de redes com integração de serviços, a complexidade do dimensionamento está relacionada com a diversidade dos parâmetros de QoS (atraso, jitter, perda de pacotes, etc). A fim de simplificar o cálculo da largura de banda mínima de um determinado enlace, a escolha de apenas alguns destes parâmetros pode ser feita. Um exemplo desta simplificação é visto no modelo conhecido como “estratégia da raiz quadrada”. Este se baseia na divisão da largura de banda global da rede, enlace por enlace, considerando apenas a minimização dos atrasos fim-a-fim [20].

Neste capítulo são propostos dois modelos de dimensionamento da capacidade de enlaces em redes de pacotes, levando-se em conta os parâmetros de atraso e taxa de perda dos pacotes definidos. O primeiro modelo refere-se ao dimensionamento onde o escalonador transmite os pacotes armazenados numa fila limitada do tipo FIFO, sem distinção entre os pacotes. Já o segundo modelo, prevê um esquema

de prioridade entre os pacotes. Os pacotes também são colocados numa fila com capacidade limitada, porém o escalonador os transmite conforme prioridades pré-definidas.

3.2 - Modelo de Dimensionamento FIFO (sem prioridade)

Considere uma rede de pacotes utilizada para transmissão de serviços integrados de dados e voz, conforme mostra o exemplo da Fig. 3.1. Considere também que a chegada de pacotes obedeça a uma distribuição de Poisson com taxa média λ (em pacotes por segundo), que o comprimento dos pacotes obedeça a uma distribuição exponencial negativa com média $E\{x\}$ (em bits) e que a capacidade do *buffer* está limitada a N pacotes.

Duas observações devem ser feitas para esta rede. Na primeira, em se tratando de uma rede que transporta pacotes de voz, o dimensionamento deve garantir que os pacotes não sofrerão atrasos superiores aos tolerados. Na segunda, como a capacidade do *buffer* é limitada, o dimensionamento dos enlaces deve garantir, também, que a perda de pacotes (voz e dados) seja tolerável.

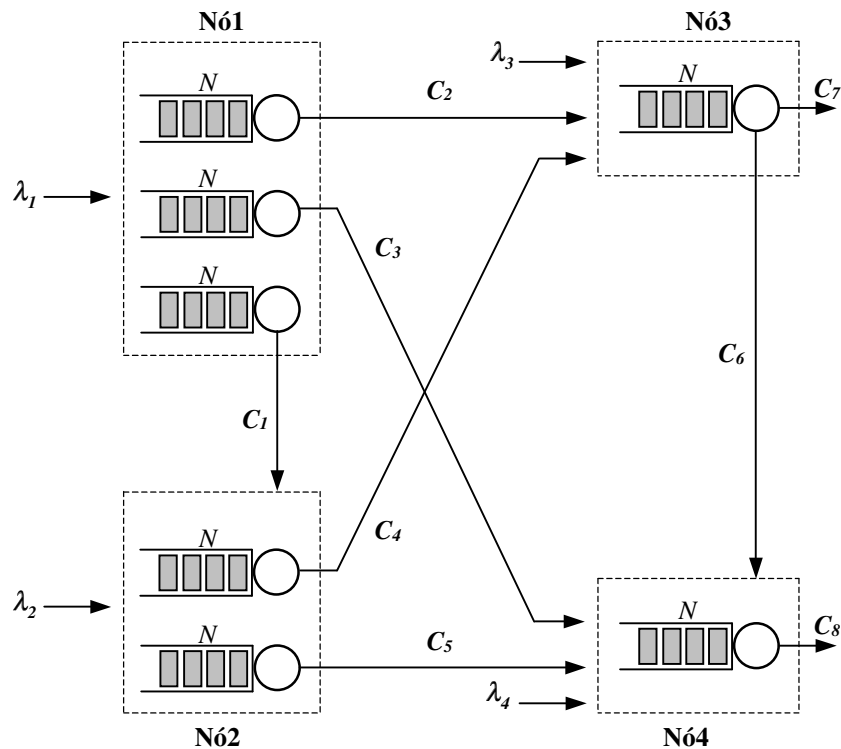


Figura 3.1 - Exemplo de uma rede IP com 4 Nós.

Levando em conta o exemplo da Fig. 3.1 e as suposições adotadas, o modelo de filas mais apropriado para o dimensionamento da capacidade de um enlace entre dois nós quaisquer, sujeito às restrições de uma dada taxa de perdas de pacotes (PB) e de um dado atraso de pacotes ($E\{T\}$), é o M/M/1/N.

3.2.1 - Modelo de Filas M/M/1/N [23]

O diagrama de transição de estados, representado na Fig. 3.2, é de um sistema M/M/1/N, onde N representa o número de pacotes que podem aguardar na fila incluindo o pacote atendido pelo único servidor do sistema. Se existirem N pacotes no sistema e ocorrer a chegada de um novo pacote, este último é bloqueado ou, em outras palavras, ocorreu a perda de um pacote.

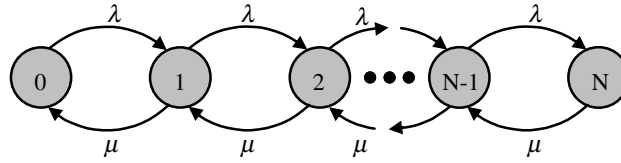


Figura 3.2 - Diagrama de transição de estados para o modelo M/M/1/N.

Para melhor expressar as equações utilizadas nesta seção, as seguintes definições são adotadas neste trabalho:

$$\rho = \lambda / \mu \quad (3.1)$$

$$E\{s\} = \frac{E\{x\}}{C} \quad (3.2)$$

$$E\{T\} = E\{T_q\} + E\{s\} \quad (3.3)$$

$$E\{L\} = E\{L_q\} + \rho \quad (3.4)$$

onde,

- ρ intensidade de tráfego;
- λ taxa média de chegada de pacotes;
- μ taxa média de partida de pacotes;

- $E\{s}$ tempo médio de transmissão de um pacote;
- $E\{x}$ comprimento médio dos pacotes;
- C capacidade de transmissão do enlace;
- $E\{T}$ atraso médio total dos pacotes no sistema;
- $E\{T_b}$ atraso médio dos pacotes no *buffer*;
- $E\{L}$ número médio total de pacotes no sistema;
- $E\{L_b}$ número médio de pacotes no *buffer*.

Escrevendo as equações de equilíbrio para cada estado da Fig. 3.2, resolvendo recursivamente para cada probabilidade de estado e impondo a condição de que a probabilidade de bloqueio (PB) é dada pela probabilidade de N pacotes no sistema, tem-se [23]:

$$PB = \frac{\rho^N (1 - \rho)}{(1 - \rho^{N+1})} \quad (3.5)$$

O número médio de pacotes no sistema é dado pela Eq. 3.6 [23]:

$$E\{L\} = \frac{\rho}{(1 - \rho)} - \frac{(N + 1)\rho^{N+1}}{(1 - \rho^{N+1})} \quad (3.6)$$

Utilizando a Lei de Little [23] pode-se obter facilmente o tempo médio total de atraso ($E\{T\}$) de um pacote no sistema (fila+servidor). A taxa média efetiva de chegada (λ_e) é dada por:

$$\lambda_e = \lambda(1 - PB) \quad (3.7)$$

Esta taxa de chegada representa somente os pacotes que efetivamente entram no sistema. Portanto a expressão do tempo médio de atraso total é dada por:

$$E\{T\} = \frac{E\{L\}}{\lambda_e} = \frac{E\{L\}}{\lambda(1 - PB)} \quad (3.8)$$

As Eqs. 3.5 e 3.8 são utilizadas para cálculos de probabilidade de bloqueio e atrasos de pacotes respectivamente, considerando que a intensidade de tráfego e o tamanho do *buffer* são conhecidos. Entretanto, para o dimensionamento de enlaces o problema é diferente, ou seja, dada a probabilidade de bloqueio e o atraso médio, quer-se calcular a capacidade do enlace que satisfaça as duas restrições definidas (PB e $E\{T\}$).

3.2.2 - Metodologia para o Cálculo da Capacidade do Enlace

A capacidade C (em bits/segundo) de um determinado enlace pode ser determinada através da Eq. 3.9.

$$C = E\{x\} \frac{\lambda}{\rho} \quad (3.9)$$

Considerando que $E\{x\}$ (tamanho médio dos pacotes, em bits) e λ da Eq. 3.9 são parâmetros conhecidos, para a obtenção de C deve-se determinar o valor da intensidade de tráfego (ρ). A partir das Eqs. 3.5, 3.6 e 3.8 pôde-se extrair duas expressões para ρ , uma em função de PB e outra em função de $E\{T\}$. O valor de ρ escolhido deve satisfazer os requisitos tanto de probabilidade de bloqueio quanto de atraso médio.

3.2.2.1 - Determinação de ρ a partir de PB

A Eq. 3.10, demonstrada a seguir, relaciona a taxa de perda de pacotes com a intensidade do tráfego de entrada de cada enlace em forma de polinômio na variável ρ e é decorrente da Eq. 3.5.

$$\begin{aligned} PB &= \frac{\rho^N (1 - \rho)}{(1 - \rho^{N+1})} \Rightarrow PB(1 - \rho^{N+1}) = \rho^N (1 - \rho) \Rightarrow \\ &(1 - PB)\rho^{N+1} - \rho^N = -PB \Rightarrow \\ \therefore f_{PB}(\rho) &= (1 - PB)\rho^{N+1} - \rho^N + PB \end{aligned} \quad (3.10)$$

Quando o polinômio da Eq. 3.10 é resolvido para $f_{PB}(\rho)=0$, obtém-se o valor da intensidade de tráfego que satisfaça PB e N especificados. No item 3.2.2.3 é apresentado um método para resolução desta equação.

3.2.2.2 - Determinação de ρ a partir de $E\{T\}$

Da mesma forma que foi feito para $f_{PB}(\rho)$, o objetivo aqui é encontrar uma expressão para determinação de ρ em função do atraso total $E\{T\}$. Juntando-se as Eqs. 3.5 e 3.8 tem-se como resultado a Eq. 3.11.

$$E\{T\} = E\{L\} \frac{(1 - \rho^{N+1})}{\lambda(1 - \rho^N)} \quad (3.11)$$

Substituindo a Eq. 3.6, que expressa o número médio de pacotes no sistema fila mais servidor, na Eq. 3.11 obtém-se o atraso médio de um pacote em um determinado nó apenas em função de ρ , λ e N , como mostra o desenvolvimento da Eq. 3.12 a seguir.

$$\begin{aligned} E\{T\} &= \left[\frac{\rho}{(1 - \rho)} - \frac{(N + 1)\rho^{N+1}}{(1 - \rho^{N+1})} \right] \frac{(1 - \rho^{N+1})}{\lambda(1 - \rho^N)} = \\ &= \left[\frac{\rho(1 - \rho^{N+1}) - (1 - \rho)(N + 1)\rho^{N+1}}{(1 - \rho)(1 - \rho^{N+1})} \right] \frac{(1 - \rho^{N+1})}{\lambda(1 - \rho^N)} = \\ &= \frac{\rho - \rho^{N+2} - N\rho^{N+1} + N\rho^{N+2} - \rho^{N+1} + \rho^{N+2}}{\lambda(1 - \rho^N - \rho + \rho^{N+1})} \Rightarrow \\ E\{T\} &= \frac{N\rho^{N+2} - (N + 1)\rho^{N+1} + \rho}{\lambda(\rho^{N+1} - \rho^N - \rho + 1)} \end{aligned} \quad (3.12)$$

E finalmente, através da Eq. 3.12, obtém-se a expressão de $f_T(\rho)$, mostrada na Eq. 3.13, que relaciona o atraso médio de pacotes com a intensidade do tráfego de entrada de cada enlace.

$$\begin{aligned}
 E\{T\} &= \frac{N\rho^{N+2} - (N+1)\rho^{N+1} + \rho}{\lambda(\rho^{N+1} - \rho^N - \rho + 1)} \Rightarrow \\
 E\{T\}\lambda(\rho^{N+1} - \rho^N - \rho + 1) &= N\rho^{N+2} - (N+1)\rho^{N+1} + \rho \Rightarrow \\
 E\{T\}\lambda\rho^{N+1} - E\{T\}\lambda\rho^N - E\{T\}\lambda\rho + E\{T\}\lambda &= N\rho^{N+2} - (N+1)\rho^{N+1} + \rho \Rightarrow \\
 N\rho^{N+2} - (E\{T\}\lambda + N + 1)\rho^{N+1} + E\{T\}\lambda\rho^N + (E\{T\}\lambda + 1)\rho &= E\{T\}\lambda \\
 \therefore f_T(\rho) = N\rho^{N+2} - (E\{T\}\lambda + N + 1)\rho^{N+1} + E\{T\}\lambda\rho^N + (E\{T\}\lambda + 1)\rho - E\{T\}\lambda & \quad (3.13)
 \end{aligned}$$

Conhecendo-se N e λ e resolvendo $f_T(\rho)=0$ é possível encontrar um valor para a intensidade de tráfego (ρ), que satisfaça o atraso médio especificado. A próxima seção apresenta um método para resolução desta equação.

3.2.2.3 - Cálculo de ρ a partir de $f_{PB}(\rho)$ e $f_T(\rho)$

A determinação da intensidade de tráfego (ρ) pode ser feita a partir de $f_{PB}(\rho)=0$ e $f_T(\rho)=0$. As resoluções algébricas destas equações são possíveis somente para valores pequenos de N . Portanto, é utilizado um método numérico para resolver esses problemas para valores arbitrários de N .

Ambas as expressões são polinomiais com coeficientes reais. Assim, pela regra de sinal de Descartes [24] pode-se garantir que $f_{PB}(\rho)$ e $f_T(\rho)$, independentemente do valor de N (que determina o grau de ambas) e dos demais parâmetros ($E\{T\}$, PB e λ), terão sempre duas e três raízes reais positivas, respectivamente. Dessas raízes, percebe-se que $\rho=1$ é sempre solução para $f_{PB}(\rho)$ (com multiplicidade um) e para $f_T(\rho)$ (com multiplicidade dois).

Dessa forma, como resta apenas uma raiz real positiva diferente de 1 em ambas as funções, o método da bissecção [24] é utilizado para determinação destas raízes. Este método consiste em aproximar-se da raiz através de múltiplas divisões de um intervalo inicial (ρ_1, ρ_2) , que contenha a raiz, até que se atinja a precisão desejada. A Fig. 3.3 mostra como este método pode ser implementado.

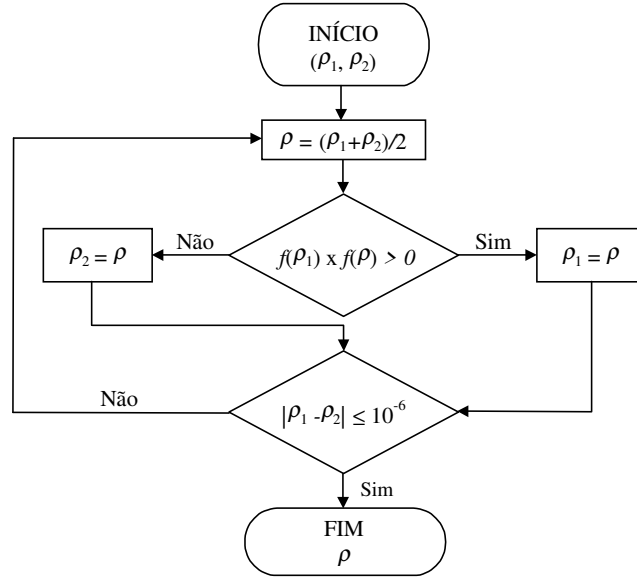


Figura 3.3 - Fluxograma do algoritmo para o método da bissecção.

Dividindo o polinômio da Eq. 3.10 por $(\rho - 1)$, uma vez que $\rho = 1$ é sempre raiz deste, elimina-se a raiz 1 do polinômio quociente $g_{PB}(\rho)$. Como resultado, obtém-se a Eq. 3.14.

$$g_{PB}(\rho) = (1 - PB)\rho^N - PB \sum_{k=0}^{N-1} \rho^k \quad (3.14)$$

Resolvendo $g_{PB}(\rho) = 0$, através do método da bissecção, encontra-se um valor aproximado para ρ que satisfaz a taxa de perda de pacotes solicitada. Para tanto, é necessário que se forneça um intervalo inicial (ρ_1, ρ_2) para que o método numérico possa iniciar o processo iterativo. O fluxograma da Fig. 3.4 mostra como este intervalo pode ser encontrado.

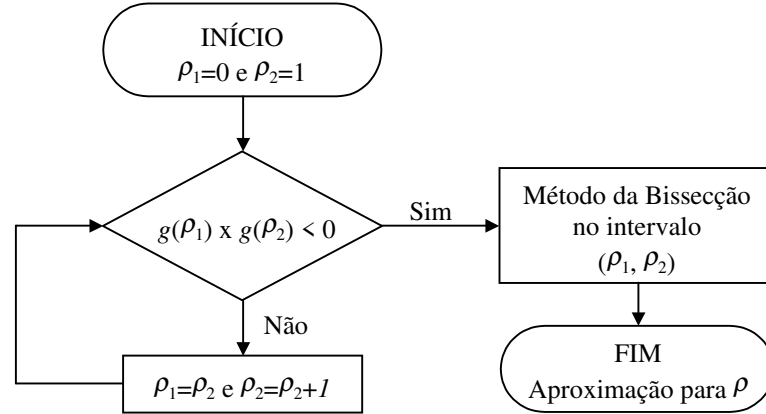


Figura 3.4 - Fluxograma do algoritmo que obtém um intervalo inicial para o método da bissecção.

Após a localização de um intervalo inicial, que contenha um zero de $g_{PB}(\rho)$, é aplicado o método da bissecção e, a partir da precisão desejada, obtém-se uma aproximação para ρ .

A determinação de ρ a partir da Eq. 3.13 segue os mesmos passos utilizados em $f_{PB}(\rho)$. Neste caso, o polinômio $f_T(\rho)$ é dividido por $(\rho-1)^2$, pois $f_T(\rho)$ contém sempre a raiz 1 com multiplicidade 2. Esta divisão resulta no polinômio mostrado na Eq. 3.15.

$$g_T(\rho) = N\rho^N + \sum_{k=1}^N \left\{ \rho^{N-k} [(N-k) - E\{T\}\lambda] \right\} \quad (3.15)$$

Resolvendo $g_T(\rho)=0$ através do método da bissecção, encontra-se o valor de ρ que satisfaz o atraso $E\{T\}$ solicitado. Neste caso, também é necessário que se forneça um intervalo inicial (ρ_1, ρ_2) , que contenha a raiz procurada, para que o método numérico possa iniciar o processo iterativo, conforme o procedimento mostrado na Fig. 3.4.

A capacidade C de um enlace qualquer pode ser então calculada levando-se em conta a intensidade de tráfego de entrada que satisfaça as duas restrições de QoS especificadas, o atraso e a taxa de perda dos pacotes. Portanto, para resolver a Eq. 3.9, é utilizada a menor intensidade encontrada da resolução das Eqs. 3.14 e 3.15.

Este modelo de dimensionamento pode ser utilizado em redes IP com integração de sinais de voz, conhecidas como VoIP (*Voice over IP*) - discutida na seção 2.6. Como essa rede é baseada na comutação de pacotes tradicional, os roteadores não estão preparados para garantir o tratamento de sinais de voz em tempo real. Desse modo, uma maneira de garantir a QoS é dimensionar o enlace para satisfazer os requisitos mais restritivos de atraso e de perda de pacotes.

3.3 - Modelo de Dimensionamento com Prioridades

A Fig. 3.5 mostra um exemplo de rede que suporta de forma integrada diferentes classes de serviço (ex.: voz, vídeo e dados). Considere para cada classe (k) que λ_k represente a taxa média de chegada de pacotes, $E\{x_k\}$ represente o comprimento médio dos pacotes e N represente a capacidade total do *buffer*.

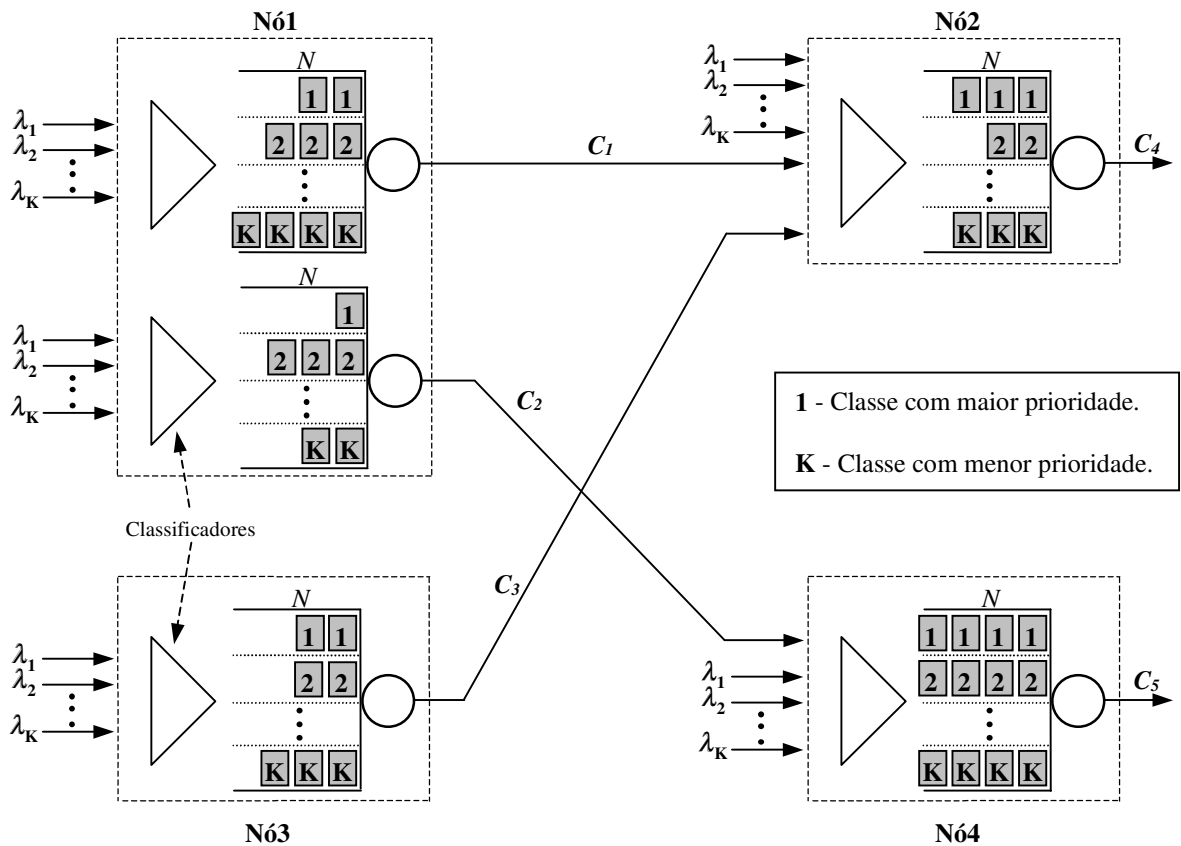


Figura 3.5 - Exemplo de rede com quatro nós implementando K classes de serviço.

Na rede mostrada na Fig 3.5 o *buffer* de entrada de cada enlace é particionado em K classes de serviço. Estas classes de serviço podem possuir requisitos distintos de QoS entre elas, por exemplo, um serviço de voz requer menor atraso do que um serviço de e-mail. Para resolver este problema, o modelo proposto nesta seção considera níveis de prioridades pré-definidos entre as classes. Desta forma, os pacotes com maior prioridade são sempre transmitidos primeiramente aos de prioridade inferiores, não importando a ordem de chegada dos pacotes.

3.3.1 - Modelo de Filas com Prioridades e sem Preempção [25]

Na Fig. 3.6 é mostrado um sistema de filas com prioridades, no qual os pacotes de diferentes classes de serviços são classificados e atendidos de acordo com uma prioridade pré-definida. Os pacotes marcados com menor prioridade são transmitidos através do enlace somente após todos os pacotes com prioridade superior terem sido. Todos os pacotes em atendimento, independentemente da prioridade, são transmitidos por inteiro, ou seja, sem preempção.

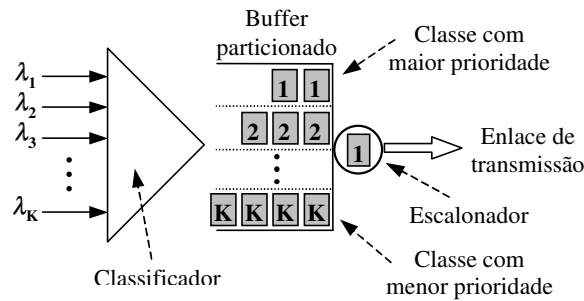


Figura 3.6 - Sistema de filas com prioridades e sem preempção.

Considere que a chegada de pacotes obedeça a uma distribuição poissoniana com média λ_k (em pacotes/segundo) e o comprimento dos pacotes obedeça a uma distribuição geral com média $E\{x_k\}$ (em bits) - sistema de filas M/G/1. Para que esta fila seja estável é necessário que a utilização total do enlace não ultrapasse a unidade, ou seja:

$$\rho = \rho_1 + \rho_2 + \rho_3 + \dots + \rho_K < 1 \quad (3.16)$$

onde, $\rho_k = \lambda_k \cdot E\{s_k\}$ e $E\{s_k\}$ representa o tempo médio de serviço gasto para a transmissão de um pacote da classe k através do enlace de saída.

Para se calcular o tempo médio de atraso em uma fila com prioridades, é estimado inicialmente o atraso residual $E\{R\}$ [25]. Este atraso representa o tempo médio gasto para o término do atendimento de um pacote que se encontra em serviço num determinado instante. Este pacote pode pertencer a qualquer classe de serviço. Desta forma, todas as classes de serviço estão sujeitas a este atraso. O tempo residual é dado por:

$$E\{R\} = \sum_{j=1}^K \frac{\lambda_j \cdot E\{s_j^2\}}{2} \quad (3.17)$$

onde, $E\{s_j^2\}$ representa a média quadrática do tempo de serviço, que por sua vez é representada pela soma dos quadrados do tempo médio de serviço e do desvio padrão deste, como é mostrado na Eq. 3.18.

$$E\{s_j^2\} = \sigma_j^2 + E^2\{s_j\} \quad (3.18)$$

A partir da Eq. 3.17, o tempo médio de atraso sofrido pelos pacotes da fila de maior prioridade é descrito através da Eq. 3.19 [25].

$$E\{T_{q_1}\} = E\{R\} + \lambda_1 E\{T_{q_1}\} \cdot E\{s_1\} \Rightarrow$$

$$E\{T_{q_1}\} = \frac{E\{R\}}{(1 - \rho_1)} \quad (3.19)$$

O tempo médio esperado pelos pacotes na segunda fila de prioridades é dado pela Eq. 3.20. O termo $E\{M_1\} = \lambda_1 \cdot E\{T_{q_1}\}$ é a quantidade de pacotes da classe 1 (maior prioridade) que chegam ao sistema enquanto o escalonador está servindo a fila da classe 2 [25].

$$E\{T_{q_2}\} = E\{R\} + \lambda_1 \cdot E\{T_{q_1}\} \cdot E\{s_1\} + \lambda_2 \cdot E\{T_{q_2}\} \cdot E\{s_2\} + E\{M_1\} \cdot E\{s_1\} \Rightarrow$$

$$E\{T_{q_2}\} = \frac{E\{R\}}{(1 - \rho_1) \cdot (1 - \rho_1 - \rho_2)} \quad (3.20)$$

Os atrasos na fila referentes aos pacotes das outras classes de prioridade podem ser obtidos seguindo-se a mesma lógica empregada nas duas primeiras classes. Dessa forma, é obtida a Eq. 3.21 que representa o atraso médio de fila sofrido pelos pacotes para cada classe de prioridade [25].

$$E\{T_{q_k}\} = \frac{E\{R\}}{\left(1 - \sum_{j=1}^{k-1} \rho_j\right) \cdot \left(1 - \sum_{j=1}^k \rho_j\right)} \quad (3.21)$$

Considere agora que o tamanho médio dos pacotes de qualquer classe de prioridade obedeça a uma distribuição exponencial negativa com média $E\{x_k\}$. Então, o tempo médio de serviço $E\{s_k\}$ de cada pacote também obedece a essa distribuição e $E\{s_k^2\} = 2E^2\{s_k\}$. Dessa forma, com o auxílio das Eqs. 3.17 e 3.18, o atraso médio sofrido por um pacote é dado pela Eq. 3.22.

$$E\{T_k\} = \frac{\sum_{j=1}^K \lambda_j \cdot E^2\{s_j\}}{\left(1 - \sum_{j=1}^{k-1} \rho_j\right) \cdot \left(1 - \sum_{j=1}^k \rho_j\right)} + E\{s_k\} \quad (3.22)$$

A Eq. 3.22 refere-se ao atraso médio total sofrido pelos pacotes ao atravessarem todo o sistema - fila mais servidor. Em outras palavras, é o tempo gasto na fila mais o tempo gasto no atendimento ($E\{T_k\} = E\{T_{qk}\} + E\{s_k\}$).

3.3.2 - Metodologia para o Cálculo da Capacidade do Enlace

O método de dimensionamento apresentado nesta seção utiliza, como parâmetros de restrição de QoS, o atraso e a taxa de perda dos pacotes especificados para cada classe de serviço/prioridade. O cálculo da capacidade do enlace é feito levando-se em conta o atraso médio de cada classe de serviço através da Eq. 3.22. A garantia de que a taxa de perda de pacotes não ultrapasse a especificada é obtida através de um modelo equivalente para dimensionamento da capacidade do *buffer*.

3.3.2.1 - Modelo para o Cálculo da Capacidade do Enlace

A Eq. 3.23 é uma forma de representar a Eq. 3.22 em termos da capacidade do enlace (C). Para tanto, foi definida a expressão da utilização do enlace para cada classe de serviço, ou seja,

$$\rho_k = \frac{\lambda_k \cdot E\{x_k\}}{C}$$

Desta forma, a expressão para o atraso médio total pode ser escrita como:

$$E\{T_k\} = \frac{\frac{1}{C^2} \sum_{j=1}^K \lambda_j \cdot E^2\{x_j\}}{\left(1 - \frac{1}{C} \sum_{j=1}^{k-1} \lambda_j \cdot E\{x_j\}\right) \cdot \left(1 - \frac{1}{C} \sum_{j=1}^k \lambda_j \cdot E\{x_j\}\right)} + \frac{E\{x_k\}}{C} \Rightarrow$$

$$E\{T_k\} = \frac{\sum_{j=1}^K \lambda_j \cdot E^2\{x_j\}}{\left(C - \sum_{j=1}^{k-1} \lambda_j \cdot E\{x_j\}\right) \cdot \left(C - \sum_{j=1}^k \lambda_j \cdot E\{x_j\}\right)} + \frac{E\{x_k\}}{C} \quad (3.23)$$

A fim de facilitar a manipulação da Eq. 3.23, lembrando que o objetivo é encontrar uma capacidade C que atenda a restrição de atraso $E\{T_k\}$, são definidos alguns parâmetros, conforme Eqs. 3.24, 3.25 e 3.26.

$$A = \sum_{j=1}^K \lambda_j \cdot E^2\{x_j\} \quad (3.24)$$

$$B_k = \sum_{j=1}^{k-1} \lambda_j \cdot E\{x_j\}, \quad B_1 = 0 \quad (3.25)$$

$$B'_k = \sum_{j=1}^k \lambda_j \cdot E\{x_j\} \quad (3.26)$$

Substituindo estes parâmetros na Eq. 3.23 e reescrevendo a no formato de polinômio na variável C , obtém-se a Eq. 3.27.

$$E\{T_k\} = \frac{A}{(C - B_k)(C - B'_k)} + \frac{E\{x_k\}}{C} \Rightarrow$$

$$E\{T_k\} = \frac{AC + E\{x_k\}(C - B_k)(C - B'_k)}{C(C - B_k)(C - B'_k)} \Rightarrow$$

$$E\{T_k\} = \frac{AC + E\{x_k\}C^2 - E\{x_k\}CB'_k - E\{x_k\}B_kC + E\{x_k\}B_kB'_k}{C^3 - C^2B'_k - C^2B_k + CB_kB'_k} \Rightarrow$$

$$\therefore E\{T_k\}C^3 - [E\{T_k\}(B_k + B'_k) + E\{x_k\}]C^2 + [E\{T_k\}B_kB'_k + E\{x_k\}(B_k + B'_k) - A]C - E\{x_k\}B_kB'_k = 0 \quad (3.27)$$

Uma análise detalhada do polinômio representado pela Eq. 3.27 sugere que este terá, no mínimo, uma raiz real positiva. Visto que todos os coeficientes deste polinômio são sempre números reais, conforme a regra do sinal de Descartes [24]. Outro aspecto importante é que o polinômio em questão é de

grau 3 - diferente do que acontece com as Eqs. 3.14 e 3.15 do modelo FIFO, onde o grau é N - assim é possível o cálculo das três raízes a partir de fórmulas conhecidas.

Ainda com base na Eq. 3.27, para cada classe de serviço/prioridade ($k = 1, 2, 3, \dots$) existe um polinômio de grau 3 a ser resolvido, onde estes terão parâmetros específicos de cada classe (atraso médio, taxa de chegada e comprimento médio dos pacotes). Abaixo segue um exemplo para um sistema, como o da Fig. 3.6, com 3 classes de serviço/prioridade.

- Primeira Classe de Serviço ($k = 1$)

$$E\{T_1\}C^3 - [E\{T_1\}(B_1 + B'_1) + E\{x_1\}]C^2 + [E\{T_1\}B_1B'_1 + E\{x_1\}(B_1 + B'_1) - A]C - E\{x_1\}B_1B'_1 = 0$$

- Segunda Classe de Serviço ($k = 2$)

$$E\{T_2\}C^3 - [E\{T_2\}(B_2 + B'_2) + E\{x_2\}]C^2 + [E\{T_2\}B_2B'_2 + E\{x_2\}(B_2 + B'_2) - A]C - E\{x_2\}B_2B'_2 = 0$$

- Terceira Classe de Serviço ($k = 3$)

$$E\{T_3\}C^3 - [E\{T_3\}(B_3 + B'_3) + E\{x_3\}]C^2 + [E\{T_3\}B_3B'_3 + E\{x_3\}(B_3 + B'_3) - A]C - E\{x_3\}B_3B'_3 = 0$$

Ao término da resolução destas três equações são encontradas nove raízes, três para cada equação, candidatas à capacidade do enlace. De cada classe de serviço escolhe-se a raiz real positiva com maior valor entre as três. Dessa forma, restam apenas três raízes válidas, candidatas à capacidade do enlace, uma de cada equação. A raiz que atende ao mesmo tempo os requisitos de atraso das três classes de serviço é escolhida como sendo a maior entre as três possíveis candidatas.

Um método para se resolver a Eq. 3.27 é descrito com detalhes no Apêndice I. Este método de resolução das raízes de equações do terceiro grau, também conhecido como método de Tartaglia ou Cardano, fornece as três raízes de um polinômio de maneira rápida e direta [26].

3.3.2.2 - Modelo Equivalente para o Dimensionamento do Buffer

O modelo de dimensionamento proposto utiliza como parâmetros de restrição de QoS o atraso e a taxa de perda dos pacotes, definidos para cada classe de serviço/prioridade. No item anterior foi

proposto um método de cálculo da capacidade do enlace, levando-se em conta apenas o atraso médio de cada classe de serviço, sem um limite para a capacidade do *buffer*.

A Fig. 3.7 ilustra um sistema de enfileiramento com três classes de serviço onde os pacotes de qualquer uma das classes são armazenados num *buffer* de capacidade limitada (N). Neste sistema de enfileiramento, independentemente da classe de prioridade, se não existir espaço suficiente no *buffer*, os pacotes que chegarem serão descartados conforme uma taxa de bloqueio (PB_k) pré-definida.

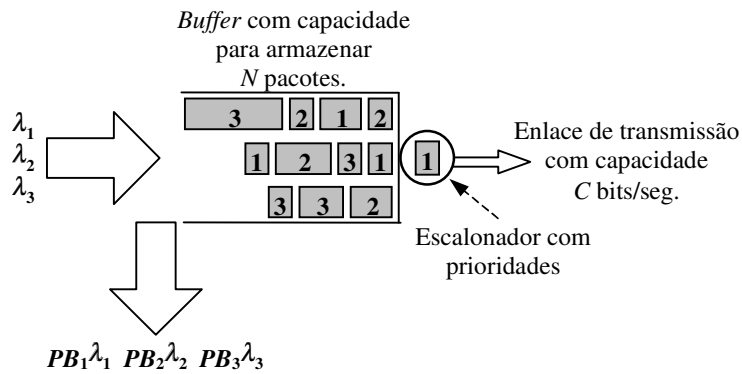


Figura 3.7 - Modelo equivalente para o sistema de filas com prioridade e *buffer* limitado.

A seguir é apresentado um modelo equivalente para o dimensionamento da capacidade do *buffer* que utiliza como ponto de partida a capacidade do enlace (C), obtida pelo modelo apresentado no item anterior, e a taxa de perda de pacotes especificada para cada uma das classes de serviço/prioridade.

A Eq. 3.16, que expressa a utilização média do enlace, é reescrita em termos da capacidade do enlace (em bits/seg), da taxa de chegada dos pacotes (em pacotes/seg) e do tamanho médio dos pacotes (em bits), conforme mostrado na Eq. 3.28.

$$\rho = \sum_{j=1}^K \frac{\lambda_j \cdot E\{x_j\}}{C} < 1 \quad (3.28)$$

A partir da utilização média do enlace, dada pela Eq. 3.28, é possível a realização do cálculo da capacidade equivalente do *buffer*, a fim de que sejam atendidos os requisitos de perda de pacotes especificados para cada classe de serviço. Para isso, o processo de perda de pacotes do sistema da Fig. 3.7

é considerado equivalente ao de um sistema M/M/1/N, ou seja, estando o *buffer* cheio, a chegada de novos pacotes resultará em perdas, independentemente da prioridade destes pacotes. Portanto, a partir da Eq.3.5, obtém-se a Eq.3.29 que é utilizada para o cálculo da capacidade equivalente do *buffer* [27].

$$N = -\frac{\ln\left(\frac{PB_{\min}}{1 - \rho + \rho \cdot PB_{\min}}\right)}{\ln(\rho)} \quad (3.29)$$

Na Eq.3.29, $PB_{\min}$ refere-se à taxa de perda de pacotes mais restritiva entre todas as classes de serviço existentes no sistema e ρ é dado pela Eq. 3.28. Por exemplo, na Fig. 3.7, $PB_{\min}$ é escolhido como sendo o menor entre PB_1 , PB_2 e PB_3 . Dessa forma, a capacidade encontrada para o *buffer* atenderá aos requisitos de perda de pacotes de todas as classes de serviços.

O critério proposto para o dimensionamento pode ser dividido em duas etapas. Na primeira etapa é feito o cálculo da capacidade do enlace, considerando o atraso médio dos pacotes para cada classe e o *buffer* com capacidade infinita. Já a segunda etapa consiste em calcular uma capacidade equivalente para o *buffer*, satisfazendo as perdas de pacotes definidas para cada classe. O resultado final do dimensionamento do enlace é obtido pela união destas duas etapas, ou seja, obtém-se uma capacidade para o enlace (C) e uma capacidade para o *buffer* (N) que satisfazem os requisitos de QoS, definidos para todas as classes de prioridades.

3.4 - Conclusões

Neste capítulo foram propostos dois modelos para o dimensionamento da capacidade dos enlaces em redes de comutação de pacotes, ambos considerando apenas dois dos principais parâmetros de QoS - o atraso médio e a taxa média de perda de pacotes.

O primeiro modelo, denominado modelo FIFO, não faz nenhuma discriminação entre os pacotes. Desta forma, para o dimensionamento, considera-se os parâmetros de QoS que pertencerem a tipos de serviço mais restritivos (ex.: atraso médio para pacotes de voz e taxa de perda para pacotes de dados). Com isso, pode ocorrer que a capacidade encontrada para o enlace seja superestimada e, conseqüentemente, este enlace estará sendo mal utilizado.

No outro modelo, os pacotes são separados por classes de serviços e o escalonador transmite os pacotes de acordo com uma prioridade pré-definida. O dimensionamento aqui é dividido em duas etapas. Na primeira etapa, o dimensionamento da capacidade dos enlaces é feito levando-se em conta o requisito de atraso médio de pacotes de cada classe, considerando a capacidade do *buffer* infinita. Na segunda etapa, para garantir a taxa de perda de pacotes de todas as classes, é proposto um modelo de cálculo da capacidade equivalente do *buffer*. Neste caso, como são levados em conta os parâmetros de QoS de todas as classes de serviço, e não somente o mais restritivo, é possível se obter uma melhor utilização para o enlace.

Capítulo 4

Software de Dimensionamento de Enlaces

4.1 - Introdução

No capítulo anterior foram propostos dois métodos para o dimensionamento da capacidade de enlaces, ambos podem ser considerados integrantes de um processo de projeto de redes. Nestes processos, além da capacidade dos enlaces, outros fatores devem ser considerados, por exemplo, a agilidade e a confiabilidade com que um projeto é realizado. Neste sentido, o desenvolvimento de ferramentas computacionais (*softwares*), que automatizem o projeto de redes, torna-se indispensável.

Ferramentas mais completas, de auxílio no projeto de redes, podem incorporar funcionalidades que prevêm estimação do tráfego, dimensionamento de enlaces, cálculo de custos, banco de dados com informações dos equipamentos, etc. Operadores de redes podem ser beneficiados com a utilização destas ferramentas, pois o dinamismo das redes atuais exige uma rápida capacidade de acesso e atualização das informações relacionadas ao projeto.

Neste capítulo, primeiramente, é proposto um software como ferramenta de auxílio na tarefa do projeto de redes. Esta ferramenta está destinada exclusivamente a automatizar o cálculo da capacidade dos enlaces de uma rede, bastando que se forneça a topologia e alguns parâmetros de configuração dessa rede. A essência do software consiste na utilização dos modelos de dimensionamento propostos no capítulo anterior. Na seqüência, são apresentados alguns exemplos de dimensionamento realizados pelo software e, através de simulações de eventos discretos, é feita a validação dos resultados obtidos. Ao

final, são mostradas algumas considerações a respeito da utilização prática dos resultados obtidos nos exemplos de dimensionamento.

4.2 - Descrição do Software Desenvolvido

“EasyLink” é o nome do software que foi desenvolvido como ferramenta de dimensionamento de enlaces. O EasyLink foi desenvolvido em Delphi - ferramenta de desenvolvimento de ambientes visuais para Windows - que possui recursos de programação orientada a objetos, além de prover suporte para acesso e manipulação de arquivos de bancos de dados. A Fig. 4.1 mostra a tela principal do EasyLink.

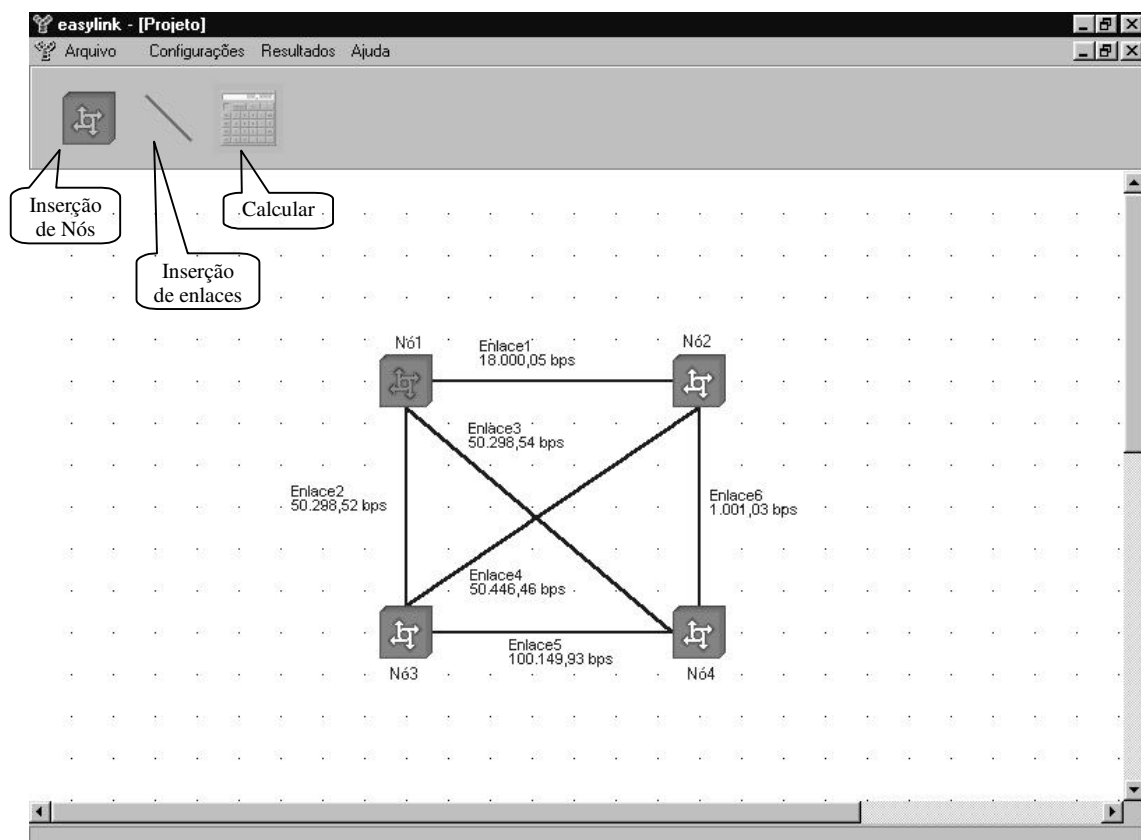


Figura 4.1 - Tela principal do EasyLink.

Na Fig.4.1 é ilustrado, na área de trabalho do EasyLink, um exemplo de rede com quatro nós e seis enlaces, juntamente com a capacidade de cada enlace. Para que seja possível a realização de exemplos

como o da Fig 4.1, um conjunto de ferramentas está incorporado ao software. As ferramentas, também mostradas na Fig.4.1, e suas funcionalidades são:

- **Inserção de Nós:** Esta ferramenta é utilizada para informar ao software a topologia da rede, a partir da criação e inserção de nós a esta.
- **Inserção de Enlaces:** Tem a função de complementar a informação de topologia da rede, iniciada pela ferramenta anterior, através da criação de enlaces entre dois nós. Estes enlaces são unidirecionais e devem ser criados informando-se os nós de origem e de destino do tráfego.
- **Calcular:** Após todo o processo de criação e configuração da rede, utiliza-se esta ferramenta para ativar o processo de cálculo da capacidade dos enlaces.

O processo de representação topológica da rede é bastante simples. Este requer apenas que o usuário, com a utilização das ferramentas de inserção de nós e de enlaces, “desenhe” na área de trabalho do software a topologia física da rede. A partir daí, o usuário pode iniciar o processo de configuração da rede. O processo de configuração é realizado nó-a-nó, conforme Fig. 4.2.

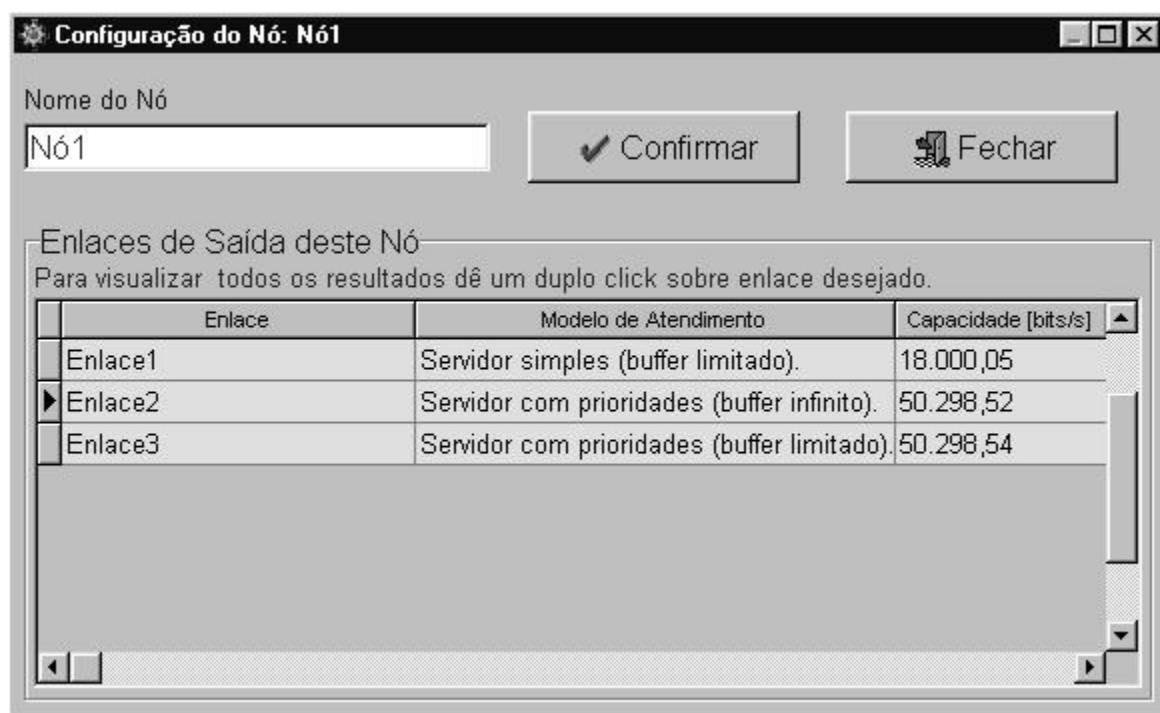


Figura 4.2 - Tela de acesso à configuração de cada nó da rede.

Na Fig.4.2 são mostrados também os enlaces que estão relacionados ao nó em questão. Cada enlace é relacionado ao nó de origem do tráfego, isto é feito automaticamente no processo de criação destes enlaces. É a partir desta tela que se obtém acesso às configurações dos enlaces de saída de cada nó.

O processo de configuração dos enlaces inicia-se com a escolha do modelo de atendimento dos pacotes, utilizado pelo servidor de cada enlace. A versão atual do EasyLink (versão 1.00) contempla apenas três destes modelos, que são implementações dos modelos de dimensionamento propostos no capítulo anterior - o modelo servidor simples (com *buffer* limitado), representa o modelo FIFO (item 3.2) e os modelos servidor com prioridades (com *buffer* infinito) e servidor com prioridades (com *buffer* limitado), representam o modelo de prioridades (item 3.3).

Na Fig. 4.3 é mostrada a tela de configuração de um enlace, o qual utiliza o modelo de atendimento servidor simples (*buffer* limitado).

A imagem mostra uma janela de software intitulada "Configurações dos Enlaces".

- Nome do Enlace:** Um campo de texto contendo "Enlace1".
- Modelo de Atendimento dos pacotes:** Um menu suspenso com a opção selecionada "Servidor simples (buffer limitado)".
- Botões:** "Confirmar" (com uma seta verde) e "Eechar" (com um ícone de porta).
- Parâmetros de Entrada e Resultados:**
 - Taxa de chegada dos pacotes:** Um campo de entrada com o valor "10000" e a unidade "pacotes/s".
 - Tamanho médio dos pacotes:** Um campo de entrada com o valor "2" e a unidade "bits".
 - Atraso médio total:**
 - Dado:** Um campo com o valor "0,1" e a unidade "s".
 - Calculado:** Um campo com o valor "1,011E-2" e a unidade "s".
 - Taxa de perda de pacotes:**
 - Dado:** Um campo com o valor "0,1".
 - Calculado:** Um campo com o valor "1E-1".
 - Capacidade do buffer:** Um campo de entrada com o valor "100" e a unidade "pacotes".

Figura 4.3 - Tela de configuração de um enlace no modo servidor simples (*buffer* limitado).

Se o modelo de atendimento escolhido for o de servidor com prioridades (com *buffer* limitado) ou de servidor com prioridades (com *buffer* infinito), a tela que aparecerá será semelhante à mostrada na Fig. 4.4, com a exceção de que no último modelo não aparecerão as opções de taxa de perda de pacotes e de capacidade do *buffer*.

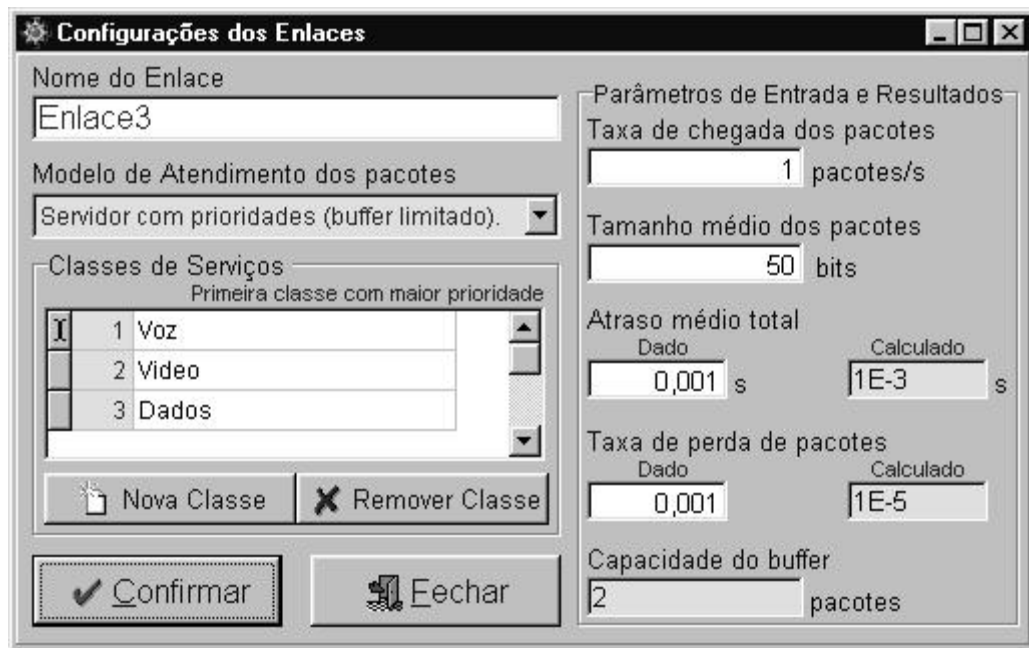
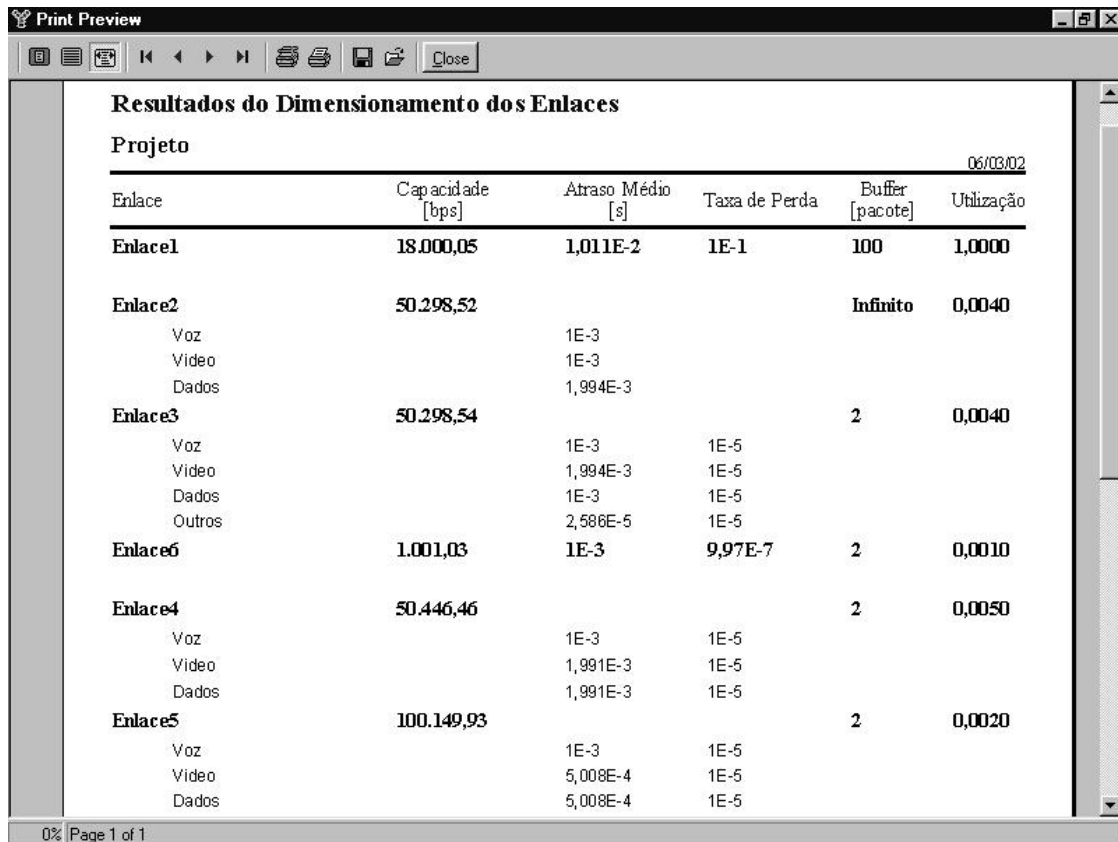


Figura 4.4 - Tela de configuração de um enlace no modo servidor com prioridades.

Ainda com relação a Fig. 4.4, o modelo escolhido prevê no mínimo duas classes de serviço/prioridade por enlace. Neste caso, os parâmetros de dimensionamento são especificados por classe de prioridade e não mais por enlace, como no modelo de servidor simples. Portanto, para cada classe de serviço é especificado um conjunto de parâmetros, tais como, taxa de chegada dos pacotes, tamanho médio dos pacotes, atraso médio dos pacotes, etc.

O processo de configuração dos enlaces é finalizado após todos os parâmetros, necessários ao dimensionamento, terem sido informados ao EasyLink. Estes parâmetros são provenientes do modelo de dimensionamento utilizado e estão disponíveis nas telas de configurações dos enlaces, mostradas nas Figs. 4.3 e 4.4, conforme o modelo de atendimento especificado para cada enlace.

Feita toda a configuração da rede, o software está pronto para realizar a tarefa de dimensionamento da capacidade dos enlaces. Esta tarefa se inicia com o acionamento da ferramenta especificada para sua automatização (botão calcular). Os resultados do dimensionamento são apresentados na própria área de trabalho do software, ao lado de cada enlace, e através de um relatório que reúne os resultados do dimensionamento de todos os enlaces da rede. Um exemplo deste relatório é mostrado na Fig. 4.5.



Print Preview

Resultados do Dimensionamento dos Enlaces

Projeto 06/03/02

Enlace	Capacidade [bps]	Atraso Médio [s]	Taxa de Perda	Buffer [pacote]	Utilização
Enlace1	18.000,05	1,011E-2	1E-1	100	1,0000
Enlace2	50.298,52			Infinito	0,0040
Voz		1E-3			
Video		1E-3			
Dados		1,994E-3			
Enlace3	50.298,54			2	0,0040
Voz		1E-3	1E-5		
Video		1,994E-3	1E-5		
Dados		1E-3	1E-5		
Outros		2,586E-5	1E-5		
Enlace6	1.001,03	1E-3	9,97E-7	2	0,0010
Enlace4	50.446,46			2	0,0050
Voz		1E-3	1E-5		
Video		1,991E-3	1E-5		
Dados		1,991E-3	1E-5		
Enlace5	100.149,93			2	0,0020
Voz		1E-3	1E-5		
Video		5,008E-4	1E-5		
Dados		5,008E-4	1E-5		

0% Page 1 of 1

Figura 4.5 - Exemplo de um relatório de resultados gerado pelo EasyLink.

Todas as informações, relacionadas ao projeto, são armazenadas automaticamente num conjunto de banco de dados incorporado ao software. Dessa forma, a qualquer momento, é possível recuperar todas essas informações, bem como alterá-las, bastando apenas que o projeto especificado seja reaberto a partir do EasyLink.

4.3 - Exemplos de Dimensionamento

A fim de ilustrar exemplos de utilização do software desenvolvido, foi feito o dimensionamento da capacidade de enlaces da rede da Fig. 4.1. Os exemplos são baseados em dois enlaces - Enlace 1 e Enlace 2. No Enlace 1 foi considerado o modo servidor simples (*buffer* limitado) e no Enlace 2 o modo servidor com prioridades (*buffer* limitado). O dimensionamento destes dois enlaces foi feito seguindo-se o procedimento de utilização do EasyLink mostrado no item anterior. Por fim, os resultados obtidos no dimensionamento foram simulados, a partir de simulações de eventos discretos, a fim de se validar as metodologias propostas no capítulo anterior.

Nas etapas de dimensionamento e de simulação dos enlaces 1 e 2, mostradas nos itens seguintes, são especificados alguns parâmetros de configuração. As simbologias adotadas para representar estes parâmetros estão definidas a seguir:

- λ taxa média de chegada de pacotes (em pacotes/s);
- $E\{x\}$ tamanho médio dos pacotes (em bits);
- $E\{T\}$ atraso médio total dos pacotes (em segundos);
- PB taxa de perda de pacotes;
- N capacidade do *buffer* (em pacotes);
- C capacidade do enlace (em bits/s).

4.3.1 - Dimensionamento do Enlace 1

Através da tela de configuração do Enlace 1, semelhante à mostrada na Fig. 4.3, foi informado ao software os parâmetros de dimensionamento utilizados para o cálculo da capacidade deste enlace. Na Tab. 4.1 são mostrados os parâmetros utilizados e os resultados obtidos no dimensionamento do Enlace 1, para quinze exemplos distintos, considerando o modelo de dimensionamento FIFO com *buffer* limitado.

Exemplos	Parâmetros de dimensionamento					Resultados do EasyLink		
	λ	$E\{x\}$	$E\{T\}$	PB	N	C	$E\{T\}$	PB
1	100	500	0,150	1×10^{-3}	10	92284	0,012	$1,0 \times 10^{-3}$
2	200	500	0,150	1×10^{-3}	10	184568	0,006	$1,0 \times 10^{-3}$
3	300	500	0,150	1×10^{-3}	10	276852	0,004	$1,0 \times 10^{-3}$
4	100	600	0,150	1×10^{-3}	10	110741	0,012	$1,0 \times 10^{-3}$
5	100	700	0,150	1×10^{-3}	10	129198	0,012	$1,0 \times 10^{-3}$
6	100	800	0,150	1×10^{-3}	10	147655	0,012	$1,0 \times 10^{-3}$
7	100	500	0,001	1×10^{-3}	10	549998	0,001	$3,5 \times 10^{-11}$
8	100	500	0,005	1×10^{-3}	10	149989	0,005	$1,1 \times 10^{-5}$
9	100	500	0,010	1×10^{-3}	10	99750	0,010	$5,0 \times 10^{-4}$
10	100	500	0,150	1×10^{-4}	10	118929	0,007	$1,0 \times 10^{-4}$
11	100	500	0,150	1×10^{-5}	10	151928	0,005	$1,0 \times 10^{-5}$
12	100	500	0,150	1×10^{-6}	10	193180	0,004	$1,0 \times 10^{-6}$
13	100	500	0,150	1×10^{-3}	50	54661	0,102	$1,0 \times 10^{-3}$
14	100	500	0,150	1×10^{-3}	100	53299	0,150	$1,0 \times 10^{-4}$
15	100	500	0,150	1×10^{-3}	1000	53333	0,150	$1,0 \times 10^{-30}$

Tabela 4.1 - Exemplos de dimensionamento do Enlace 1 feitos através do EasyLink.

Os campos em destaque na Tab. 4.1 servem para indicar qual parâmetro está sobre observação, ou seja, se a alteração de cada um dos parâmetros, isoladamente, influencia no cálculo da capacidade do enlace, a fim de que esta capacidade atenda os requisitos de atraso e de perda de pacotes.

4.3.2 - Dimensionamento do Enlace 2

O cálculo da capacidade do Enlace 2 tem início com a especificação dos parâmetros de dimensionamento, conforme uma tela de configuração semelhante à mostrada na Fig. 4.4. Os parâmetros de dimensionamento, neste exemplo, estão distribuídos em três classes de serviço - voz (maior prioridade), vídeo (média prioridade) e dados (menor prioridade) - uma vez que o modelo de dimensionamento utilizado é o que considera classes com prioridades. Na Tab. 4.2 são mostrados os parâmetros utilizados, para cada classe, e os resultados obtidos no dimensionamento do Enlace 2, para 5 exemplos distintos.

Exemplos		Parâmetros de dimensionamento				Resultados do EasyLink			
		λ	$E\{x\}$	$E\{T\}$	PB	C	N	$E\{T\}$	PB
1	voz	20	500	0,150	1×10^{-3}	51139	37	0,020	$1,0 \times 10^{-3}$
	vídeo	30	500	0,150	1×10^{-3}			0,030	$1,0 \times 10^{-3}$
	dados	40	500	0,150	1×10^{-3}			0,150	$1,0 \times 10^{-3}$
2	voz	20	50	0,150	1×10^{-3}	16307	23	0,024	$1,0 \times 10^{-3}$
	vídeo	20	100	0,150	1×10^{-3}			0,032	$1,0 \times 10^{-3}$
	dados	20	500	0,150	1×10^{-3}			0,150	$1,0 \times 10^{-3}$
3	voz	20	500	0,100	1×10^{-3}	33809	40	0,033	$1,0 \times 10^{-3}$
	vídeo	20	500	0,200	1×10^{-3}			0,060	$1,0 \times 10^{-3}$
	dados	20	500	0,300	1×10^{-3}			0,300	$1,0 \times 10^{-3}$
4	voz	20	500	0,150	1×10^{-4}	36620	61	0,030	$1,0 \times 10^{-6}$
	vídeo	20	500	0,150	1×10^{-5}			0,048	$1,0 \times 10^{-6}$
	dados	20	500	0,150	1×10^{-6}			0,150	$1,0 \times 10^{-6}$
5	voz	30	50	0,100	1×10^{-3}	9529	81	0,042	$1,0 \times 10^{-5}$
	vídeo	20	100	0,200	1×10^{-4}			0,068	$1,0 \times 10^{-5}$
	dados	10	500	0,500	1×10^{-5}			0,500	$1,0 \times 10^{-5}$

Tabela 4.2 - Exemplos de dimensionamento do Enlace 2 feitos através do EasyLink.

Com a finalidade de observar o efeito provocado por todos os parâmetros, sobre a capacidade do enlace, considerou-se a alteração de apenas um destes parâmetros, entre um exemplo e outro, conforme os campos destacados na Tab. 4.2.

4.3.3 - Simulações com os Resultados do Dimensionamento

Nesta seção são mostradas as simulações feitas com os resultados dos dimensionamentos do Enlace 1 e do Enlace 2, através de um software de simulação dinâmica de eventos discretos - O ARENA (Versão 3.5 - Demo). Softwares de simulações são poderosas ferramentas para a realização de análises de desempenho de redes. Dessa forma, a partir da capacidade dos enlaces e da capacidade do *buffer* (no caso do Enlace 2), Tabs. 4.1 e 4.2, é possível verificar se os requisitos de atraso e de perda de pacotes estão compatíveis aos especificados no dimensionamento.

4.3.3.1 - Simulações com os resultados do Enlace 1

Os resultados das simulações, referentes aos exemplos de dimensionamento da Tab.4.1, estão mostrados na Tab. 4.3. Observe que a capacidade do enlace, calculada através do modelo de dimensionamento FIFO, proposto no capítulo anterior, é utilizada como parâmetro de configuração na simulação.

Exemplos	Parâmetros utilizados na simulação				Resultados da simulação	
	λ	$E\{x\}$	C	N	$E\{T\}$	PB
1	100	500	92284	10	0,012	$6,0 \times 10^{-4}$
2	200	500	184568	10	0,006	$5,4 \times 10^{-4}$
3	300	500	276852	10	0,004	$5,8 \times 10^{-4}$
4	100	600	110741	10	0,012	$5,8 \times 10^{-4}$
5	100	700	129198	10	0,012	$5,6 \times 10^{-4}$
6	100	800	147655	10	0,012	$5,6 \times 10^{-4}$
7	100	500	549998	10	0,001	$< 10^{-6}$
8	100	500	149989	10	0,005	$< 10^{-6}$
9	100	500	99750	10	0,010	$1,8 \times 10^{-4}$
10	100	500	118929	10	0,007	$2,6 \times 10^{-5}$
11	100	500	151928	10	0,005	$5,0 \times 10^{-6}$
12	100	500	193180	10	0,004	$1,0 \times 10^{-6}$
13	100	500	54661	50	0,091	$4,6 \times 10^{-4}$
14	100	500	53299	100	0,131	$3,0 \times 10^{-5}$
15	100	500	53333	1000	0,145	$1,0 \times 10^{-3}$

Tabela 4.3 - Resultados obtidos através das simulações do Enlace 1.

A partir dos resultados da simulação, mostrados na Tab. 4.3, é possível verificar se a capacidade do enlace, obtida através do EasyLink, atende aos requisitos de atraso e de perda de pacotes, especificados no momento do dimensionamento. No item 4.3.3.3 é feita uma comparação entre os resultados calculados pelo EasyLink e os resultados obtidos na simulação.

4.3.3.2 - Simulações com os resultados do Enlace 2

Na Tab. 4.4 são mostrados os resultados obtidos na simulação, para cada classe de prioridade, conforme os exemplos mostrados na Tab. 4.2. As capacidades obtidas para o enlace e para o *buffer*, através do modelo de dimensionamento com prioridades, proposto no capítulo anterior, também são utilizadas como parâmetros de configuração para a simulação. No apêndice II são mostrados os relatórios obtidos nas simulações, através do software Arena.

Exemplos		Parâmetros utilizados na simulação				Resultados da simulação	
		λ	$E\{x\}$	C	N	$E\{T\}$	PB
1	voz	20	500	51139	37	0,021	$1,0 \times 10^{-3}$
	vídeo	30	500			0,031	$9,6 \times 10^{-4}$
	dados	40	500			0,147	$1,0 \times 10^{-3}$
2	voz	20	50	16307	23	0,024	$8,6 \times 10^{-4}$
	vídeo	20	100			0,032	$9,1 \times 10^{-4}$
	dados	20	500			0,147	$9,1 \times 10^{-4}$
3	voz	20	500	33809	40	0,033	$8,01 \times 10^{-4}$
	vídeo	20	500			0,060	$8,5 \times 10^{-4}$
	dados	20	500			0,288	$7,8 \times 10^{-4}$
4	voz	20	500	36620	61	0,029	6×10^{-7}
	vídeo	20	500			0,048	6×10^{-7}
	dados	20	500			0,149	9×10^{-7}
5	voz	30	50	9529	81	0,042	$3,5 \times 10^{-7}$
	vídeo	20	100			0,068	$4,8 \times 10^{-7}$
	dados	10	500			0,499	$3,8 \times 10^{-7}$

Tabela 4.4 - Resultados obtidos através das simulações do Enlace 2.

A partir dos resultados da Tab. 4.4 é possível verificar se as capacidades do enlace e do *buffer*, obtidas através do EasyLink, atendem aos requisitos de atraso e de perda de pacotes, de cada classe de prioridade, especificados no momento do dimensionamento. No item seguinte, é mostrada uma comparação entre os resultados obtidos através do EasyLink e através da simulação.

4.3.3.3 - Validação dos modelos de dimensionamento propostos

A validação dos modelos de dimensionamento, propostos no capítulo 3 (Modelo FIFO e modelo com prioridades), é feita através da comparação dos requisitos de QoS - atraso médio e perda de pacotes - onde estes são obtidos, primeiramente, pelo EasyLink e, posteriormente, através das simulações (Arena). Na Tab. 4.5 é mostrada a comparação entre os resultados referentes ao Enlace 1, Tabs. 4.1 e 4.3.

Exemplos	Resultados do EasyLink		Resultados da simulação	
	$E\{T\}$	PB	$E\{T\}$	PB
1	0,012	$1,0 \times 10^{-3}$	0,012	$6,0 \times 10^{-4}$
2	0,006	$1,0 \times 10^{-3}$	0,006	$5,4 \times 10^{-4}$
3	0,004	$1,0 \times 10^{-3}$	0,004	$5,8 \times 10^{-4}$
4	0,012	$1,0 \times 10^{-3}$	0,012	$5,8 \times 10^{-4}$
5	0,012	$1,0 \times 10^{-3}$	0,012	$5,6 \times 10^{-4}$
6	0,012	$1,0 \times 10^{-3}$	0,012	$5,6 \times 10^{-4}$
7	0,001	$3,5 \times 10^{-11}$	0,001	$< 10^{-6}$
8	0,005	$1,1 \times 10^{-5}$	0,005	$< 10^{-6}$
9	0,010	$5,0 \times 10^{-4}$	0,010	$1,8 \times 10^{-4}$
10	0,007	$1,0 \times 10^{-4}$	0,007	$2,6 \times 10^{-5}$
11	0,005	$1,0 \times 10^{-5}$	0,005	$5,0 \times 10^{-6}$
12	0,004	$1,0 \times 10^{-6}$	0,004	$1,0 \times 10^{-6}$
13	0,102	$1,0 \times 10^{-3}$	0,091	$4,6 \times 10^{-4}$
14	0,150	$1,0 \times 10^{-4}$	0,131	$3,0 \times 10^{-5}$
15	0,150	$1,0 \times 10^{-30}$	0,145	$1,0 \times 10^{-3}$

Tabela 4.5 - Comparação entre os resultados referentes ao Enlace 1.

A comparação entre os resultados do dimensionamento do Enlace 2, obtidos pelo EasyLink, Tab. 4.2, com os resultados obtidos na simulação, Tab. 4.4, estão mostrados na Tab. 4.6.

Exemplos		Resultados do EasyLink		Resultados da simulação	
		$E\{T\}$	PB	$E\{T\}$	PB
1	voz	0,020	$1,0 \times 10^{-3}$	0,021	$1,0 \times 10^{-3}$
	vídeo	0,030	$1,0 \times 10^{-3}$	0,031	$9,6 \times 10^{-4}$
	dados	0,150	$1,0 \times 10^{-3}$	0,147	$1,0 \times 10^{-3}$
2	voz	0,024	$1,0 \times 10^{-3}$	0,024	$8,6 \times 10^{-4}$
	vídeo	0,032	$1,0 \times 10^{-3}$	0,032	$9,1 \times 10^{-4}$
	dados	0,150	$1,0 \times 10^{-3}$	0,147	$9,1 \times 10^{-4}$
3	voz	0,033	$1,0 \times 10^{-3}$	0,033	$8,010^{-4}$
	vídeo	0,060	$1,0 \times 10^{-3}$	0,060	$8,5 \times 10^{-4}$
	dados	0,300	$1,0 \times 10^{-3}$	0,288	$7,8 \times 10^{-4}$
4	voz	0,030	$1,0 \times 10^{-6}$	0,029	6×10^{-7}
	vídeo	0,048	$1,0 \times 10^{-6}$	0,048	6×10^{-7}
	dados	0,150	$1,0 \times 10^{-6}$	0,149	9×10^{-7}
5	voz	0,042	$1,0 \times 10^{-5}$	0,042	$3,5 \times 10^{-7}$
	vídeo	0,068	$1,0 \times 10^{-5}$	0,068	$4,8 \times 10^{-7}$
	dados	0,500	$1,0 \times 10^{-5}$	0,499	$3,8 \times 10^{-7}$

Tabela 4.6 - Comparação entre os resultados referentes ao Enlace 2.

Nas Tab. 4.5 e 4.6 é observada a coerência entre os valores dos atrasos e das perdas de pacotes fornecidas pelas simulações e os valores destes mesmos parâmetros fornecidos pelo EasyLink. Com isso, é estabelecida uma validação discreta dos modelos de dimensionamento, propostos no capítulo anterior, com base nos exemplos mostrados nos itens 4.3.1 e 4.3.2.

4.4 - Considerações sobre os Resultados do Dimensionamento

Os resultados dos dimensionamentos apresentados nas Tabs. 4.1 e 4.2 são absolutamente coerentes do ponto de vista didático. Entretanto, algumas considerações devem ser feitas para aplicações práticas desses resultados. Nesse item são feitas duas observações acerca da sensibilidade do dimensionamento em relação ao tráfego de entrada do enlace.

Na primeira observação, considera-se que a intensidade do tráfego de entrada do enlace, ou a taxa de chegadas de pacotes cresça repentinamente. Na segunda, deseja-se utilizar os modelos propostos para o dimensionamento de enlaces, onde o tráfego a ser transportado seja moldado por distribuições estatísticas diferentes das consideradas pelo modelamento (ex.: chegada de pacotes conforme uma distribuição *on-off* e tamanho de pacotes exponencial negativo). Em ambas as observações, são feitas análises de como o desempenho da rede é afetado em termos de atraso médio e taxa de perda de pacotes.

4.4.1 - Oscilações na intensidade do tráfego de entrada do enlace

Os gráficos mostrados na Fig. 4.6 mostram o comportamento da probabilidade de perda de pacotes e do atraso médio, conforme a variação da intensidade do tráfego de entrada, para o enlace do exemplo 13 (Tab. 4.1). Para tanto, foram utilizadas as Eqs. 3.5 e 3.12.

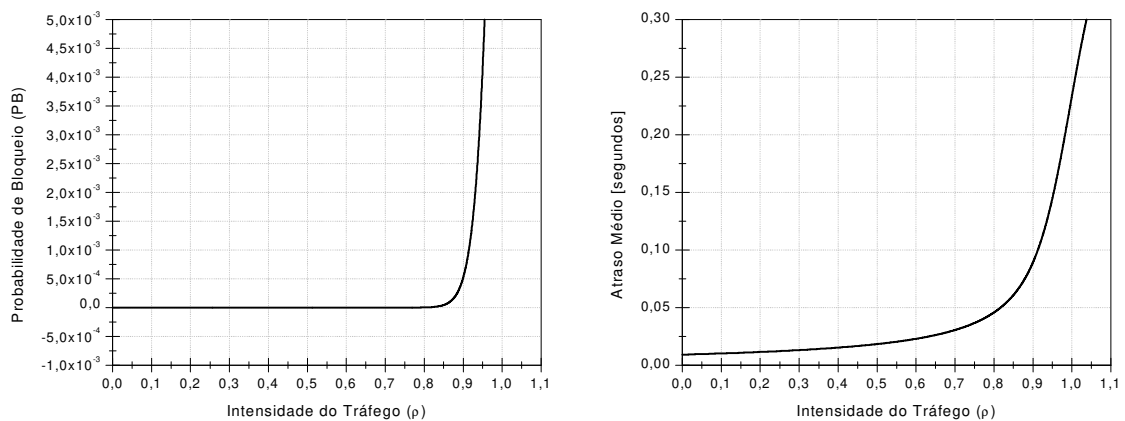


Figura 4.6 - Comportamento da probabilidade de perda e do atraso médio dos pacotes para o exemplo 13 da Tab. 4.1.

A partir dos dados da Tab. 4.1 e da Eq. 3.9, tem-se que a intensidade do tráfego para o enlace é aproximadamente $\rho = 0,92$. Com essa intensidade de tráfego, a probabilidade de perda e o atraso médio dos pacotes especificados no dimensionamento são satisfeitos, conforme Tab. 4.1. Porém, um pequeno acréscimo na taxa de chegada resultaria numa grande perda no desempenho da rede, conforme ilustrado na Fig. 4.6. Em casos semelhantes a esse, no ato do dimensionamento, é aconselhável que se especifique valores para a capacidade do enlace que sugiram uma certa margem de segurança para o desempenho da rede. Por exemplo, no caso observado, seria mais seguro trabalhar com a intensidade do tráfego próximo a 0,5 ao invés de 0,92.

4.4.2 - Tráfego de entrada com outra distribuição estatística

Nessa seção, é considerado que o tráfego de entrada do enlace possui a chegada dos pacotes moldada por uma distribuição *on-off* [22], e que o tamanho dos pacotes seja dado por uma distribuição exponencial negativa. A capacidade do enlace, obtida através dos modelos propostos (Tabs. 4.1 e 4.2) é, então, utilizada para a realização de simulações. Por fim, é analisado o quanto essa capacidade satisfaz os

requisitos de desempenho da rede (atraso médio e probabilidade de bloqueio dos pacotes), quando a chegada dos pacotes for representada por uma fonte *on-off*.

Na Tab. 4.7 estão mostrados os resultados, obtidos através do ARENA, referentes às simulações do exemplo 1 da Tab. 4.3 (modelo FIFO). Para tanto, considera-se que o valor da taxa média de chegada dos pacotes, para a fonte de tráfego *on-off*, seja dado por:

$$\lambda_{on-off} = \lambda_{on} \frac{t_{on}}{(t_{on} + t_{off})} \quad (4.1)$$

onde, λ_{on} é a taxa média de geração de pacotes (em pacotes/seg) quando a fonte está no estado *on*, t_{on} é o tempo médio (em segundos) que a fonte de tráfego permanece no estado *on* e t_{off} é o tempo médio (em segundos) que a fonte de tráfego permanece no estado *off*. Os parâmetros t_{on} e t_{off} representam valores médios de distribuições exponenciais negativas. Além disso, o valor de λ_{on-off} é considerado equivalente ao valor de λ (distribuição poissoniana) do exemplo 1 da Tab. 4.3.

Simulações	Parâmetros utilizados na simulação							Resultados da simulação	
	λ_{on-off}	t_{on}	t_{off}	λ_{on}	$E\{x\}$	C	N	$E\{T\}$	PB
1	100	0,25	1	500	500	92284	10	0,049	$5,6 \times 10^{-1}$
2	100	0,5	1	300	500	92284	10	0,047	$3,4 \times 10^{-1}$
3	100	1	1	200	500	92284	10	0,035	$1,1 \times 10^{-1}$
4	100	1	0,5	150	500	92284	10	0,021	$1,7 \times 10^{-2}$
5	100	1	0,05	105	500	92284	10	0,012	$7,7 \times 10^{-4}$

Tabela 4.7 - Simulações para o exemplo 1 da Tab. 4.3, considerando fonte de tráfego *on-off*.

Análise semelhante pode ser estendida para o caso do modelo de dimensionamento com prioridade (item 3.3.1). Nesse caso, considerou-se o exemplo 1 da Tab. 4.4 e, através de simulações no ARENA, obteve-se os resultados apresentados na Tab. 4.8.

Simulações	Serviço	Parâmetros utilizados na simulação							Resultados da simulação	
		λ_{on-off}	t_{on}	t_{off}	λ_{on}	$E\{x\}$	C	N	$E\{T\}$	PB
1	voz	20	0,25	1	100	500	51139	37	0,046	$1,9 \times 10^{-1}$
	vídeo	30	0,25	1	150	500			0,151	$2,3 \times 10^{-1}$
	dados	40	0,25	1	200	500			0,363	$3,2 \times 10^{-1}$
2	voz	20	1	1	40	500	51139	37	0,023	$8,2 \times 10^{-2}$
	vídeo	30	1	1	60	500			0,048	$1,0 \times 10^{-1}$
	dados	40	1	1	80	500			0,367	$1,3 \times 10^{-1}$
3	voz	20	1	0,05	21	500	51139	37	0,021	$3,0 \times 10^{-4}$
	vídeo	30	1	0,05	31,5	500			0,031	$5,0 \times 10^{-4}$
	dados	40	1	0,05	42	500			0,139	$1,0 \times 10^{-4}$

Tabela 4.8 - Simulações para o exemplo 1 da Tab. 4.4, considerando fonte de tráfego *on-off*.

O parâmetro λ_{on-off} (fonte *on-off*) das Tabs. 4.7 e 4.8 tem valor equivalente ao de λ (fonte poissoniana) das Tabs. 4.3 e 4.4, respectivamente. A Eq. 4.1 é utilizada para a obtenção dos outros parâmetros das simulações - λ_{on} , t_{on} e t_{off} . Na Eq. 4.1, quando t_{on} é relativamente maior que t_{off} , a taxa média de pacotes gerada pela fonte *on-off* em todo o período (λ_{on-off}) se aproxima da taxa média de pacotes gerada apenas quando a fonte está no estado *on* (λ_{on}). Em casos onde isso ocorre ($t_{on} \gg t_{off}$) e quando a capacidade do enlace é calculada levando-se em conta fontes poissonianas o desempenho da rede será pouco afetado. Em casos diferentes do descrito acima, ou seja, quando a diferença entre λ_{on} e λ_{on-off} for relativamente alta, o desempenho da rede pode ser bastante afetado. Comparando-se os resultados das Tabs. 4.7 e 4.8 com as Tabs. 4.3 e 4.4, respectivamente, pode-se verificar essas observações.

Uma observação, especificamente em relação à taxa de perda de pacotes (PB), é que quando a fonte *on-off* está no estado *on* o surto de pacotes provocados por λ_{on} pode ser suficientemente grande, provocando o *overflow* do *buffer*, e conseqüentemente resultando numa quantidade indesejada de perda de pacotes. Esse efeito, indesejado, pode ser contornado fazendo-se uma ampliação na capacidade do *buffer*. Essa observação se sustenta no fato de que o efeito causado pela combinação dos tempos t_{on} e t_{off} (responsável pelo surto de pacotes), mostrado nas Tabs. 4.7 e 4.8, é muito mais significativo em relação à taxa de perda de pacotes do que em relação ao atraso médio dos pacotes.

4.5 - Conclusões

Neste capítulo foi discutido o desenvolvimento de um software (o EasyLink), utilizado como ferramenta de auxílio à tarefa de projeto de redes. Esta ferramenta realiza de forma automatizada o cálculo da capacidade dos enlaces de uma rede, bastando que um usuário forneça a topologia e alguns parâmetros de configuração dessa rede. A essência do EasyLink consiste na implementação dos modelos de dimensionamento propostos no capítulo anterior - modelo FIFO, com *buffer* limitado, e modelo com prioridades, também com *buffer* limitado. Também são feitos alguns exemplos de dimensionamento de enlaces, através do EasyLink, utilizando estes dois modelos. Ao final, através da comparação dos resultados obtidos pelo EasyLink e dos resultados obtidos em simulações, através do Arena, uma validação discreta para os dois modelos foi feita.

A utilização do EasyLink para a realização do cálculo da capacidade de enlaces, como os mostrados nas Tabs. 4.1 e 4.2, torna esta parte da tarefa de dimensionamento de redes mais rápida e mais eficiente. Isso porque, além da velocidade obtida pela automatização computacional o software, também, deixa os usuários livres da realização de cálculos complexos.

As validações dos exemplos, citados no parágrafo anterior, são obtidas através da comparação dos resultados, mostrados nas Tabs. 4.5 e 4.6, sendo que estas comparações sugerem que os modelos de dimensionamento propostos são adequados às tarefas de cálculo da capacidade de enlaces, em redes que demandam a necessidade de prover diferentes requisitos de QoS. Nas Tabs. 4.7 e 4.8 são mostrados os resultados de algumas simulações, as quais evidenciam a sensibilidade dos modelos de dimensionamento propostos quanto às variações no tráfego de entrada do enlace.

Novos modelos de cálculo de capacidade de enlaces (ex.: RR, WRR, WFQ, etc) ou outras funcionalidades (ex.: minimização de custos, distribuição de labels MPLS, etc) podem ser incorporados facilmente ao EasyLink, pois a arquitetura deste software é dividida em dois planos de programação independentes - um que trata a topologia da rede (parte gráfica) e outro que trata os assuntos relacionados às suas funcionalidades. Dessa forma, novas funcionalidades podem se beneficiar com a utilização de informações obtidas pela parte gráfica já desenvolvida.

Capítulo 5

Conclusões

Neste trabalho é tratado o dimensionamento da capacidade de enlaces em redes de comutação de pacotes. Atualmente as técnicas de dimensionamento utilizadas constituem-se basicamente na minimização do atraso dos pacotes entre os nós de origem e de destino. Dessa forma, as redes não possuem qualquer tipo de garantia de QoS, isto é, elas provêem o máximo de recursos disponíveis, mas não podem garantir nem mesmo que os pacotes alcancem o seu destino.

Os interesses acerca deste assunto vieram à tona com o surgimento de tecnologias para o transporte de forma integrada, em uma única rede, dos serviços de voz, vídeo e dados - redes digitais de serviços integrados (RDSI). Porém, em uma RDSI é desejável que condições mínimas de QoS sejam garantidas para todos os tipos de serviços. Uma das formas de se prover QoS nestas redes é através do dimensionamento adequada da capacidade de seus enlaces, ou seja, uma largura mínima de banda é garantida aos diferentes tipos de serviços.

No entanto, no projeto de redes com integração de serviços, a complexidade do dimensionamento está relacionada com a diversidade dos parâmetros (largura de banda, atraso, QoS, topologia da rede, custos, etc). A fim de simplificar o cálculo da capacidade de um enlace, foi feita a escolha de apenas dois dos principais parâmetros de QoS - atraso médio dos pacotes e a taxa de perda de pacotes. E então, foram propostos dois métodos para a realização desta tarefa. No primeiro método, denominado modelo FIFO, o escalonador encaminha os pacotes armazenados num *buffer* limitado do tipo FIFO, sem distinção entre os pacotes. O segundo método, denominado modelo de prioridades, o escalonador encaminha os pacotes armazenados num *buffer* limitado, conforme prioridades pré-definidas.

O critério de dimensionamento de enlaces adotado no modelo FIFO considera os parâmetros de QoS dos pacotes que pertençam a tipos de serviço mais restritivos (ex.: atraso médio para pacotes de voz e taxa de perda para pacotes de dados). Dessa forma, apesar da capacidade encontrada para o enlace atender os dois requisitos de QoS especificados, ela pode também estar sendo superestimada e, conseqüentemente, o enlace estará sendo mal utilizado.

No modelo de prioridades, o critério de dimensionamento é dividido em duas etapas. Na primeira etapa, o cálculo da capacidade dos enlaces é feito levando-se em conta o atraso médio dos pacotes, para cada classe, considerando a capacidade do *buffer* infinita. Na segunda etapa, para garantir a taxa de perda de pacotes de todas as classes, é proposto um modelo de cálculo da capacidade equivalente do *buffer*. Como são levados em conta os parâmetros de QoS de todas as classes de serviço, e não somente os da classe de serviço mais restritiva, é possível se obter uma melhor utilização do enlace, comparado ao modelo FIFO. Enquanto que no modelo FIFO a capacidade do *buffer* deve ser um parâmetro conhecido, este modelo fornece meios para a obtenção de uma capacidade equivalente para o *buffer*.

Os dois modelos de dimensionamento propostos requerem a realização de cálculos complexos, tais como a aproximação numérica de raízes de polinômios de grau N (modelo FIFO) ou a resolução de equações do terceiro grau (modelo de prioridades). No projeto de uma rede estes processos de cálculo são realizados para todos os enlaces. Com o propósito de dar agilidade e confiabilidade à resolução destes processos, foi desenvolvido um software interativo (o EasyLink) como ferramenta de dimensionamento de redes.

O EasyLink destina-se exclusivamente à automatização do cálculo da capacidade dos enlaces, bastando apenas que sejam fornecidas a topologia da rede, através de uma interface gráfica, e alguns parâmetros de configuração desta. A arquitetura do software é dividida basicamente em dois planos de programação independentes - um que trata a topologia da rede (parte gráfica) e outro que cuida dos modelos matemáticos. Dessa forma, novos modelos de dimensionamento podem ser adicionados, sendo que estes podem se beneficiar das facilidades gráficas já desenvolvidas.

Alguns exemplos de cálculo de capacidade de enlaces estão mostrados no item 4.3. Estes exemplos levaram em conta os dois modelos de dimensionamento propostos e foram obtidos com a utilização do EasyLink. Uma validação discreta para estes exemplos foi feita através da comparação dos resultados obtidos pelo EasyLink e dos resultados obtidos em simulações, feitas através de um software de simulação dinâmica de eventos discretos - O ARENA (Versão 3.5 - Demo). Estas comparações, conforme Tabs. 4.5 e 4.6, sugerem que os modelos de dimensionamento propostos são adequados ao cálculo da capacidade de enlaces em redes que tenham a necessidade de prover diferentes requisitos de QoS.

O cálculo da capacidade de enlaces através das metodologias propostas pode ser de grande utilidade para o projeto de redes. Porém, deve-se levar sempre em conta as características estatísticas do tráfego utilizado, ou seja, o resultado final do dimensionamento de um enlace qualquer pode ser complementado a partir de algumas análises de sensibilidade quanto a variações no tráfego de entrada do enlace. No item 4.4 foram feitas duas considerações a esse respeito. Na primeira, observou-se o comportamento do desempenho da rede após variações na intensidade média do tráfego. Na segunda, verificou-se quanto o desempenho da rede é afetado quando há alterações na distribuição estatísticas do tráfego.

Por fim, ficam algumas sugestões para trabalhos futuros: Uma primeira sugestão seria sobre o estudo de novos modelos para o cálculo da capacidade de enlaces, considerando outras disciplinas de escalonamento (ex.: RR, WRR, WFQ, etc). Outra sugestão seria a agregação, ao software desenvolvido, de novas funcionalidades (ex.: minimização de custos, distribuição de labels MPLS, etc) e de novos parâmetros de QoS. Uma última sugestão seria a possibilidade de se utilizar no modelo FIFO o mesmo critério de dimensionamento adotado para o modelo de prioridades, ou seja, a capacidade do enlace seria calculada conforme o atraso médio de pacotes definido, baseando-se numa fila M/M/1, e uma capacidade equivalente para o *buffer* seria calculada conforme a taxa de bloqueio de pacotes também definida.

Referências Bibliográficas

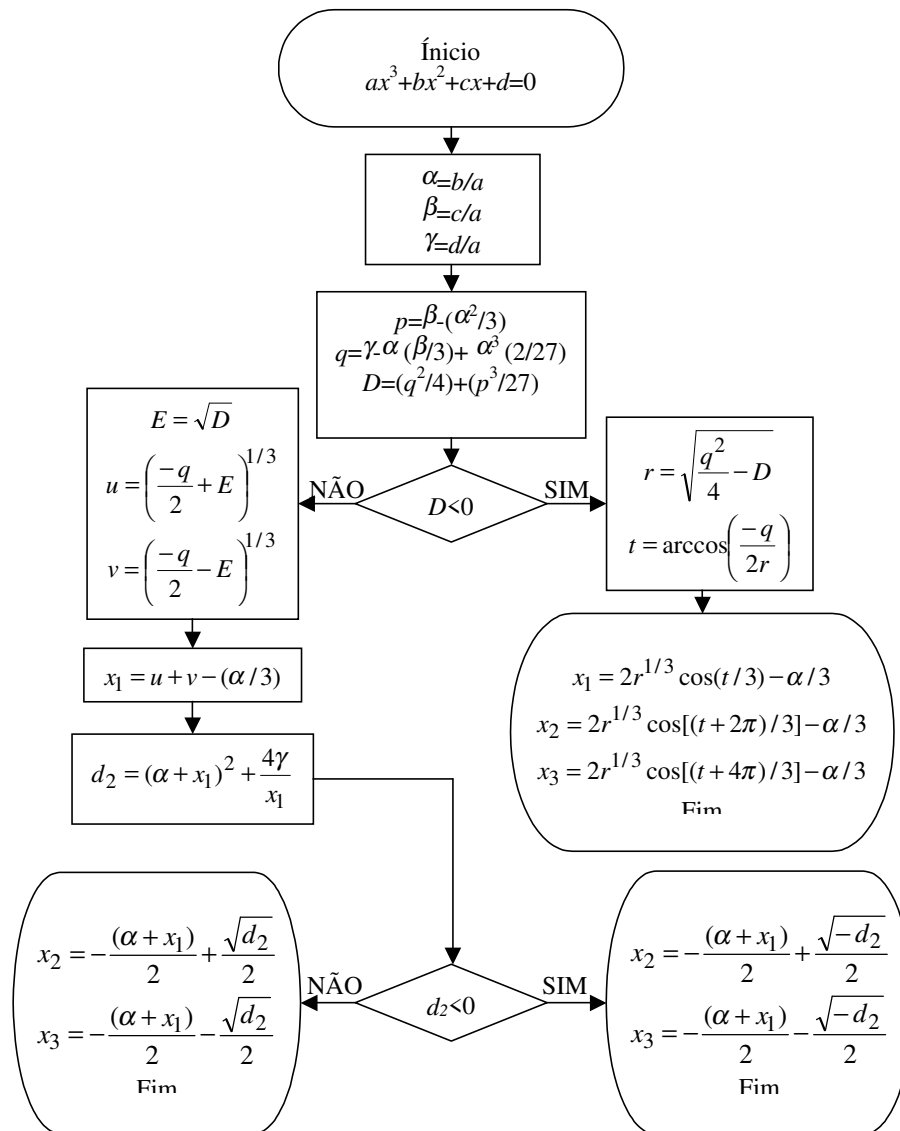
- [1] P. Newman. "Traffic management for ATM local area networks". IEEE Communication Magazine, pp. 44-50, August 1994.
- [2] "Integrated Services charter".
Documento disponível em: <http://www.ietf.org/html.charters/intserv-charter.html>.
- [3] Braden, R. et al. "Resource Reservation Protocol (RSVP) Version 1 Functional Specification". RFC2205. Setembro, 1997. Documento disponível em: <http://www.ietf.org/rfc/rfc2205.txt>
- [4] P. P. White. "RSVP and integrated services in the Internet: A Tutorial". IEEE Network, vol. 34, nº 5, pp. 100-106, May 1997.
- [5] "Differentiated Services charter".
Documento disponível em: <http://www.ietf.org/html.charters/diffserv-charter.html>.
- [6] E. C. Rosen, A. V. Than and R. Callon. "Multiprotocol label switching architecture". *IETF draft*.
Documento disponível em: <http://search.ietf.org/internet-drafts/draft-ietf-mpls-arch-06.txt>.
- [7] Krishna M. Sivalingam, Suresh Subramanian. "Optical WDM Networks - *Principles and Practice*". Chap. 16, Kluwer Academic Publishers, Third Printing 2001.
- [8] Soares, Luiz Fernando Gomes; LEMOS, Guido; COLCHER; Sérgio. "Redes de Computadores: das LANs MANs e WANs às Redes ATM". 2. ed. Rio de Janeiro: editora Campus, 1995.
- [9] Jacobson, V., et al. "An Expedited Forwarding PHB". RFC2598. Junho, 1999.
Documento disponível em: <http://www.ietf.org/rfc/rfc2598.txt>
- [10] Xiao, Xipeng and Ni, Lionel M. - "Internet Qos: A Big Picture". IEEE Network Magazine, Mar. 2000. Documento disponível em: <http://www.cse.msu.edu/~xiaoxipe>
- [11] Tutorial H.323. Material utilizado na disciplina EE981 - Telefonia. FEEC-Unicamp, 2002.
Documento disponível em: <http://www.dt.fee.unicamp.br/~motoyama/ee981/aulas/H323.pdf>
- [12] Prycker, Martin De. "Asynchronous Transfer Mode, Solution for Broadband ISDN". Third Edition. Prentice Hall, 1995.
- [13] Casner, et al. "RTP: A Transport Protocol for Real-Time Applications".
RFC 1889. January, 1996. Documento disponível em: <http://www.ietf.org/rfc/rfc1889.txt>
- [14] Kostas, Thomas J., et al. "Real-Time Voice Over Packet-Switched Network". IEEE Network, January/February 1998, pp. 18-27.

- [15] Cisco Systems Inc. “Quality of Service Solutions Configuration Guide, Classification Overview”. June, 1999. Documento disponível em: <http://www.cisco.com/>
- [16] Cisco Systems Inc. “Planning for Quality of Service”. May, 2000.
Documento disponível em: <http://www.cisco.com/>
- [17] Cisco Systems Inc. “Quality of Service for Voice over IP Solutions Guide”. July, 2001.
Documento disponível em: <http://www.cisco.com/>
- [18] Cisco Systems Inc. “Quality of Service Solutions Configuration Guide, Congestion Management Overview”. April, 2001. Documento disponível em: <http://www.cisco.com/>
- [19] Magalhães, Maurício F.; Cardoso, Eleri. “Qualidade de Serviço na Internet”. Versão Draft. Unicamp, 1999. Documento disponível em: <http://www.dca.fee.unicamp.br/~mauricio>
- [20] Schwartz, Mischa. “Computer-Communication Network Design and Analysis”. Chap. 3, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1977.
- [21] J. C. de Oliveira. “Dimensionamento de Enlaces em Redes de Telecomunicações ATM”. Dissertação de Mestrado. FEEC - UNICAMP. Fevereiro de 1998.
Documento disponível em: <http://users.ece.gatech.edu/~jau/publicacoes.html>
- [22] Bengt Ahlgren, Anders Andersson, Olof Hagsand and Ian Marsh. “Dimensioning Links for IP Telephony”. IPTEL, 2001. Documento disponível em: <http://citeseer.nj.nec.com/473527.html>
- [23] Robertazzi, Thomas G., “Computer Networks and Systems: Queuing Theory and Performance Evaluation”, Cap. 2, Springer-Verlag, 1994.
- [24] Ruggiero, Márcia A. Gomes, “Cálculo Numérico: Aspectos teóricos e computacionais”, Cap. 2, McGraw-Hill, 1988.
- [25] DaSilva, Luiz A., “Introduction to Queuing Theory (Part II)”, NET-2, Fall 1999, Virginia Tech. Documento disponível em: <http://www.ee.vt.edu/~ee5984ld/>
- [26] Sodré, Ulysses. “Raízes da Equação do Terceiro Grau”. Novembro, 2000.
Documento disponível em: <http://pessoal.sercomtel.com.br/matematica/medio/215/tartagli.htm>
- [27] J. MacGregor Smith and F. R. B. Cruz, “The Buffer Allocation Problem for General Finite Buffer Queuing Network”, September 2000.
Documento disponível em: <http://www.ecs.umass.edu/mie/faculty/smith/>

Apêndice I

Raízes de Polinômios do 3º Grau [32]

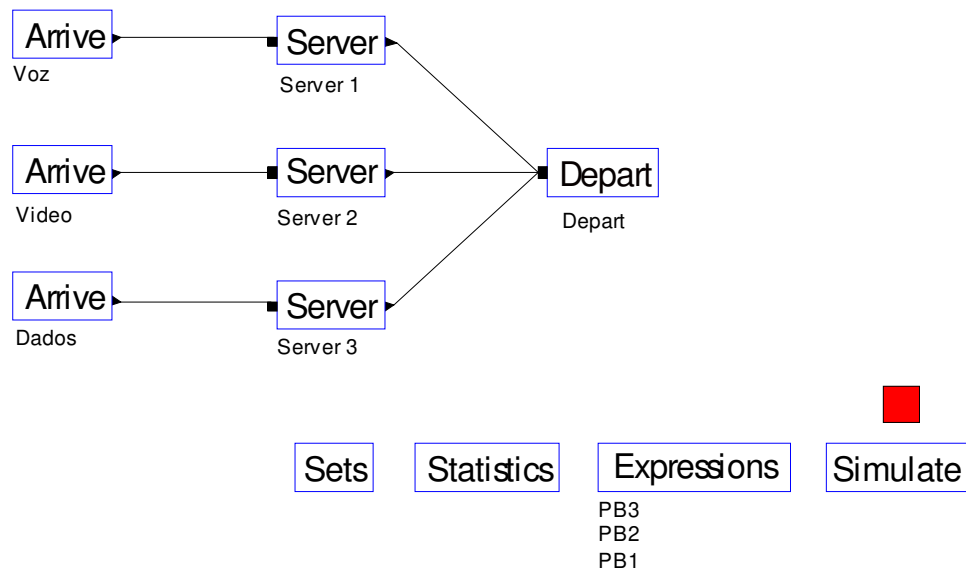
Considere um polinômio do terceiro grau dado por $f(x) = ax^3 + bx^2 + cx + d$, onde a, b, c e d são coeficientes reais não nulo. O fluxograma abaixo apresenta o método de Tartaglia, também conhecido como método de Cardano, para resolução deste tipo de equação.



Apêndice II

Relatórios das simulações feitas através do Arena

Os relatórios mostrados a seguir são referentes às simulações dos exemplos de dimensionamento do enlace 2, usando o modelo de filas com prioridade, mostrados na Tab. 4.4. Estas simulações foram realizadas a partir do esquema mostrado abaixo:



Simulação do exemplo 1 - modelo com prioridades

ARENA Simulation Results					
Summary for Replication 1 of 1					
Project:			Run execution date :	3/13/2002	
Analyst:			Model revision date:	3/13/2002	
TALLY VARIABLES					
Identifier	Average	Half Width	Minimum	Maximum	Observations
T1	.02047	6.1942E-05	2.0147E-08	.18128	998947
T2	.03066	1.3036E-04	1.6007E-09	.41847	150029E+01
T3	.14707	.00252	1.5098E-08	1.2852	199835E+01
DISCRETE-CHANGE VARIABLES					
Identifier	Average	Half Width	Minimum	Maximum	Final Value
Servidor Busy	.88036	.00140	.00000	1.0000	.00000
# in Buffer1	.21360	.00122	.00000	8.0000	.00000
# in Buffer2	.62634	.00370	.00000	16.000	.00000
# in Buffer3	5.4877	(Corr)	.00000	37.000	.00000
COUNTERS					
Identifier			Count	Limit	
VozGerados			1000000	Infinite	
VozAtendidos			998947	Infinite	
VideoGerados			1501741	Infinite	
VideoAtendidos			1500298	Infinite	
DadosGerados			2000397	Infinite	
DadosAtendidos			1998353	Infinite	
OUTPUTS					
Identifier			Value		
PB1			.00105		
PB2			9.6088E-04		
PB3			.00102		
Simulation run time: 49.22 minutes.					
Simulation run complete.					

Simulação do exemplo 2 - modelo com prioridades

ARENA Simulation Results					
Summary for Replication 1 of 1					
Project:			Run execution date :	3/13/2002	
Analyst:			Model revision date:	3/13/2002	
TALLY VARIABLES					
Identifier	Average	Half Width	Minimum	Maximum	Observations
T1	.02403	1.2512E-04	4.1728E-09	.37180	100056E+01
T2	.03179	1.2978E-04	7.3742E-09	.43279	100250E+01
T3	.14749	.00166	9.1080E-08	1.5675	999092
DISCRETE-CHANGE VARIABLES					
Identifier	Average	Half Width	Minimum	Maximum	Final Value
Servidor Busy	.79573	.00248	.00000	1.0000	.00000
# in Buffer1	.41883	.00270	.00000	14.000	.00000
# in Buffer2	.51354	.00321	.00000	16.000	.00000
# in Buffer3	2.3303	.04538	.00000	23.000	.00000
COUNTERS					
Identifier			Count	Limit	
VozGerados			1001430	Infinite	
VozAtendidos			1000568	Infinite	
VideoGerados			1003419	Infinite	
VideoAtendidos			1002502	Infinite	
DadosGerados			1000000	Infinite	
DadosAtendidos			999092	Infinite	
OUTPUTS					
Identifier			Value		
PB1			8.6077E-04		
PB2			9.1388E-04		
PB3			9.0800E-04		
Simulation run time: 28.62 minutes.					
Simulation run complete.					

Simulação do exemplo 3 - modelo com prioridades

ARENA Simulation Results					
Summary for Replication 1 of 1					
Project:			Run execution date :	3/13/2002	
Analyst:			Model revision date:	3/13/2002	
TALLY VARIABLES					
Identifier	Average	Half Width	Minimum	Maximum	Observations
T1	.03349	1.1279E-04	1.4261E-07	.40235	999240
T2	.06018	4.3054E-04	2.2837E-08	.82598	100043E+01
T3	.28795	.00508	4.6508E-07	2.5554	100084E+01
DISCRETE-CHANGE VARIABLES					
Identifier	Average	Half Width	Minimum	Maximum	Final Value
Servidor Busy	.88762	.00148	.00000	1.0000	.00000
# in Buffer1	.37305	.00173	.00000	12.000	.00000
# in Buffer2	.90767	.00934	.00000	20.000	.00000
# in Buffer3	5.4650	.09785	.00000	40.000	.00000
COUNTERS					
Identifier			Count	Limit	
VozGerados			1000042	Infinite	
VozAtendidos			999240	Infinite	
VideoGerados			1001287	Infinite	
VideoAtendidos			1000437	Infinite	
DadosGerados			1001623	Infinite	
DadosAtendidos			1000841	Infinite	
OUTPUTS					
Identifier			Value		
PB1			8.0197E-04		
PB2			8.4891E-04		
PB3			7.8073E-04		
Simulation run time: 28.15 minutes.					
Simulation run complete.					

Simulação do exemplo 4 - modelo com prioridades

ARENA Simulation Results					
Summary for Replication 1 of 1					
Project:			Run execution date :	3/13/2002	
Analyst:			Model revision date:	3/13/2002	
TALLY VARIABLES					
Identifier	Average	Half Width	Minimum	Maximum	Observations
T1	.02904	3.1124E-05	2.0169E-08	.31233	999999E+01
T2	.04751	1.0991E-04	1.8495E-08	.83406	100060E+02
T3	.14937	9.2237E-04	1.2224E-08	2.6275	100026E+02
DISCRETE-CHANGE VARIABLES					
Identifier	Average	Half Width	Minimum	Maximum	Final Value
Servidor Busy	.81916	3.9706E-04	.00000	1.0000	.00000
# in Buffer1	.30759	6.5275E-04	.00000	13.000	.00000
# in Buffer2	.67739	.00228	.00000	24.000	.00000
# in Buffer3	2.7139	.01793	.00000	60.000	.00000
COUNTERS					
Identifier			Count	Limit	
VozGerados			10000001	Infinite	
VozAtendidos			9999995	Infinite	
VideoGerados			10006099	Infinite	
VideoAtendidos			10006093	Infinite	
DadosGerados			10002658	Infinite	
DadosAtendidos			10002649	Infinite	
OUTPUTS					
Identifier			Value		
PB1			6.0000E-07		
PB2			5.9963E-07		
PB3			8.9976E-07		
Simulation run time: 276.90 minutes.					
Simulation run complete.					

Simulação do exemplo 5 - modelo com prioridades

ARENA Simulation Results					
Summary for Replication 1 of 1					
Project:			Run execution date :	3/18/2002	
Analyst:			Model revision date:	3/15/2002	
TALLY VARIABLES					
Identifier	Average	Half Width	Minimum	Maximum	Observations
T1	.04150	1.6517E-05	.00000	1.0036	290575E+03
T2	.06780	2.8828E-05	4.6566E-10	1.1499	193747E+03
T3	.49936	9.4837E-04	2.3283E-09	7.3139	968456E+02
DISCRETE-CHANGE VARIABLES					
Identifier	Average	Half Width	Minimum	Maximum	Final Value
Servidor Busy	.89189	1.6165E-04	.00000	1.0000	1.0000
# in Buffer1	1.0875	5.9797E-04	.00000	37.000	1.0000
# in Buffer2	1.1462	6.5913E-04	.00000	31.000	1.0000
# in Buffer3	4.4680	.01147	.00000	80.000	.00000
COUNTERS					
Identifier			Count	Limit	
VozGerados			290575562	Infinite	
VozAtendidos			290575459	Infinite	
VideoGerados			193747730	Infinite	
VideoAtendidos			193747637	Infinite	
DadosGerados			96845642	Infinite	
DadosAtendidos			96845605	Infinite	
OUTPUTS					
Identifier			Value		
PB1			3.5447E-07		
PB2			4.8001E-07		
PB3			3.8205E-07		
Simulation run time: 4223.80 minutes.					
Simulation run complete.					