

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

**Análise Comparativa de Desempenho de Redes IP e ATM
com tráfego Multimídia Interativo**

por

TATIANA ROTAVA SCHABBACH

Dissertação submetida à avaliação,
como requisito parcial para a obtenção do grau de Mestre
em Ciência da Computação

Prof. João Cesar Netto
Orientador

Porto Alegre, maio de 2002.

CIP – CATALOGAÇÃO DA DISSERTAÇÃO

Schabbach, Tatiana Rotava

Análise Comparativa de Desempenho de Redes IP e ATM com tráfego Multimídia Interativo / por Tatiana Rotava Schabbach. Porto Alegre: PPGC da UFRGS, 2002.

138 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2002. Orientador: Netto, João Cesar.

1. Redes 2. Desempenho 3. IP 4. ATM 5. QoS 6. Multimídia. I. Netto, João Cesar. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Profa. Wrana Panizi

Pró-Reitor de Ensino: Prof. José Carlos Ferraz Hennemann

Pró-Reitor Adjunto de Pós-Graduação: Prof. Jaime Evaldo Fensterseifer

Diretor do Instituto de Informática: Prof. Philippe Olivier Alexandre Navaux

Coordenador do PPCG: Prof. Carlos Alberto Heuser

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

Agradecimentos

Ao orientador, professor Dr. João Netto que foi sempre um tranquilizador e estimulador com suas palavras seguras nos momentos de dificuldades e incertezas.

Ao professor Dr. Juergen Rochol que é um amigo que encontrei na minha caminhada, e que sempre me ajudou muito com suas palavras positivas durante estes anos.

Ao CNPq pela bolsa de estudo que sem dúvida nenhuma contribuiu para a realização deste trabalho permitindo uma maior dedicação ao meu estudo.

Aos amigos especiais, Adriano Ramos Gonçalves e Eduardo Bastos, que me ajudaram nos momentos de dúvidas, dividindo comigo horas que poderiam estar passando com seus familiares.

A todas aquelas pessoas, amigas e conhecidas, que conviveram e que contactaram comigo pessoalmente e por e-mail nestes anos e que tiveram sensibilidade e em muitos momentos me ofereceram ajuda de diversas maneiras, ouvindo, falando, rindo, criticando, trocando experiências, ajudando na instalação da máquina e dos softwares utilizados para fazer esse trabalho, e muito, muito mais.

Aos "FADISTAS" que foram muito legais e companheiros durante estes anos.

Ao meu pai Laudelino que sempre me estimulou a investir em educação e que me apoiou a fazer este mestrado.

A minha mãe Lia e minha sogra Marlene que me ajudaram em muitos momentos dividindo comigo minhas responsabilidades de mãe.

Ao meu esposo Gilvan que com seu amor e seu apoio irrestrito deu-me a força necessária para eu ir em frente.

A minha filha Giulia que é a razão do meu viver e que teve menos tempo ao meu lado do que gostaria e que pela sua imensa compreensão com seus 3 anos de idade, é para ela que eu ofereço este trabalho.

E a DEUS que sempre está do nosso lado nos protegendo e nos livrando de todos os males e dando forças para vencermos as muitas dificuldades que a vida apresenta.

*"O futuro pertence àqueles que acreditam na
beleza dos seus sonhos."*

Eleonor Roosevelt

*"Dificuldade reais podem ser resolvidas;
apenas as imagináveis são insuperáveis."*

Thoedore N. Vail

*"Felicidade é a certeza de que a nossa vida
não está se passando inutilmente."*

Érico Veríssimo

"O tempo amadurece todas as coisas."

Nenhum homem nasce sábio."

Cervantes

Sumário

<u>Sumário</u>	4
<u>Lista de Abreviaturas</u>	6
<u>Lista de Figuras</u>	7
<u>Lista de Tabelas</u>	8
<u>Resumo</u>	9
<u>Abstract</u>	11
<u>1 Introdução</u>	13
<u>1.1 Motivações e objetivos</u>	13
<u>1.2 Organização do trabalho</u>	15
<u>2 Avaliação de desempenho em redes de comunicação de dados</u> ..	16
<u>2.1 Elementos relacionados em uma rede</u>	16
<u>2.2 Parâmetros Estatísticos de uma Variável Aleatória</u>	18
<u>3 Aplicações multimídia em redes de computadores</u>	21
<u>3.1 Qualidade de Serviço em redes</u>	22
3.1.1 <u>Definição de Qualidade de Serviço</u>	22
3.1.2 <u>Modelo de referência</u>	25
<u>3.2 Gerações de redes</u>	25
3.2.1 <u>Redes sem qualidade de serviço: IP</u>	25
3.2.2 <u>Redes com qualidade de serviço: ATM</u>	31
<u>3.3 Caracterização de Serviços em ATM e IP</u>	34
3.3.1 <u>Categorias de Serviços ATM</u>	35
3.3.2 <u>Serviços definidos na arquitetura IntServ/RSVP</u>	45
<u>3.4 Implementação de QoS</u>	47
3.4.1 <u>O Fenômeno congestionamento</u>	47
3.4.2 <u>Controle de Tráfego e Congestionamento</u>	47
3.4.3 <u>Parâmetros de QoS em IP e ATM</u>	51
<u>3.5 Modelagem de Tráfego</u>	53
3.5.1 <u>Caracterização de Tráfego</u>	53
3.5.2 <u>Modelos de Tráfego</u>	54
<u>4 Técnica desenvolvida</u>	58
<u>4.1 Método para avaliação de desempenho escolhido</u>	58

4.1.1	Simulação - Vantagens e Limitações	58
4.1.2	Simulação da rede ATM - Simulador ATM-HFC	58
4.1.3	Simulação da rede IP - Simulador NS-2	62
4.1.4	Validação	63
4.2	Método da Simulação	64
4.3	Cenários simulados	66
4.4	Valores Iniciais	68
4.4.1	Caracterização do tráfego VBR para os simuladores	69
4.5	Métricas utilizadas	70
4.5.1	Estimativa de CTD - Atraso	70
4.5.2	Estimativa de CDV - Variação do Atraso	72
4.5.3	Estimativa de CLR - Perda/Descarte	75
4.6	Processo das simulações	78
4.7	Análise dos Resultados	80
5	Considerações Finais	87
5.1	Conclusões	87
5.2	Trabalhos Futuros	91
	Anexo 1 Relação dos programas alterados no simulador NIST	92
	Anexo 2 Listagem dos principais "scripts" e arquivos das simulações para os simuladores NIST e NS-2	95
	Anexo 3 Execução dos Simuladores	124
	Bibliografia	133

Lista de Abreviaturas

AAL1	ATM Adaption Layer Tipo 1
ABR	Available Bit Rate
ATM	Asynchronous Transfer Mode
BTE	Broadband Terminal Equipment
CAC	Connection Admission Control
CBR	Constant Bit Rate
CDV	Cell Delay Variation
CLR	Cell Loss Ratio
CTD	Cell Transfer Delay
FIFO	First In First Out
INTSERV	Integrated Services
IP	Internet Protocol
ITU-T	International Telecommunications Union - Telecommunication
MBS	Maximum Burst Size
MCR	Minimum Cell Rate
PCM	Pulse Code Modulation
PCR	Peak Cell Rate
QoS	Quality of Service
RM	Resource Management
RSVP	Resource ReSerVation Protocol
RTP	Real-time Transport Protocol
RTCP	Real-Time Control Protocol
SCR	Sustainable Cell Rate
SS	Self-Similar
TCP	Transmission Control Protocol
UBR	Unspecified Bit Rate
UDP	User Datagram Protocol
VBR	Variable Bit Rate
VBR-nrt	Variable Bit Rate - non real time
VBR-rt	Variable Bit Rate - real time
VCI	Virtual Channel Identifier
VPI	Virtual Path Identifier
WAN	Wide Area Network

Lista de Figuras

FIGURA 2.1 - Elementos do Relacionamento de Dependência [GON 2001]	17
FIGURA 2.2 - Níveis de serviço (QoS), carga e recursos da rede [GON 2001].....	17
FIGURA 2.3 - Comprometimento dos níveis de serviço pela sobrecarga [GON 2001]	18
FIGURA 2.4 – Intervalo de confiança e nível de confiança [MCD 2000].....	19
FIGURA 3.1 - Necessidades de QoS para diferentes aplicações [MCD 2000]	24
FIGURA 3.2 – Qualidade de Serviço Fim-a-Fim [MCD 2000].....	25
FIGURA 3.3 - Fontes de atraso em transmissão de voz em redes de pacotes [DEL 99]	27
FIGURA 3.4 - Operação do protocolo RSVP [BER 99].....	29
FIGURA 3.5 - Reserva RSVP para um fluxo “unicast” [MCD 2000].....	31
FIGURA 3.6 – Estabelecimento de conexão ponto-a-ponto ATM [MCD 2000]	34
FIGURA 3.7 - Qualidade oferecida pela Rede ATM [GON 2001].....	35
FIGURA 3.8 - Variação na taxa de pico de uma conexão CBR [GON 2001]	37
FIGURA 3.9 - Chegada simultânea de células [GON 2001]	38
FIGURA 3.10 - Alocação pela taxa de pico para CBR e VBR [GON 2001].....	38
FIGURA 3.11 - Multiplexação estatística de fontes heterogêneas [MCD 2000].....	40
FIGURA 3.12 - BE: Banda Equivalente, $SCR \leq BE \leq PCR$ [GON 2001]	42
FIGURA 3.13 - Roteador/Switch genérico em um arquitetura com QoS [MCD 2000]	48
FIGURA 3.14 - Algoritmo do CAC pela taxa de pico [MCD 2000]	49
FIGURA 3.15 - Operação do RSVP em hosts e roteadores [MCD 2000]	50
FIGURA 3.16 - Pacotes/unidade de Tempo" x "Unidade de Tempo [STA 98].....	56
FIGURA 4.1 - Ações simuladas de um comutador ATM [GON 2001].....	60
FIGURA 4.2 - Estrutura de filas e escalonador por prioridade de categoria de serviço [GON 2001]	61
FIGURA 4.3 - Simulação do acesso dos equipamentos terminais à rede [GON 2001]	61
FIGURA 4.4 - NS-2: visão simplificada [CLA 2001].....	63
FIGURA 4.5 - Cenário de Simulação.....	65
FIGURA 4.6 - Estimativa do CTD e CDV para uma replicação [GON 2001].....	75
FIGURA 4.7 - Estimativa do CLR para duas replicações [GON 2001].....	78
FIGURA 4.8 - Intervalo de Confiança: Método de Replicação da Simulação.....	80
FIGURA 4.9 - Intervalo de Confiança: Método de Subintervalo da Simulação.....	80
FIGURA 4.10 - CLR para o fluxo auto-similar com banda variável	81
FIGURA 4.11 - CLR para os fluxos VBR agregados com banda variável.....	81
FIGURA 4.12 - CTD para o fluxo auto-similar com banda variável.....	82
FIGURA 4.13 - CTD para os fluxos VBR agregados com banda variável.....	82
FIGURA 4.14 - CDV para o fluxo auto-similar com banda variável	83
FIGURA 4.15 - CDV para os fluxos VBR agregados com banda variável	83
FIGURA 4.16 - CLR do fluxo auto-similar com banda fixa.....	84
FIGURA 4.17 - CLR do fluxo VBR agregado com banda fixa	84
FIGURA 4.18 - CTD para o fluxo auto-similar com banda fixa.....	85
FIGURA 4.19 - CTD para o fluxo VBR agregado com banda fixa	85
FIGURA 4.20 - CDV para o fluxo auto-similar com banda fixa	86
FIGURA 4.21 - CDV para o fluxo VBR agregado com banda fixa.....	86

Lista de Tabelas

TABELA 3.1 - Principais métricas de performance	24
TABELA 3.2 - Correspondência entre parâmetros e protocolos em IP e ATM	51
TABELA 3.3 - Parâmetros de QoS em redes ATM	51
TABELA 3.4 - Parâmetros de QoS em redes IP	52
TABELA 4.1 - Alocação de Recursos para Serviço de Vídeo Conferência	65
TABELA 4.2 - Cenários Simulados - Banda Variável (PCR Agregado)	67
TABELA 4.3 - Cenários Simulados - Banda Fixa (10Mbit/s)	67
TABELA 4.4 - Serviço VBR simulado	68
TABELA 4.5 - Serviço Auto-similar simulado	69
TABELA 4.6 - Parâmetros Descritores VBR Poisson	70
TABELA 4.7 - Exemplos de Requisitos de CTD e CDV	72

Resumo

A década de 80 é um marco para a área de comunicação de dados. Muitos estudos, pesquisas e especificações foram iniciados para permitir a integração de sistemas de comunicação para voz, vídeo e dados. Desde essa década, os cientistas procuram investir na Internet, que é um dos mais importantes e populares meios de transmissão de informações no mundo, para acomodar aplicações que permitam enviar imagens e sons por esta imensa rede de comunicação. Também nessa década, no final, o International Telecommunications Union – Telecommunication (ITU-T), especificou a tecnologia ATM, cujas pesquisas vinham ocorrendo desde o seu início.

O serviço de rede tradicional na Internet é transmissão de datagramas 'best-effort', conforme será visto neste trabalho. Neste serviço, os pacotes da origem são enviados para o destino, sem garantia de entrega. Para aquelas aplicações que requerem garantia de entrega, o protocolo TCP (equivalente à camada 4 do RM-OSI) se encarregará da retransmissão dos pacotes que falharam ao atingir o destino para então se conseguir uma recepção correta. Para aplicações de comunicação tradicionais, como FTP e Telnet, em que uma entrega correta é mais importante que a perda de tempo da retransmissão, este serviço é satisfatório. Entretanto, uma nova classe de aplicações, as quais usam mídias múltiplas (voz, som e dados), começam a aparecer na Internet. Exemplos desta classe de aplicação são: vídeo teleconferência, vídeo sob demanda e simulação distribuída. Operações de modo tradicional para estas aplicações resultam em redução da qualidade da informação recebida e, potencialmente, ineficiência do uso da largura de banda. Para remediar este problema é desenvolvido um ambiente de serviço para tempo real, no qual múltiplas classes de serviços são oferecidas. Este ambiente estende o modelo de serviços existentes para ir ao encontro das necessidades das aplicações multimídia com obrigatoriedade de tempo real, porém esta não é uma meta muito fácil. Contudo, a comunidade pesquisadora tem conseguido desenvolver alguns mecanismos que vão pouco a pouco permitindo que este objetivo se transforme em realidade.

O ATM é uma tecnologia que provê dutos de alta velocidade para a transferência de qualquer tipo de informação em pequenos pacotes de tamanho fixo, chamados células. A idéia básica é estabelecer entre dois pontos que desejam se comunicar um circuito virtual que é mantido pelos comutadores de células para levar a informação de um lado a outro. A característica marcante do ATM é a Qualidade de Serviço – QoS, que garante o desempenho predefinido que determinado serviço necessita. Isso permite suportar aplicações de tempo real que são sensíveis ao atraso. O atendimento à diversidade de características de tráfego e exigências distintas de grandes quantidades de serviços é possível pelo ATM graças ao controle de tráfego reunido à capacidade de multiplexação estatística do meio em altas taxas de transmissão.

O objetivo principal desta dissertação é elaborar uma comparação quantitativa e qualitativa do funcionamento de aplicações multimídia sobre IP com RSVP - Protocolo desenvolvido para realizar reservas de recursos integrante da arquitetura de Serviços Integrados (IntServ) proposta pelo IETF para oferecer qualidade de serviço para aplicações tais como aplicações multimídia. Essa proposta também inclui duas classes de serviços, sendo que nessa dissertação, o serviço de carga controlada é que está sendo utilizado. Isso deve-se a implementação dos módulos apresentados em [GRE 2001] e que foram utilizados na realização desse trabalho - e sobre ATM. A proposta final é a elaboração de uma técnica de análise baseado nas principais métricas de desempenho

em redes que deve permitir uma melhor visualização do comportamento das tecnologias sob determinadas cargas de tráfego, permitindo assim uma tomada de decisão entre qual das tecnologias que deverá ser adotada em um dado momento, em uma dada situação, ou seja, a indicação do ponto de quebra tecnológica na situação modelada.

Para que fosse possível fazer esta comparação, foi necessário dividir-se este trabalho em 3 grandes etapas, que são:

- Estudo e desenvolvimento da técnica para análise do elemento carga de tráfego na tecnologia ATM;
- Estudo e desenvolvimento da técnica para análise do elemento carga de tráfego na tecnologia IP com RSVP;
- Comparativo quantitativo e qualitativo dos estudos acima.

Palavras-Chave: Desempenho, Redes IP, Redes ATM, Qualidade de Serviço e Multimídia.

TITLE: “COMPARATIVE ANALYSIS OF PERFORMANCE ON IP AND ATM NETWORKS WITH INTERACTIVE MULTIMEDIA TRAFFIC”

Abstract

The decade of 1980 represents a new era for data communications. Several studies, researches and specifications were started to allow the integration of voice, video and data communication systems. Since then, scientists have been investigating the Internet, that is one of the most important and popular mean of information transmission in the world, to accommodate applications that allow the transmission of sound and image through this huge communication network. Moreover, by the end of this decade, the International Telecommunications Union – Telecommunication (ITU-T) specified the ATM technology, which had been developed since the beginning of the 1980's.

The traditional network service on the Internet is based on datagram transmission, and provides a best-effort approach to data transmission. In this service, packets from a source are sent to a destination with no guarantee of delivery. For those applications that require a guarantee of delivery, the TCP protocol, on the layer 4 of RM-OSI, will handle errors and will retransmit those packets that failed to reach the destination. For traditional computer-communication applications, such as FTP and Telnet, in which correct delivery is more important than timing guarantees, this service is satisfactory. However, a new class of applications that use multiple media (voice, video, and computer data) started appearing on the Internet. Examples of this class of applications are video teleconference, video-on-demand, and distributed simulation. For these applications, the traditional operation mode results in reduced quality of the received information and, potentially, inefficient use of bandwidth. To solve this problem, it is developed a real-time service environment in which multiple classes of service are provided. This environment will greatly extend the existing best-effort service model to meet the needs of multimedia applications with real-time constraints. However, this is not an easy task, since there are many details that complicate the process. Nevertheless, researchers have managed to develop mechanisms that are slowly turning this task into reality.

ATM is a technology that provides high-speed channels for transmitting any type of information in small, fixed size packets, called cells. The idea is to establish a virtual circuit between two connecting points, which must be handled by network commutators, in order to transmit data. The ATM main feature is Quality of Service, which provides the performance required by each different application. This characteristic supports real time applications that are time sensitive. With ATM, it is possible to support the diversity of traffic features and distinct requirements of many services due to the traffic control and the medium capacity to perform statistical multiplexing in high transmission rates.

The main objective of this dissertation is to elaborate a quantitative and qualitative comparison of the operation of multimedia on IP with RSPV - Protocol developed to perform resource reservation, is part of the Integrated Services (IntServ) architecture proposed by IETF to provide Quality of Service to applications like multimedia applications. This architecture also includes two classes of service, and this thesis uses the controlled load service. This is due to the implementation of the modules presented in [GRE 2001] that were used in this work - and on ATM. It aims to propose an analysis technique based on the main metrics of network performance, which should

allow a better visualization of technologies' behavior under certain traffic loads. That would allow choosing the technologies that should be adopted in each situation, suggesting the technological break point in the modeled situation.

To make this comparison possible, it was necessary to split this work into three parts, which are:

- To study and develop the appropriate technique for analysis of the traffic load element on ATM technology;
- To study and develop the appropriate technique for analysis of the traffic load element on IP technology with RSVP;
- Quantitative and qualitative comparison of the results obtained in the other parts of this work.

Keywords: performance, IP and ATM networks, Quality of Service and multimedia.

1 Introdução

1.1 Motivações e objetivos

Os desafios que surgem na transmissão de dados multimídia em rede podem ter como resposta a utilização das arquiteturas de software de rede e o desenvolvimento de hardware mais rápido.

Os protocolos base da Internet, TCP/IP e UDP/IP, oferecem um conjunto de serviços que podem ser usados pelas aplicações multimídia. Todavia, possuem limitações. Tais limitações estão sendo contornadas pelo desenvolvimento de protocolos para tempo-real, numa tentativa de tornar a grande rede um meio de transmissão adequado para a era da multimídia.

Por outro lado, a Internet cresce exponencialmente. De fato, a Internet se tornou a plataforma da maioria das atividades de rede no mundo. Esta é a primeira razão para desenvolver protocolos multimídia sobre a Internet. Outro benefício em rodar multimídia sobre IP é que os usuários podem integrar dados e serviços de multimídia numa mesma rede, sem grandes investimentos em outros equipamentos de rede e na construção de interfaces entre duas redes diferentes.

As redes ATM [GIR 99] [PRY 95], por sua vez, oferecem a largura de banda requerida por áudio e vídeo digital. Além disso, são orientadas a conexão e podem definir diferentes níveis de qualidade de serviço para diferentes tipos de aplicações (dependendo do tipo de tráfego). Todavia, exigem conexões dedicadas, equipamentos e instalações especiais, bem como novos softwares, que são fatores bastante dispendiosos. Por isso, até este momento muito poucos usuários têm ATM chegando nas redes de suas organizações e menos ainda têm conexões ATM nas suas estações de trabalho.

Dependendo da natureza da aplicação e dos equipamentos existentes em uma determinada empresa, a primeira decisão é a seleção entre redes baseada em IP ou ATM. Frequentemente, a maturidade de uma tecnologia ou o suporte para um determinado conjunto de aplicações determina o requerimento de equipamentos de rede. Até o presente momento, especialistas geralmente reconhecem o ATM como sendo a tecnologia mais madura em termos da implementação de qualidade de serviço e funções de gerenciamento de tráfego do que o IP. Entretanto, técnicas baseadas em IP, obtidas pesadamente da experiência obtida através do ATM, surgem estendendo e refinando capacidades em certas áreas. Desde que a principal "interface" com o usuário é, provavelmente para muitas aplicações, baseada em IP, muitos especialistas predizem que o mercado force rumo a maturidade de QoS e soluções de gerenciamento de tráfego em IP no início deste novo milênio.

Na situação atual de mercado, IP e Ethernet estão sendo utilizados preferencialmente em estações de trabalho e nas redes locais, enquanto o ATM é utilizado principalmente nos backbones das WANS. Ainda é prematuro imaginar que uma ou outra tecnologia venha a prevalecer em toda a extensão das redes. A expectativa é de que essa situação permaneça equilibrada, tendo em vista que os fabricantes das duas tecnologias continuam desenvolvendo soluções para diminuir suas limitações.

Nessa dissertação a avaliação de desempenho trata da verificação do

comportamento do tráfego de dados¹ quando acrescido tráfego multimídia² no enlace sobre a estrutura de duas tecnologias: ATM e IP com RSVP. Sendo os recursos da rede limitados, como capacidade de transmissão dos enlaces, a rede possui um ponto de saturação que depende de suas características e técnicas utilizadas para atender os serviços demandados. Partindo de uma determinada carga de serviço, especificada a partir de fatores externos como quantidade de tráfego, requisitos de desempenho e topologia das conexões, o comportamento de desempenho da rede depende exclusivamente das características dos fatores internos físicos, como quantidade de buffers, capacidade dos enlaces, capacidade de comutação, topologia dos enlaces e fatores internos de controle, como a configuração lógica dos mecanismos adotados para o controle de tráfego tais como algoritmos que implementam a admissão de conexões e tratamento de congestionamento. O aperfeiçoamento dos elementos de controle otimiza a utilização dos recursos físicos ajudando a aumentar a capacidade de serviço suportada pela rede.

Quando os recursos necessários forem iguais aos recursos disponíveis, o limite da rede foi alcançado. Nesse caso são duas as possibilidades para o equilíbrio: aumentar os recursos da rede ou diminuir a carga de serviço desejado. Porém no caso dos recursos da rede já estarem determinados, como no caso de uma rede já instalada, cabe saber quanto de carga de serviço é possível de ser suportada pela quantidade de recurso disponível. E para responder a esta dúvida torna-se necessário e importante estabelecer uma técnica que possa ser utilizada de forma fácil e que ofereça apoio à decisão no momento de expansão de serviços em uma rede, permitindo a visualização antecipada do impacto desta expansão.

A adequação dos recursos que compõem a estrutura de uma rede determina o grau de disponibilidade em atender certa quantidade de serviços. As perguntas que desejamos responder com certa precisão são: qual é a quantidade de serviço ou carga de tráfego de aplicações multimídia que uma rede³ ATM e IP com reserva via RSVP pode suportar, tendo em vista os seus parâmetros de desempenho, e qual é a perturbação que causará nas transmissões das aplicações elásticas já existentes na rede, também em relação aos seus parâmetros de desempenho, e que representam, geralmente, o fluxo convencional de uma rede?

A partir de toda esta realidade de mercado, com ampla utilização de aplicações multimídia, pesquisas indicam soluções para permitir um tratamento diferenciado e com qualidade para esses fluxos. Com isto surgiu outra questão que atraiu atenção e tornou-se um grande fator de motivação para o desenvolvimento desse trabalho, que é conhecer o desempenho dos dados gerados pelas aplicações tradicionais ou em outras palavras, mensurar as perturbações que podem ocorrer nos dados tradicionais que trafegam em uma rede de serviços integrados. Além disto, com o emprego cada vez maior de ATM por empresas de telecomunicações, em contrapartida, com a imensa base já instalada e operacional IP, conhecer o limite⁴ da rede é estar ciente da potencialidade de negócios sem comprometimento da qualidade. É poder expandir com qualidade e segurança para todos os serviços, os já utilizados e os novos, satisfazendo os usuários finais que são quem solicitam novas demandas.

¹ Aplicações do tipo FTP, e-mail, WEB, Telnet, etc ...

² Considerado como carga adicional, ou seja, aquele tipo de fluxo ocasional na rede, como por exemplo uma videoconferência.

³ Neste caso particular, o modelo proposto para essa dissertação.

⁴ Limite neste caso representa o ponto de saturação da rede modelada, não da tecnologia utilizada, pois cada caso é um caso, ou seja, cada situação é única e requer uma solução única também.

1.2 Organização do trabalho

Esta dissertação está dividida em 5 capítulos da seguinte forma:

- **Capítulo 1: Introdução**

São apresetados os objetivos e motivações para o desenvolvimento deste trabalho, assim como este trabalho está estruturado.

- **Capítulo 2: Avaliação de desempenho em redes de comunicação de dados**

Apresenta conceitos importantes em relação ao aspecto de desempenho em uma rede, quais os elementos relacionados e de que forma podemos fazer uma análise estatística desses elementos. Traz também um pouco da teoria estatística relevante para essa dissertação.

- **Capítulo 3: Aplicações multimídia em redes de computadores**

A parte densa da teoria estudada em relação aos diversos aspectos que se relacionam ao tema dessa dissertação está nesse capítulo. São apresentados conceitos sobre qualidade de serviço em redes, caracterização de serviços, modelagem de tráfego, implementação de qualidade de serviço e finalmente como as duas tecnologias envolvidas nessa dissertação operam em relação a qualidade de serviço.

- **Capítulo 4: Técnica desenvolvida**

Aqui é abordado o método de simulação para avaliação de desempenho em redes de comunicação de dados, apresentados sucintamente os dois simuladores utilizados nessa dissertação, os cenários que foram modelados e simulados, as métricas de desempenho medidas nos experimentos, os valores iniciais escolhidos e utilizados e os resultados obtidos, mostrados em gráficos, com suas respectivas análises.

- **Capítulo 5: Considerações Finais**

Conclusões e sugestões de melhorias e novos trabalhos que podem ser desenvolvidos a partir dos resultados dessa dissertação.

Possuindo os seguintes anexos:

- **Anexo 1 Relação dos programas alterados no simulador NIST;**
- **Anexo 2 Listagem dos scripts e arquivos das simulações para os simuladores NIST e NS-2;**
- **Anexo 3 Execução dos simuladores.**

2 Avaliação de desempenho em redes de comunicação de dados

A análise de desempenho em redes de computadores tem como objetivo verificar se os níveis de serviço oferecidos aos usuários estão dentro dos requerimentos do projeto da rede, juntamente com a utilização eficiente dos recursos.

Um administrador de rede típico gasta uma grande parte do seu tempo resolvendo problemas e tentando avaliar o desempenho da rede. Quanto melhor compreendê-la menos freqüente serão as falhas, visto que, se evitam problemas quando o desempenho é gerenciado adequadamente [SPO 91].

A análise de desempenho de uma rede pode ser definida como uma investigação quantitativa contínua da rede de comunicação, tendo por objetivos [SPO 91]:

- Verificar se os níveis de serviços são mantidos;
- Identificar e predizer gargalos pontenciais¹;
- Dar suporte à administração e planejamento da capacidade da rede.

Redes de computadores não podem ser gerenciadas quando não é possível medir indicadores significativos de desempenho da rede. Um requisito fundamental é o de que os níveis de serviços especificados no projeto da rede sejam mantidos durante todo o ciclo de vida para a satisfação do usuário [SPO 91].

Os usuários, em geral, percebem os serviços de informática através do tempo de resposta, da disponibilidade do sistema e da facilidade de uso. A última está associada ao software utilizado, enquanto as outras duas estão relacionadas ao dimensionamento de recursos [GON 2001].

Então para podermos responder à pergunta chave deste trabalho que é como estimar com certa precisão qual é a quantidade adicional de serviço ou carga de tráfego de aplicações multimídia que uma determinada rede ATM ou IP com reserva via RSVP pode suportar, sem que isto perturbe a transmissão das aplicações elásticas já existentes na rede e que representam, geralmente, o fluxo convencional de uma rede, temos que conhecer os aspectos relacionados ao dimensionamento de recursos e quais são os indicadores significativos que vão permitir analisar o desempenho de uma dada rede.

2.1 Elementos relacionados em uma rede

O desempenho de um sistema de computação resulta da interação da carga de trabalho com os recursos que compõem o sistema. Uma rede é composta por elementos físicos (*switches*, equipamentos terminais, enlaces de transmissão), cada qual com sua capacidade finita de transmissão de informações. Se a capacidade não é adequada para suportar a carga de serviços demandada pelos usuários, o desempenho da rede em transmitir as informações será prejudicado. Existe uma relação de dependência entre carga, capacidade e desempenho de uma rede (fig. 2.1). A carga é a quantidade de tráfego a ser carregada na rede. A capacidade da rede é dada pelos recursos que a

¹ Pontos de saturação

constituem. E o desempenho representa a qualidade de serviço oferecida pela rede aos usuários. A figura 2.1 ilustra a dependência destes elementos. A capacidade de um sistema de computação é definida como sendo a carga de trabalho que o sistema pode processar, sem ultrapassar os limites de desempenho estabelecidos pelos níveis de serviço da instalação [GON 2001].

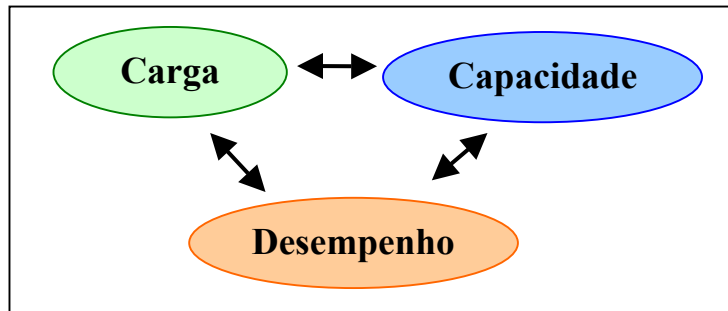


FIGURA 2.1 - Elementos do Relacionamento de Dependência [GON 2001]

A fase de previsão de desempenho deve relacionar o impacto da carga prevista sobre os níveis de serviços e a utilização corrente dos recursos. Podemos fixar um elemento em determinada quantidade para avaliar o comportamento dos outros dois elementos entre si. Por exemplo, fixemos a capacidade (quantidade de recursos) de uma rede de comunicação. Aumentando continuamente a quantidade de carga (tráfego) sobre a rede, atingiremos, em determinado momento, o esgotamento dos recursos ocasionando a redução no desempenho da rede percebido em questões de tempo de resposta, congestionamento e indisponibilidade do serviço (fig 2.2).

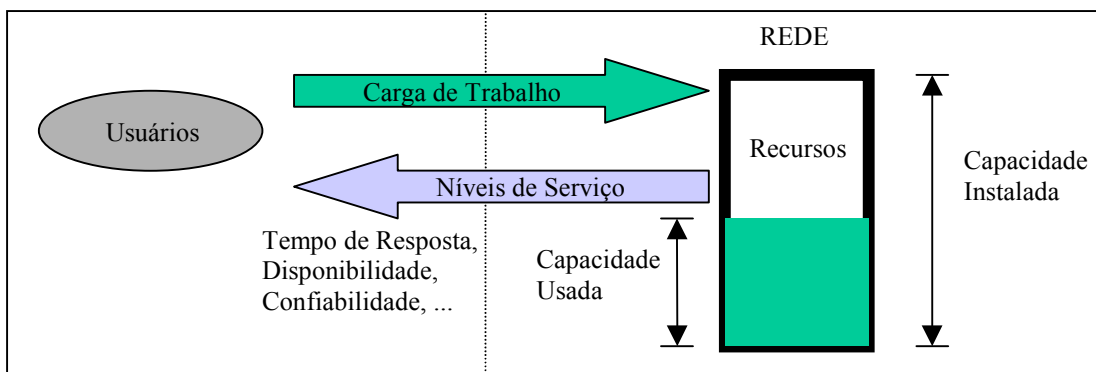


FIGURA 2.2 - Níveis de serviço (QoS), carga e recursos da rede [GON 2001]

O ponto de exaustão do sistema ocorrerá quando, em função das demandas previstas, o nível de serviço for considerado insatisfatório, conforme ilustra a figura 2.3 [GON 2001].

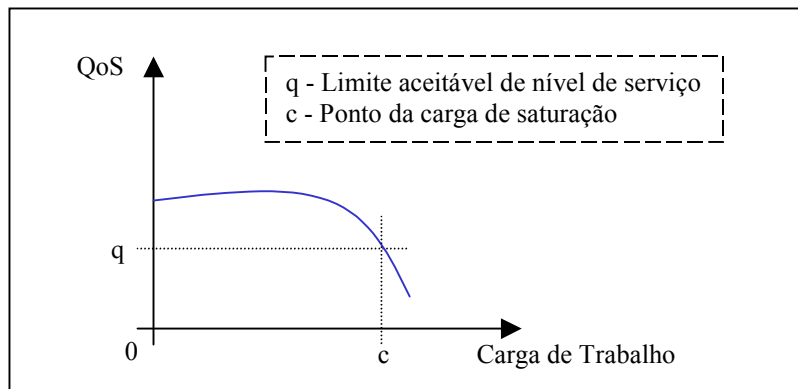


FIGURA 2.3 - Comprometimento dos níveis de serviço pela sobrecarga [GON 2001]

A avaliação de um sistema pode ser a combinação de diversas medidas de desempenho alcançadas durante a realização de suas atividades. Métodos para a avaliação de desempenho estão divididos em duas categorias [GON 2001]:

- de medição: são utilizadas quando é possível fazer medições sobre a rede nas condições de tráfego que se deseja. A técnica de medição requer que redes reais sejam avaliadas por experimentação.
- de previsão: são utilizadas em situações de projeto de rede, quando se deseja avaliar o desempenho da futura rede ou em situações de expansão de tráfego. São feitas a partir de modelos analíticos ou simulados. Desta forma possibilita que os processos de planejamento e projeto sejam executados com confiança, além dos custos de rede poderem ser otimizados. Também, garante-se que são dados aos usuários os níveis de serviço desejados sem degradação do desempenho, causada por uma subestimativa dos recursos necessários; ou custos excessivos, causados por superestimativas.

Nesta dissertação o enfoque é analisar o desempenho de um particular projeto¹ de rede com ênfase na variação do tráfego, medindo o desempenho a partir desse modelo simulado para uma dada capacidade de rede.

Os resultados irão variar de acordo com o modelo simulado, porém a técnica utilizada nessa dissertação será a mesma para a avaliação dos resultados de desempenho para outros modelos.

2.2 Parâmetros Estatísticos de uma Variável Aleatória

Uma vez que os sistemas simulados são em geral compostos de um ou mais elementos que possuem alguma incerteza a eles associados, temos que fazer uso intenso de conceitos de estatística e probabilidade. Estes sistemas evoluem no tempo de um modo imprevisível; são chamados de sistemas estocásticos. A simulação de sistemas

¹ Modelo de simulação

estocásticos requer que a variabilidade dos elementos no sistema seja caracterizada utilizando-se conceitos probabilísticos. As saídas destes sistemas são também probabilísticas e, portanto, interpretações estatísticas são usualmente requeridas [SOA 90].

Em estudos de simulação, inferências ou predições devem ser realizadas baseadas nos resultados experimentais obtidos da simulação. Devido ao fato da simulação possuir elementos aleatórios, as saídas de uma simulação são amostras observadas de variáveis aleatórias que podem ser caracterizadas por parâmetros estatísticos como o seu valor médio; $E[X]$ ou μ_X , (também chamado de espectância, valor esperado ou primeiro momento), o segundo momento; $E[X^2]$, a variância; $Var[X]$ e o desvio padrão; σ_X , onde esses últimos, a variância e o desvio padrão, são uma medida do grau de espalhamento da variável em torno do valor médio, e, em consequência, qualquer assertiva que fizermos quanto à operação do sistema baseada em resultados de simulação, deve levar em consideração a variabilidade inerente destas saídas. Esta variabilidade é quantificada, ou levada em consideração, pelo uso de intervalos de confiança [SOA 90].

Intervalo de confiança, ou intervalo de estimação, é um intervalo onde o parâmetro da população deve se encontrar, com uma certa probabilidade dada, chamada nível de confiança. No desenvolvimento clássico do intervalo de confiança para a média, assume-se que as amostras são independentes e identicamente distribuídas. Pelo teorema do limite central, a média de i amostras terá, aproximadamente, distribuição normal, para grandes valores de i (normalmente $i=10$ a 15 já é suficiente) [SOA 90].

Por exemplo, dado que o valor médio para uma variável aleatória x é 350 e que seu intervalo de confiança calculado é de ± 20 , então podemos dizer com 99% de confiança que o valor médio desta população estará entre 330 e 370.

A figura 2.4 ilustra a distribuição das médias das amostras, o intervalo de confiança e o nível de confiança.

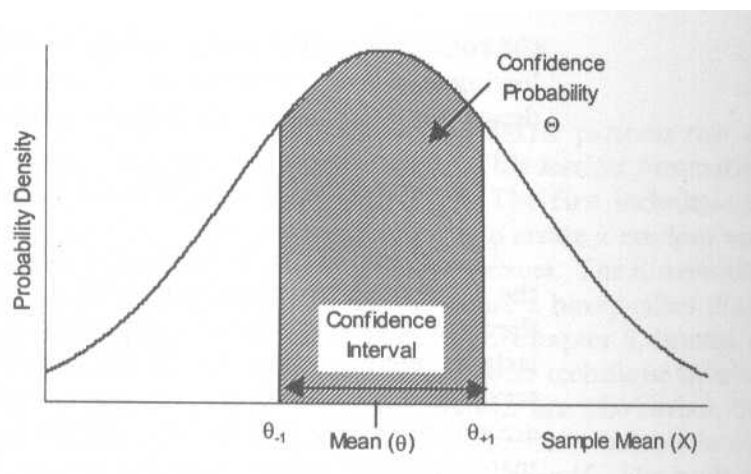


FIGURA 2.4 – Intervalo de confiança e nível de confiança [MCD 2000]

Um parâmetro da população de grande interesse em simulação é a média. Para cada métrica avaliada neste trabalho, será calculado o intervalo de confiança em relação ao valor médio das 10 amostras, com nível de confiança de 99%, ou seja, a cada

simulação executada será calculado e armazenado o valor médio de cada uma das 3 métricas de interesse e ao final de 10 execuções, em cada um dos cenários simulados, será feito a média desses 10 valores médios e para este valor será calculado o valor limite, para cima e para baixo (IC), tendo 99% de confiança de que o valor médio da população estará dentro desses limites.

3 Aplicações multimídia em redes de computadores

O termo multimídia é usado para nos referimos à representação, arquivamento, recuperação e disseminação de informação expressa em múltiplos meios, como texto, voz, vídeo, gráficos, imagem e áudio. Alguns exemplos de aplicação multimídia seriam a teleconferência, propaganda, imagens médicas, etc. [ONV 95].

Em dias passados, a maioria dos dados transmitidos nas redes era do tipo texto. Hoje, com o surgimento da multimídia e de novas tecnologias de rede, os usuários desejam cada vez mais utilizar a Internet para transmitir animação, voz e vídeo clips. Produtos de multimídia como telefonia via Internet, TV via Internet e vídeo conferência têm aparecido no mercado. No futuro, as pessoas desejarão transmitir ainda outros tipos de produtos multimídia, para ensino a distância, simulação distribuída, grupos de trabalho distribuídos e outras áreas [LIU 97].

O surgimento da multimídia sem sombra de dúvida impulsionou o uso do computador como uma ferramenta de comunicação.

Para o pessoal da área de redes de computadores, multimídia em rede significa construir equipamentos e programas de infra-estrutura e ferramentas de aplicação para suportar o transporte na rede, permitindo aos usuários se comunicarem via dados, áudio e vídeo.

As aplicações multimídia diferem das aplicações unimídia de várias formas. Várias aplicações multimídia envolvem um grupo de usuários e exigem conexões ponto-multiponto e multiponto-multiponto. Em geral, elas impõem requisitos rígidos de desempenho temporal na rede. Por empregar diferentes tipos de mídia, a maior parte das aplicações multimídia gera grande quantidade de dados [ONV95].

Assim, transportar multimídia em redes não é uma tarefa trivial. No mínimo três dificuldades existem [LIU 97]:

- Primeiro, comparando com as tradicionais aplicações textuais, as aplicações multimídia em geral requerem muito mais banda passante. Um pedaço típico de filme de 25 segundos de duração, com tamanho de 320x240 consumiria 2.3 Mbytes, que é equivalente a 1000 telas de dados do tipo texto. Isto era inimaginável nos dias passados em que somente texto era transmitido pela rede.
- Segundo, a maioria das aplicações multimídia requer tráfego de tempo real. Dados de áudio e vídeo devem ser executados com uma taxa contínua para a qual eles foram aceitos. Se o dado não chegar a tempo, o processo parará e os ouvidos e olhos do ser humano poderão facilmente detectar essa situação. Em conversa telefônica via Internet, as pessoas toleram uma latência de aproximadamente 250 milissegundos. Se a demora exceder esse limite, a voz ficará muito truncada e o usuário se queixará da qualidade da chamada. Além do atraso, o congestionamento na rede também tem sérios efeitos sobre o tráfego com requisitos temporais. Se a rede estiver congestionada, o único efeito sobre o tráfego textual é que levará mais tempo para completar a transferência, mas os dados de tempo real ficarão obsoletos e deverão ser descartados se não chegarem em tempo. Se não for adotada uma reação apropriada, a retransmissão de pacotes perdidos somente agravará a situação e a sobrecarga na rede.

- Terceiro, um fluxo de dados multimídia é geralmente em rajadas. Apenas aumentar a banda passante não resolverá o problema das rajadas. Para a maioria das aplicações multimídia, o receptor tem um "buffer" limitado. Se nenhuma medida for tomada para suavizar o fluxo de dados, poderá haver "overflow" ou "underflow" do "buffer" da aplicação. Quando os dados chegam muito rápidos, ocorrerá "overflow" do "buffer" e alguns pacotes de dados poderão ser perdidos, resultando em uma má qualidade. Quando os dados chegam muito devagar, ocorrerá "underflow" do "buffer" e a aplicação terá de esperar pelos dados, podendo eventualmente ser cancelada.

Contrariamente ao desejado, na vida real as redes são compartilhadas por milhares e milhões de usuários, possuem largura de banda limitada, atrasos e disponibilidade imprevisíveis. Assim, a solução desses conflitos torna-se um desafio para o qual é necessário o emprego de diversas técnicas que, combinadas, já mostram bons resultados em redes IP, tais como RTP, RTCP e RSVP [LIU 97].

3.1 Qualidade de Serviço em redes

3.1.1 Definição de Qualidade de Serviço

Qualidade de Serviço é um dos tópicos mais confusos e difícil de definir em redes de computadores hoje em dia. O problema é que, em geral, QoS assume significados diferentes para pessoas diferentes. Algumas definições são [KAM 2000]:

- Na visão da ISO: QoS é definida como o efeito coletivo do desempenho de um serviço, o qual determina o grau de satisfação de um usuário de serviço;
- Em um sistema multimídia distribuído a qualidade de serviço pode ser definida como a representação do conjunto de características qualitativas e quantitativas de um sistema multimídia distribuído, necessário para alcançar a funcionalidade da aplicação;
- E em redes de computadores, QoS é utilizado para definir o desempenho de uma rede relativa às necessidades das aplicações, como também o conjunto de tecnologias que possibilita às redes oferecer garantias de desempenho. Em um ambiente compartilhado de rede, QoS necessariamente está relacionado à reserva de recursos. Essa reserva pode ser feita para um conjunto (agregação) de fluxos ou sob demanda para fluxos individuais. QoS pode ser interpretada como um método para oferecer alguma forma de tratamento preferencial para determinada quantidade de tráfego da rede. Em outra definição, QoS é vista como a capacidade de um elemento da rede ter algum nível de garantia de que seus requisitos de serviço de tráfego podem ser satisfeitos.

A junção dos termos qualidade de serviço pode dar margem a várias interpretações e definições diferentes. No entanto, existe um certo consenso, que aparece em praticamente todas as definições de QoS, que é a capacidade de diferenciar entre tráfego e tipos de serviços, para que o usuário possa tratar uma ou mais classes de tráfego diferente das demais. Com relação ao tipo de tráfego, as aplicações podem ser classificadas em [KAM 2000]:

- Aplicações de Tempo Real (não elásticas): podem ser definidas como

aquelas com características rígidas de reprodução, ou seja, um fluxo de dados é empacotado na fonte e transportado através da rede ao seu destino, onde é desempacotado e reproduzido pela aplicação receptora. Essa classificação pode ainda ser quebrada em duas categorias:

- Aplicações Tolerantes: são aquelas que mesmo diante de variações no atraso (jitter) causadas pela rede, ainda assim produzem um sinal de qualidade quando reproduzidas.
- Aplicações Intolerantes: são aquelas que diante de variações no atraso produzem sinais com qualidade inaceitável, como no caso de sinais de áudio e vídeo.
- Aplicações Elásticas (não tempo real ou adaptáveis): para esse tipo de aplicação, a recepção correta dos dados é mais importante do que a sua apresentação em uma taxa constante. Exemplos são: correio eletrônico, transferência de arquivos, consultas interativas a informações, como na Web, e aplicações cliente/servidor tradicionais.

A abordagem de Qualidade de Serviço que é interesse desta dissertação diz respeito à garantia de transmissão de aplicações de tempo real sem que isto perturbe a transmissão de aplicações elásticas, que é o tipo de fluxo mais comum em uma rede. Em outras palavras, a rede deve permitir o compartilhamento de recursos por tantos serviços, que representam a sua carga de tráfego, enquanto for possível manter a qualidade de serviço, baseado nos valores das métricas de interesse pelo tipo da aplicação, para cada aplicação que esteja transmitindo, independente da natureza destes fluxos, quer sejam de tempo real ou não.

A garantia de transmissão pode ser expressa como a combinação de algumas das seguintes métricas¹ [KAM 2000]:

- Atraso: é o tempo necessário para um pacote percorrer a rede, medido do momento em que é transmitido pelo emissor até ser recebido pelo receptor.
- Variação do Atraso ou Jitter: é a variação do atraso fim-a-fim.
- Largura de banda: é a taxa de transmissão de dados máxima que pode ser sustentada entre dois pontos. Geralmente é vista como uma característica do enlace físico. O termo vazão é utilizado para designar a taxa máxima que alguma aplicação ou protocolo consegue manter em uma rede.
- Perda: relacionado com a perda de informações (pacotes/células) pela rede.

As métricas são usadas para caracterizar e descrever o comportamento da rede no que diz respeito a sua utilização e performance. Devido ao significado e definição próprio de cada métrica, a seleção das métricas mais adequadas para um estudo de performance em redes depende de vários fatores, tais como [SCH 99]:

- os objetivos específicos do estudo de performance;
- características da topologia que é objeto de estudo;
- os protocolos e serviços operados nesta topologia.

¹ Exceto quando estiver sendo tratado especificamente de um tópico das redes ATM, ao longo do texto desse trabalho, poderão aparecer as siglas CTD, CDV e CLR, sendo utilizados como atraso, jitter e perda respectivamente.

Alguns dos indicadores mais significativos para o estudo de performance em redes com aplicações multimídia estão listados na tabela 3.1.

TABELA 3.1 - Principais métricas de performance

Métrica - unidade	Descrição
Retardo/Atraso/Latência - ms	Tempo consumido pela rede para transportar um pacote/célula do transmissor ao receptor.
"Jitter"/Variação do Atraso - ms	Variação máxima do retardo entre pacotes/células de um fluxo. Se mínimo atraso é 1ms e máximo é 6ms, então jitter é 5ms.
Vazão - taxa de bits por segundo	Taxa de informação que chega e que é entregue por um nodo da rede por unidade de tempo.
Taxa de Perdas - %	Pacotes/células perdidos(as) em relação ao total de pacotes/células enviados(as).
Taxa de Erros - %	número de bits com erro em relação ao número total de bits enviados.

Fonte: [ARA 94]

Dois fatores influenciam e dificultam a garantia de bons resultados para essas métricas: os atrasos variáveis de enfileiramento e os congestionamentos [STA 2000].

A figura 3.1 apresenta a necessidade de QoS, baseado em duas métricas para algumas aplicações bem comuns em uma rede. Algumas aplicações podem tolerar um número moderado de perda, enquanto outras exigem perdas compatíveis com as disponíveis em canais dedicados. O objetivo da rede é prover os recursos necessários de forma que a integridade de cada aplicação seja mantida.

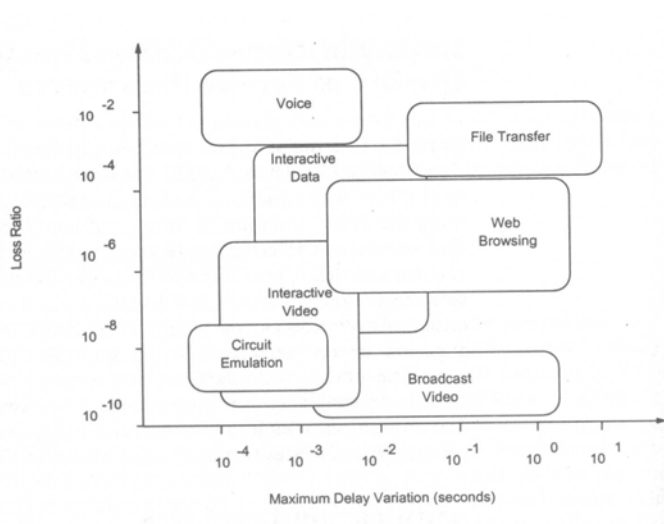


FIGURA 3.1 - Necessidades de QoS para diferentes aplicações [MCD 2000]

3.1.2 Modelo de referência

Um modelo de referência de QoS fim a fim usualmente contém uma ou mais redes entre os extremos, cada uma potencialmente com múltiplos nodos, como representado na figura 3.2. Cada uma dessas redes intermediárias podem introduzir atrasos, perdas e erros próprios da multiplexação, comutação ou transmissão, impactando no QoS.

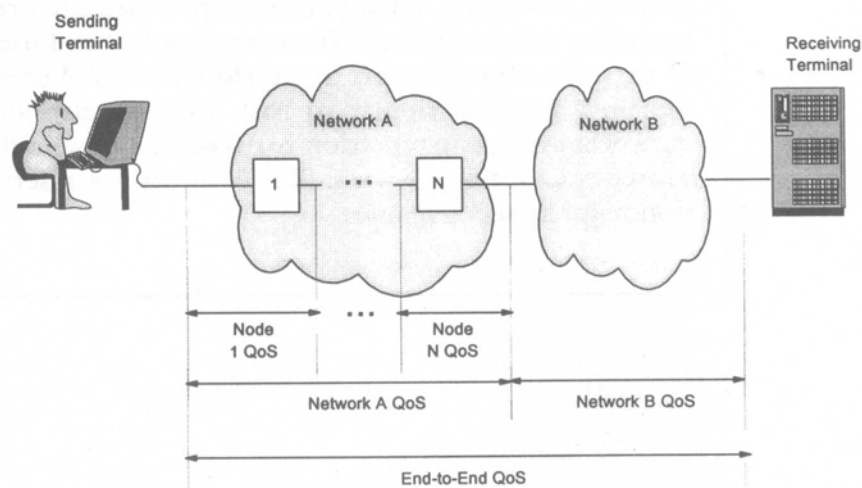


FIGURA 3.2 – Qualidade de Serviço Fim-a-Fim [MCD 2000]

Os principais parâmetros de QoS afetados por roteadores IP e switches ATM são atraso, variação do atraso e perda causada por excesso nas filas. Quanto mais nodos a informação atravessar, mais a qualidade degradará. Todos os parâmetros de QoS comportam-se aproximadamente de maneira linear, exceto a variação do atraso, a qual cresce no mínimo na razão da raiz quadrada do número de nodos atravessados[MCD 2000].

3.2 Gerações de redes

3.2.1 Redes sem qualidade de serviço: IP

As redes IP têm problemas graves quando se trata de passar mídias sensíveis ao atraso, como voz e vídeo. Isto ocorre porque o protocolo IP presta um serviço do tipo datagrama (sem conexão), sem alocar um caminho fixo para um pacote de dados trafegar na rede, o que não garante requisitos temporais para os pacotes transmitidos.

Nas redes de comutação de circuitos, tradicionalmente utilizadas para transportar voz, é estabelecido um caminho fixo ao se efetuar a conexão entre as entidades comunicantes. Além disto, após ser feita a alocação de banda e dos demais recursos da rede, esta fica dedicada somente àquela conexão, sem compartilhamento. Se Ganha, portanto, uma garantia de Qualidade de Serviço (*QoS - Quality of Service*), que

segundo o ITU-T é definido como “o efeito coletivo de desempenho que determina o grau de satisfação do usuário deste serviço específico”. Entretanto, perde-se em eficiência no uso dos recursos da rede, pois mesmo que não se transmita nada na banda alocada, esse recurso está reservado e não se pode compartilhá-lo, o que acarreta um desperdício de banda passante.

Já nas redes de comutação de pacotes os recursos são compartilhados entre diversos usuários que desejem transmitir. Além disto, estas redes não alocam um caminho dedicado a uma conexão. Com isto, pacotes diferentes de um mesmo usuário podem ser transmitidos por rotas diferentes, perdendo-se a ordenação dos pacotes transmitidos. Nesse tipo de rede, também é significativa a necessidade de processamento por parte dos nós intermediários de uma conexão.

Outro desafio é como os pacotes são tratados em uma rede do tipo IP como a Internet, já que as transmissões multimídia exigem cadência e são sensíveis ao atraso. Numa rede IP, o serviço prestado é do tipo "melhor esforço" (*best effort*) significando assim que todos os pacotes são tratados de forma igual, sem nenhuma discriminação entre os diversos tipos de tráfego, nem atribuição de prioridades aos pacotes. Além disso, o escalonamento é feito baseado em filas FIFO (*First In First Out*): se houver espaço nos buffers dos roteadores, o pacote é armazenado para transmissão; caso contrário, ele é descartado.

Um fator bastante crítico para a transmissão de voz e vídeo é o problema dos atrasos. Esses podem ser agrupados em dois tipos: os fixos e os variáveis. Os dois tipos possuem naturezas e causas diferentes. Os atrasos fixos causam desconforto na conversação ou visualização e os variáveis atrapalham a cadência da transmissão.

Os atrasos fixos são ocasionados por diversos fatores, conforme abaixo:

- Compressão e Descompressão
- Processamento interno dos roteadores/switches
- Propagação/Transmissão: dependente do meio;

Os atrasos variáveis são decorrentes do tráfego e do congestionamento da rede. Esses são causados principalmente pelo enfileiramento dos pacotes nos roteadores/switches. A combinação destes efeitos pode ser verificada na situação apresentada na figura 3.3. Vale lembrar que atrasos da ordem de 250ms (para alguns, 150ms) são considerados intoleráveis para transmissão de voz, pois causam perda de interatividade. Valores mais altos do que isso, porém, podem ser atingidos, em algumas situações, na Internet.

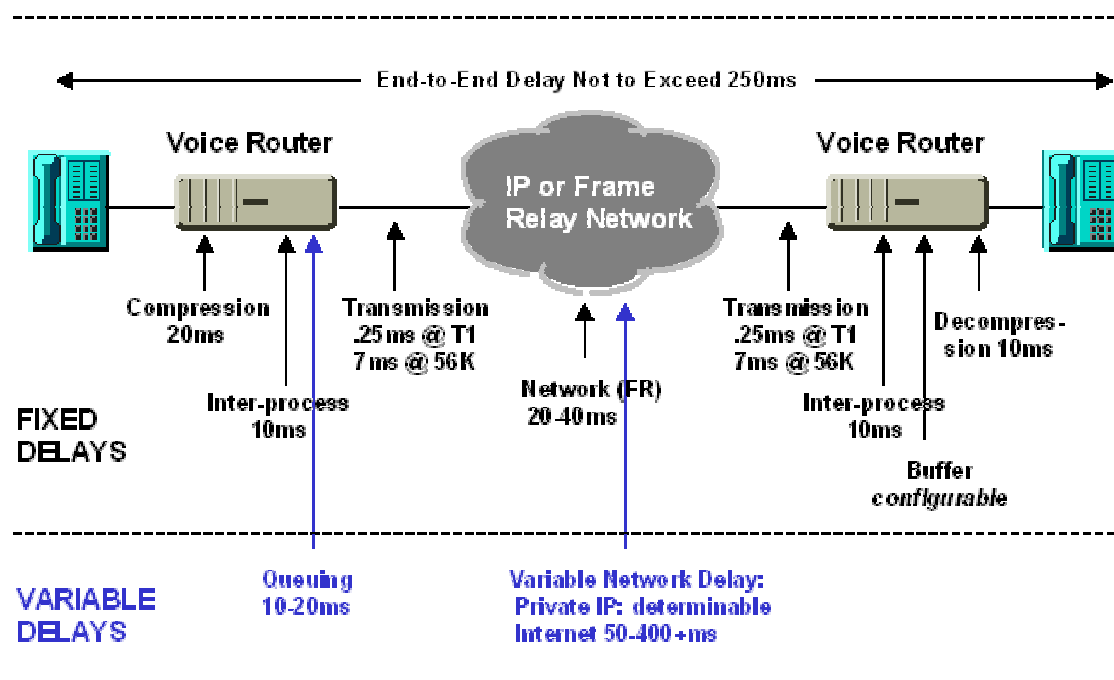


FIGURA 3.3 - Fontes de atraso em transmissão de voz em redes de pacotes [DEL 99]

Além dos atrasos, é fundamental considerar as perdas existentes na rede. Para o tráfego sem compressão, elas não são tão importantes. Contudo, ao se utilizar compressão aumenta-se a sensibilidade em relação às perdas. Apesar de o protocolo TCP tentar garantir a recuperação contra congestionamento e perdas, uma garantia maior só é obtida através de uma banda maior disponível, bem como de uma melhoria no tempo de processamento dos nós.

Por fim, pode-se mencionar também como mais uma dificuldade o uso do protocolo UDP como transporte para aplicações de multimídia. Este protocolo, que não efetua a reordenação nem a recuperação por retransmissão, é o mais adequado para se manter a cadência da transmissão. Além do mais, possui alguma tolerância a perdas quando se trata da transmissão de voz. Contudo, o UDP é do tipo datagrama e não possui controle de congestionamento algum. Por isso, ele pode ser um emissor agressivo para a rede, gerando, assim, congestionamento. Serão mostradas a seguir as tecnologias que estão em desenvolvimento, ou que já estão estabelecidas, que permitem contornar estes obstáculos.

3.2.1.1 Soluções possíveis

Alguns mecanismos foram desenvolvidos para permitir a utilização de redes de pacotes como meio de transporte para aplicações multimídia.

O primeiro deles é um conjunto de protocolos que procuram acomodar as aplicações sensíveis ao tempo, como voz e vídeo “mascarando” suas limitações.

Também são utilizados os mecanismos de suavização e priorização de tráfego,

que atuam na saída dos dados para rede, assim como outras abordagens como Engenharia de tráfego (Traffic Engineering - TE), MPLS, Diff Serv, QoS [KAM 2000].

Protocolos

A solução para multimídia sobre IP é classificar todo o tráfego, alocar prioridades para diferentes aplicações e fazer reservas de recursos. O grupo de trabalho dos Serviços Integrados no IETF desenvolveu um modelo de serviço chamado Serviços Integrados, que inclui serviços do tipo best-effort e real-time [BRA 94]. O serviço real-time irá permitir que redes IP ofereçam qualidade de serviço para aplicações multimídia. Os protocolos *RSVP* (Resource Reservation Protocol), *RTSP* (Real-Time Streaming Protocol) [LIU 97], *RTP* (Real-time Transport Protocol) [BUS 96] [LIU 97], *RTCP* (Real-Time Control Protocol) [BUS 96] [LIU 97], ST-II (Stream Protocol) ou IPv5 [EXP 90], H.323 e SIP (Session Initiation Protocol) [ARO 99] oferecem uma base de trabalho para serviços real-time. Os Serviços Integrados permitem às aplicações configurar e gerenciar uma única infra-estrutura para aplicações multimídia e para aplicações tradicionais, oferecendo o tipo de serviço e a qualidade que cada uma delas necessita.

Como nessa dissertação o protocolo de interesse é o RSVP, somente destacaremos as suas características. Em caso de haver interesse nas demais soluções, essas poderão ser mais investigadas nas referências citadas.

RSVP [BER 99] [BRA 94] [BRA 97] [LIU 97]

O RSVP é o protocolo de controle de rede que permite ao destinatário dos dados requisitar um serviço de qualidade especial fim-a-fim para seu fluxo de dados. Aplicações de tempo real usam RSVP para reservar os recursos necessários nos roteadores ao longo dos caminhos de transmissão, de tal forma que a banda solicitada possa estar disponível quando a transmissão realmente inicia. RSVP é o principal componente dos Serviços Integrados Internet que podem oferecer tanto serviços best-effort como serviços real-time.

O RSVP é usado para reservar recursos de rede. Quando uma aplicação num host (o receptor do fluxo de dados) requer uma qualidade de serviço específica (QoS) para aquele fluxo de dados, ele usa o RSVP para despachar seu pedido aos roteadores ao longo do caminho do fluxo de dados. O RSVP é responsável pela negociação dos parâmetros de conexão com esses roteadores. Quando a reserva é feita, o RSVP é também responsável por manter o "status" de roteadores e hosts para que ofereçam o serviço solicitado.

A figura 3.4 ilustra sua operação.

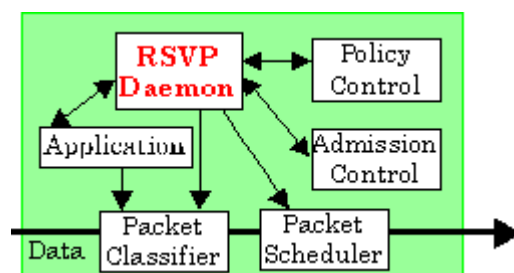


FIGURA 3.4 - Operação do protocolo RSVP [BER 99]

Cada nó capaz de reservar recursos possui diversos procedimentos locais para estabelecer e reforçar a reserva. A *política de controle* determina se o usuário tem permissão administrativa para fazer as reservas. Já o *controle de admissão* mantém um acompanhamento dos recursos do sistema e determina se o nó possui recursos suficientes para satisfazer a QoS solicitada.

O "daemon" RSVP verifica ambas as "procedures". Se qualquer verificação falhar, o programa RSVP retorna uma notificação de erro para a aplicação que originou o pedido. Se ambas as verificações forem bem sucedidas, o "daemon" RSVP estabelece os parâmetros no classificador de pacotes e no "scheduler" ou "dispatcher" de pacotes para obter a QoS solicitada. O classificador de pacotes determina a classe de serviço para cada pacote e o "scheduler" ou "dispatcher" ordena a transmissão de pacotes para obter a qualidade de serviço prometida para cada fluxo de dados.

O "daemon" RSVP também se comunica com o processo roteador para determinar o caminho por onde deve enviar os pedidos de reserva e para manusear as rotas e as trocas de componentes.

Esse procedimento de reserva é repetido nos roteadores ao longo do caminho reverso, até que a reserva se mescle com outra reserva para a mesma fonte de dados.

As reservas são implementadas através de dois tipos de mensagem RSVP: PATH e RESV, como aparece na figura 3.5. As mensagens do tipo PATH são enviadas periodicamente do remetente para o destinatário. Uma mensagem PATH contém o parâmetro "*flow spec*" que descreve o remetente (formato dos dados, endereço de origem e porta de origem) e as características de tráfego. Essa informação é usada pelo destinatário para descobrir o caminho reverso até o remetente e determinar que recursos devem ser reservados.

As mensagens RESV são geradas pelos destinatários e contêm parâmetros de reserva que incluem "*flow spec*" e "*filter spec*". O "*filter spec*" define que pacotes no fluxo devem ser usados pelo classificador de pacotes. O "*flow spec*" é usado pelo escalonador e seu conteúdo depende do serviço. Mensagens RESV seguem exatamente o caminho reverso das mensagens PATH, fazendo reservas para um ou mais remetentes em cada nó.

Os estados de reserva que o RSVP cria nos roteadores são chamados "*soft states*". O "daemon" RSVP necessita enviar periodicamente mensagens do tipo "refresh", para manter os estados de reserva, como mostra a figura 3.5. A ausência de mensagens "refresh" dentro de um certo período de tempo irá destruir o estado de reserva. Usando "soft states", o RSVP pode facilmente manusear a mudança de componentes da rede e de rotas.

Os pedidos de reserva são iniciados pelos destinatários, como mostra a figura 3.5. Eles não necessitam trafegar ao longo de todo o caminho entre a fonte e o destino. Ao invés disso, eles trafegam no caminho reverso até encontrar outro pedido de reserva para a mesma fonte de dados, então se misturam com essa outra reserva.

Essa mescla de reservas conduz à primeira vantagem do RSVP: a escalabilidade. Um grande número de usuários pode ser incluído num grupo "multicast" sem aumentar significativamente o tráfego de dados. Assim, o RSVP pode atingir a escala de grandes grupos "multicast" e o "overhead" médio do protocolo diminui conforme aumenta o número de participantes.

O processo de reserva não transmite realmente os dados nem oferece a qualidade de serviço solicitada. Através do processo de reserva, o RSVP apenas garante que os recursos de rede estarão disponíveis quando a transmissão efetivamente iniciar.

Embora o RSVP se situe no topo do conjunto de protocolos IP, ele não é um protocolo de roteamento, mas sim um protocolo de controle. Na verdade, o RSVP confia nos protocolos de roteamento das camadas inferiores para descobrir para onde poderá emitir os pedidos de reserva. O RSVP também deve cooperar com protocolos de roteamento "unicast" e "multicast". Quando um fluxo RSVP tem seu caminho alterado, o módulo de roteamento notifica o módulo RSVP sobre a mudança da rota. Assim, o RSVP pode ajustar rapidamente as reservas de recursos para as novas rotas.

A remessa de parâmetros de reserva é diferente a partir da determinação desses parâmetros. Como a definição dos parâmetros de conexão para obter a qualidade de serviço solicitada é tarefa dos dispositivos de controle de QoS, o papel do RSVP é apenas facilitar a distribuição desses parâmetros. Uma vez que diferentes aplicações podem ter diferentes dispositivos de controle de QoS, o RSVP é desenhado para tratar esses parâmetros QoS como dados numa caixa preta, que devem ser remetidos e interpretados pelos módulos de controle dos roteadores. Essa separação lógica dos dispositivos de controle de QoS e das facilidades de distribuição simplifica o projeto do RSVP e o torna mais adaptativo para novas tecnologias de rede e aplicações.

São características do RSVP:

- fluxo de dados RSVP é "simplex": o protocolo distingue remetentes e destinatários. Embora em muitos casos um host possa atuar como remetente e também como destinatário, o RSVP apenas reserva recursos para fluxos de dados numa única direção.
- RSVP suporta tanto "multicast" como "unicast" e adapta-se às mudanças de rotas e componentes de grupos. Uma vez que a reserva é iniciada pelo destinatário, e o estado de reserva é "soft", o RSVP pode facilmente manusear as trocas de componentes do grupo e de rotas. Um host pode enviar mensagens IGMP (Internet Group Management Protocol) para unir-se a um grupo "multicast". Mesclar reservas permite que o RSVP seja usado para grandes grupos de "multicast" sem causar uma sobrecarga pesada para o remetente.
- RSVP é orientado ao destinatário e pode trabalhar com destinatários heterogêneos. Em grupos "multicast" heterogêneos, os destinatários possuem diferentes características e níveis de QoS. O fato de a reserva ser orientada pelo destinatário facilita o trabalho com grupos "multicast" heterogêneos. Os destinatários são responsáveis pela escolha de seu próprio nível de QoS, iniciando a reserva e mantendo-a ativa enquanto desejar. O remetente divide

o tráfego em diversos fluxos, sendo cada um deles um fluxo RSVP separado com diferentes níveis de QoS. Cada fluxo RSVP é homogêneo e os destinatários podem escolher entre unir-se a um ou mais fluxos. Isso torna possível que destinatários heterogêneos solicitem diferentes QoS, desenhados para suas características particulares.

- RSVP tem boa compatibilidade - Estão sendo feitos esforços para rodá-lo tanto em IPv4 como em IPv6. Ele oferece transporte transparente do controle de tráfego e políticas de controle de mensagem, de forma a torná-lo mais adaptativo a novas tecnologias. Ele também oferece operação transparente através de regiões que não suportam o RSVP.

O RSVP se comunica tanto com aplicações nas pontas finais como com vários componentes dentro dos roteadores.

Na figura 3.5 está representado um exemplo de reserva RSVP para um fluxo “unicast”.

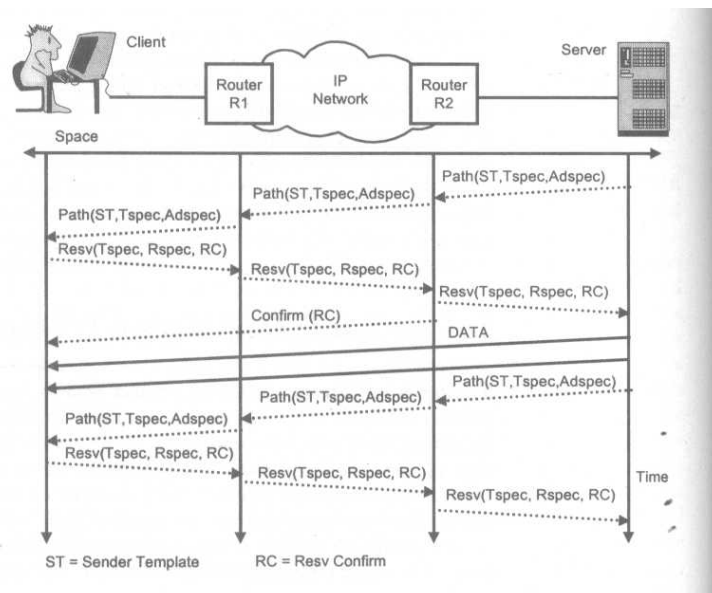


FIGURA 3.5 - Reserva RSVP para um fluxo “unicast” [MCD 2000]

3.2.2 Redes com qualidade de serviço: ATM

O Modo de Transferência Assíncrono é uma tecnologia baseada em transmissão de pequenas unidades de informação de tamanho fixo e formato padronizado, denominadas células. Células são transmitidas através de conexões com circuitos virtuais, sendo seu encaminhamento baseado em informação de um cabeçalho contido em cada uma delas. Tal tecnologia é capaz de suportar diferentes serviços, para satisfazer aos requisitos exigidos pelos diferentes tipos de tráfegos, a altas velocidades de transmissão. Por esta razão, ATM foi a tecnologia escolhida para suportar a diversidade de serviços definida para a Rede Digital de Serviços Integrados de Faixa Larga (RDSI-FL) [SOA 95].

O ATM tem como vantagem o ganho estatístico de serviços com tráfego de taxa variável. Adicionalmente, as características de tamanho fixo e reduzido das células

trazem algumas vantagens quando comparadas a unidades maiores e/ou de tamanho variável. Unidades de informação de tamanho variável aumentam a complexidade dos comutadores da rede. O tempo de empacotamento é um dos fatores a favor de células pequenas. Quanto maior o tamanho da célula, maior o tempo de empacotamento, causando maior atraso de transferência, afetando, principalmente, sinais de áudio e vídeo que, conforme já mencionado, se ultrapassarem um determinado tempo de atraso, deterioram aplicações com interatividade em tempo real entre usuários, como por exemplo, numa conversação telefônica. O único argumento em contrário à utilização de pequenas células é o grande overhead que o cabeçalho representa, o que diminui a capacidade efetiva de transmissão na rede. Contudo, como os argumentos em prol das células pequenas são mais contundentes, de forma que o ITU-T padronizou as células ATM com 53 octetos, dos quais 48 são de informação e 5 de cabeçalho [SOA 95].

Embora as redes ATM tenham sido concebidas para altas velocidades e sejam tanto orientadas à comutação de pacotes como orientadas à conexão, o que é extremamente interessante para transmissão de multimídia, ainda existem muitas questões a serem resolvidas antes que elas comecem a ser usadas intensamente. Faltam mecanismos eficientes de controle de congestionamento e de admissão de chamadas, bem como protocolos de transporte que apresentem menores overheads. Para desenvolver tais mecanismos, todavia, é necessário compreender bem o tráfego que a rede deverá suportar. Tal tráfego se origina de uma grande variedade de aplicações, cada uma delas gerando saídas com características diferentes [ONV 95].

Atualmente, excetuando-se o tráfego de voz, cujas características foram bem estudadas ao longo dos anos, os demais tipos de tráfego ainda são pouco compreendidos. Mesmo para o tráfego de dados, que tem sido intensamente utilizado, ainda existe pouco conhecimento sobre o comportamento das fontes. Essa falta de compreensão das características do tráfego dificulta a escolha de uma taxa de transmissão que atenda à QoS necessária, e que ao mesmo tempo maximize a utilização da capacidade da rede. Assim, é com grande interesse que surgem estudos sobre modelagem de tráfego e de técnicas que possam suportar de forma eficiente fontes imprevisíveis. A caracterização da fonte, numa visão macro, é obtida pela definição das características do tráfego gerado pela fonte e dos requisitos de qualidade de serviço (QoS) [ONV 95].

As características do tráfego devem ser definidas por um conjunto mínimo de parâmetros declarados pelos usuários e que sejam, todavia, suficientes para que a gerência de rede consiga controlar eficientemente o tráfego e obter a máxima utilização da rede[GIR 99].

Os parâmetros a serem declarados variam, conforme se esteja tratando de tráfego constante (CBR) ou variável (VBR). No primeiro caso (CBR), apenas o conhecimento da taxa máxima de transmissão é suficiente. Já no segundo caso (VBR) um conjunto maior de parâmetros é necessário, entre eles o número médio de células geradas num período ativo, a taxa de pico de geração de células, a taxa média de geração de células etc. [MCD 2000].

Já os requisitos de QoS são mais difíceis de serem definidos, pois eles dependem da visão do usuário. Especialmente em serviços de multimídia, definir QoS é uma tarefa complicada, pois envolve vários tipos de usuários, vários tipos de serviços oferecidos e ainda depende de uma visão subjetiva dos usuários sobre tais serviços[MCD 2000].

Além das dificuldades acima citadas, existem outros desafios quando se deseja utilizar os serviços multimídia em redes ATM [ONV 95]:

- estabelecimento de múltiplas chamadas
- controle em tempo real de serviços integrados, como vídeo, áudio e voz
- sincronização entre grupos de usuários
- alocação dinâmica de recursos da rede, levando em consideração as diferentes necessidades de cada aplicação integrada ao serviço
- transferência eficiente multiponto
- sincronização de diferentes aplicações, como entre vídeo e áudio.

O gerenciamento da comunicação entre grupos de usuários é muito mais complexo do que entre conexões ponto-a-ponto. O estabelecimento de chamadas entre mais do que dois usuários pode ser feito através de uma série de conexões ponto-a-ponto ou através de uma conexão multipartite. A última opção é preferível, pois resulta numa conexão mais rápida, mas é mais complicada de desenhar. Várias soluções têm sido propostas, nos últimos anos, para conexões ponto-a-ponto, e estendê-las para permitir a comunicação de mais de duas entidades não é uma tarefa simples. A maioria dos fabricantes de equipamentos para ATM oferece capacidade de multicasting limitada. O relacionamento temporal entre várias aplicações, requerido pelos serviços multimídia, deve ser controlado pela rede. Em redes assíncronas, sincronizar os diferentes tipos de aplicação uns com os outros é bastante difícil, devido aos atrasos randômicos que podem surgir na rede [ONV 95]. Para tanto, torna-se necessário compreender os modelos de tráfego das várias aplicações que integram os serviços multimídia, e a forma como eles se relacionam num ambiente integrado, o que será feito com o auxílio de métricas de QoS.

Na figura 3.6 está representado um exemplo de estabelecimento de conexão ponto-a-ponto ATM. No lado esquerdo está representado o cliente, quem faz a chamada. No lado direito está representado o servidor, quem recebe a chamada. Começando do lado superior esquerdo da figura, o cliente inicia a tentativa de conexão usando a mensagem SETUP. O elemento da rede responde ao solicitante da mensagem SETUP com a mensagem CALL PROCEEDING, como mostra a figura. A rede roteia a tentativa de conexão através de quaisquer switches intermediários usando a mensagem SETUP, reservando a capacidade requerida e outros recursos. O usuário chamado, o servidor, responde à chamada com uma mensagem CONNECT, indicando a localização designada pelo rótulo VPI/VCI no identificador da conexão. A rede propaga a mensagem de volta ao originador o mais rápido possível a fim de minimizar o tempo de estabelecimento da conexão. Os usuários e os elementos da rede confirmam a mensagem CONNECT usando a mensagem CONNECT ACKNOWLEDGE como mostra a figura 3.6.

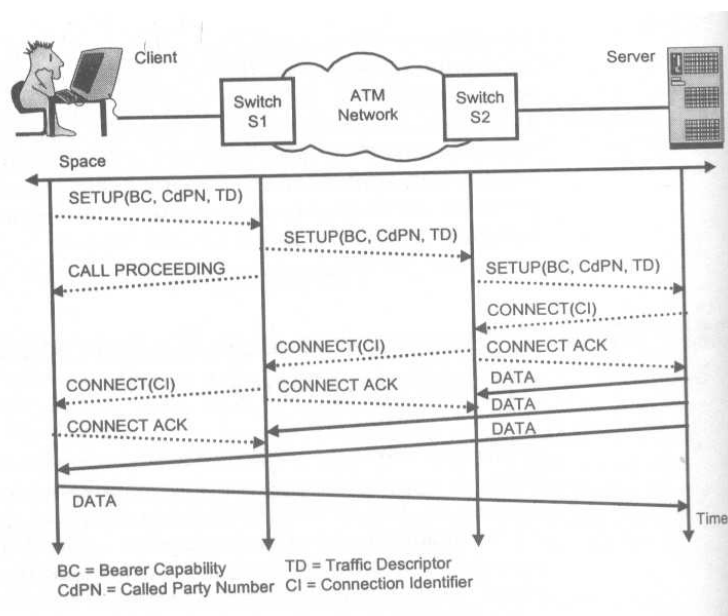


FIGURA 3.6 – Estabelecimento de conexão ponto-a-ponto ATM [MCD 2000]

3.3 Caracterização de Serviços em ATM e IP

Uma rede multi-serviços suporta uma diversidade enorme de serviços. Apesar da diversidade, as conexões desses serviços podem ser agrupadas dentro de categorias conforme suas características de tráfego e expectativas de QoS. Dessa forma a rede pode implementar procedimentos de alocação de recursos e funções de controle de tráfego por categoria de serviço [GON 2001].

Numa rede IP tradicional, um único modelo de serviço é oferecido, chamado de melhor esforço ou "best effort", significando, assim, que todos os pacotes são tratados de forma igual, sem nenhuma discriminação entre os diversos tipos de tráfego, nem atribuição de prioridades aos pacotes, apresentando um desempenho razoável para aplicações elásticas, como transferência de arquivos e mensagens de correio eletrônico. Além disso, o escalonamento é feito baseado em filas FIFO: se houver espaço nos "buffers" dos roteadores, o pacote é armazenado para transmissão; caso contrário, ele é descartado.

Entretanto, à medida que avançamos em direção à era das comunicações multimídia, estão sendo desenvolvidas novas aplicações de tempo real com uma grande sensibilidade ao atraso da rede. Para essas aplicações o modelo de melhor esforço é totalmente inadequado, mesmo em redes com cargas leves. Existe uma necessidade de garantias mais rígidas em termos de largura de banda, atraso e perda de pacotes [KAM 2000].

Nesse contexto, a IETF (Internet Engineering Task Force) criou o grupo de trabalho IntServ (Integrated Services) para viabilizar o surgimento de uma rede de serviços integrados. O termo serviços integrados é empregado para designar um modelo de serviço para a Internet que inclui o serviço de melhor esforço, serviços de tempo real e serviços de compartilhamento controlado do enlace. Esse último está relacionado com o oferecimento de algumas classes de serviços, como o serviço de QoS garantido e o

serviço de carga controlada, que é o utilizado nessa dissertação através da implementação documentada em [GRE 2001].

Numa rede ATM, as características de tráfego de uma aplicação são o conjunto mínimo de parâmetros que se pode esperar que um usuário declare possibilitando o efetivo controle da rede e a obtenção de altas percentuais de utilização dos recursos [SIL 2000]. O ATM Forum¹ já padronizou um número de categorias de serviços baseadas nos seus requerimentos de recurso da rede [GIR 99]. Algumas dessas categorias requerem requisitos de qualidade sensíveis ao atraso com taxas variáveis de transmissão, enquanto outras não. Os parâmetros de QoS e de tráfego a serem declarados pelas aplicações dependem de sua categoria de serviço [GON 2001].

A figura 3.7 mostra os requisitos de QoS e o mecanismo de controle de fluxo oferecido para cada uma das categorias de serviços padronizadas pelo ATM Forum.

	CBR	VBR-rt	VBR-nrt	ABR	UBR
CLR	Especificado				Não Esp.
CTD	Especificado			Não Especificado	
CDV	Especificado		Não Especificado		
Controle Fluxo	Não			Sim	Não
Aplicação	Voz	Vídeo	-	Dados	-

FIGURA 3.7 - Qualidade oferecida pela Rede ATM [GON 2001]

Para uma dada categoria de serviço, uma rede pode oferecer uma ou mais classes de serviço. Uma classe de serviço pode especificar valores de QoS e limitar o intervalo de alguns parâmetros descritores de tráfego. Por exemplo, uma rede deve oferecer duas classes para o serviço CBR, o serviço premium com CLR igual a 10^{-10} e um serviço normal com CLR igual a 10^{-7} . A classe com menor qualidade deve requerer menos alocação de banda. Dessa forma é possível oferecer diferentes níveis de serviços e tarifas. Quanto maior a qualidade, maior o valor a pagar pelo serviço [GIR 99].

3.3.1 Categorias de Serviços ATM

3.3.1.1 Serviços CBR

É a categoria de serviço utilizada por aplicações que geram tráfego a uma taxa constante e podem ser simplesmente descritos pelas suas taxas de pico (PCR), pois requerem uma quantidade fixa (constante) de largura de banda. Nesse caso, todas as células estão sujeitas ao mesmo atraso. A fonte de tráfego pode emitir células abaixo da

¹ * Maiores detalhes em Traffic Management Specification, version 4.0, af-tm-0056.000, April 1996. [TM4.0].

taxa de PCR, porém a largura de banda não utilizada não poderá ser aproveitada por outra conexão, o que acarretará desperdício de recursos da rede. A variação de atraso é praticamente zero. Desde que a fonte não ultrapasse o valor de PCR, o QoS é garantido. O controle de congestionamento é alcançado geralmente por meio do CAC na fase de estabelecimento da conexão. A rede aceita a requisição de chamada somente se puder prover a banda solicitada. Perda de células, ou grandes atrasos, tem um impacto negativo significativo nesse tipo de serviço. Exemplos típicos de serviços CBR incluem voz, vídeo e áudio [GIR 99] [ONV 95].

Voz CBR em redes ATM é transmitido com AAL1 usando a técnica PCM (Pulse Code Modulation). PCM é um método para converter um sinal analógico em um sinal digital. A recomendação G.711 especifica 64 Kbit/s para voz CBR [ONV 95].

A taxa na qual um fluxo de bits de vídeo é submetido à rede deve ser determinada pela taxa de bit alocada na rede. A alocação de CBR para vídeo pode causar degradação de qualidade para cenas com grande conteúdo de informação e desperdício de taxa de bit para cenas com pouco conteúdo de informação [ONV 95].

Tráfego CBR é fácil de monitorar na rede. Uma taxa constante é reservada para cada conexão CBR durante a sua duração, independente de estar ocorrendo uma transmissão ativa ou um estado de silêncio. Este é um uso ineficiente da capacidade de transmissão. Visto que a quantidade de informação gerada na rede varia com o tempo, é possível reservar menos capacidade na rede do que a taxa de pico da aplicação, permitindo a agregação de tráfegos e aumentando a eficiência na utilização dos recursos da rede [ONV 95].

Alocação de recursos para tráfego CBR

A fonte de uma conexão CBR envia uma célula periodicamente a cada $1/PCR$ intervalo de tempo. A alocação de recursos para tráfego CBR é baseada na taxa de pico (PCR) de cada conexão. Portanto, a banda necessária em uma porta de comutador para um grupo de conexões CBR é dado por:

$$BA_{CBR} = \sum_{i=1}^n PCR_i$$

Onde,

BA_{CBR} é a banda alocada para as conexões CBR;

PCR_i é a taxa de pico da conexão "i".

n é o número de conexões do tipo CBR nessa porta;

Quando as células atravessam diversos pontos de fila, a periodicidade de um tráfego CBR será perdida. As células podem se aproximar ou se dispersar devido à ação de outro tráfego sobre a mesma fila. Quando o tempo entre células diminui (quando se aproximam uma da outra) a taxa efetiva de transmissão aumenta em relação ao PCR declarado da fonte (fig. 3.8). Dessa forma, alocar o PCR declarado pode não ser suficiente para garantir o QoS da conexão. Essas variações não podem ultrapassar o

máximo CDV (*Cell Delay Variation*) comprometido pela rede. O CDV pode ser inserido em maior grau quando são multiplexados na mesma fila conexões de tráfego VBR-rt. Mesmo sem essa variação, a chegada simultânea de células (congestionamento em escala de célula¹) devido a multiplexação de conexões pode causar a perda de células em filas pequenas [GON 2001].

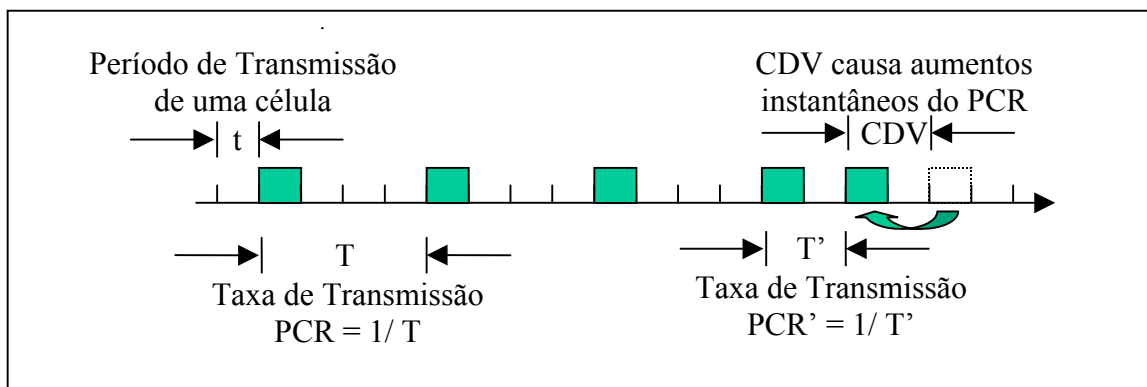


FIGURA 3.8 - Variação na taxa de pico de uma conexão CBR [GON 2001]

Alocação pela taxa de pico

O esquema mais simples de alocação lógica de banda é utilizar a taxa de pico da conexão como a quantidade de banda a ser reservada no ponto de fila. As vantagens dessa opção são:

- valor para alocação de banda é diretamente pego de um dos descritores de tráfego da conexão (PCR);
- mecanismo de policiamento da conexão é simples de ser implementado;
- a alocação não depende da característica de tráfego das outras conexões.

Portanto, a implementação dessa opção é muito simples, tanto do ponto de vista do usuário como da rede. A alocação de *buffer* também é simples. Para garantir taxas baixas de CLR, o *buffer* necessita ter o tamanho igual ao número de conexões que o compartilham. A alocação de *buffer* é feita de forma a acomodar a chegada simultânea de células de diferentes conexões (fig. 3.9). A qualidade de serviço em termos de CTD e CDV também é garantida para todas essas conexões visto que o *buffer* é de tamanho pequeno.

¹ Congestionamento em escala de célula ocorre em buffers pequenos devido à chegada simultânea de células de diferentes conexões. O outro tipo de congestionamento é o congestionamento em escala de rajada que ocorre tipicamente em buffers grandes devido à chegada de rajadas de células de diferentes conexões.

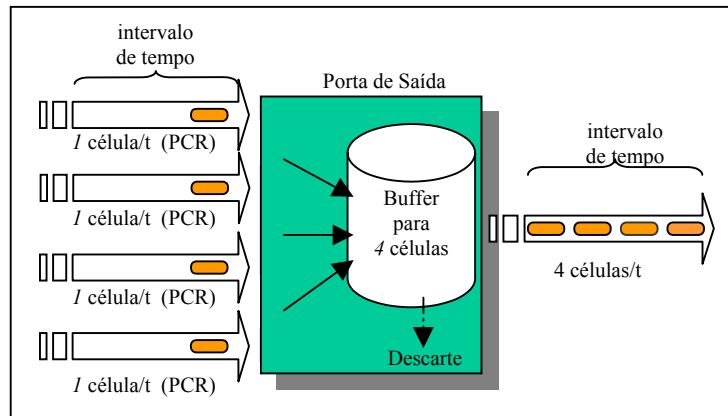


FIGURA 3.9 - Chegada simultânea de células [GON 2001]

A desvantagem óbvia desse modo de alocação é que os recursos da rede podem ser extremamente subutilizados quando a fonte das conexões tem características de tráfego com taxa de transmissão variável (fig. 3.10). A banda disponível das conexões que não estão transmitindo à taxa PCR poderia ser aproveitada por outras conexões. A alocação pela taxa de pico não tira proveito da capacidade de multiplexação estatística do ATM.

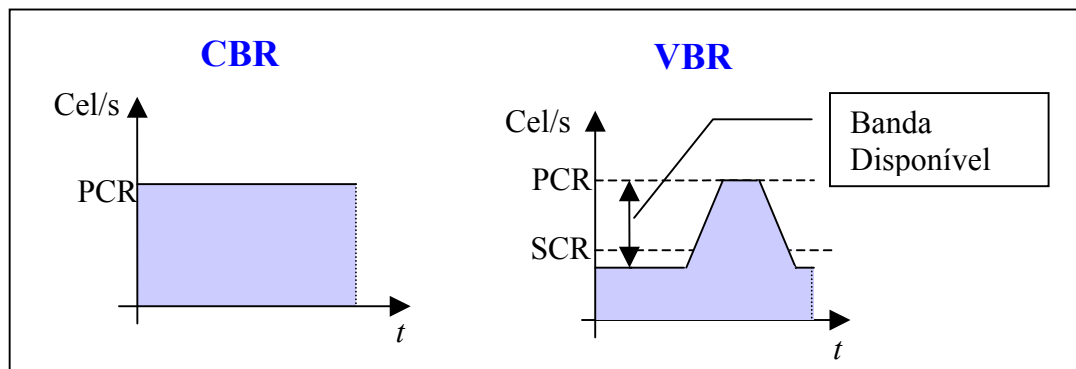


FIGURA 3.10 - Alocação pela taxa de pico para CBR e VBR [GON 2001]

3.3.1.2 Serviços VBR

Segundo ATM Forum, este tipo de serviço está dividido em duas subcategorias conforme os requerimentos temporais das aplicações. As subcategorias são referenciadas como: VBR-rt, real-time ou VBR-nrt, non-real-time [GIR 99].

Ambas categorias requerem largura de banda variável, controladas pelo CAC, durante a duração da conexão. A alocação de banda e recursos é feita considerando a taxa média de transmissão (SCR), a taxa máxima (PCR) e o tamanho máximo de rajada (MBS). A banda alocada é menor que o PCR e maior ou igual a SCR, ou seja, a rede determina os recursos alocados pelo PCR, SCR e MBS declarados pelo usuário [GIR 99].

No caso da categoria VBR-rt, com natureza de serviço de tempo real, requer

baixo atraso (CTD), baixa variação no atraso (CDV) e pequena taxa de perda de células (CLR), como a categoria CBR. Aplicações multimídia de tempo real, como vídeo conferência, são exemplos dessa categoria de serviço [GIR 99].

Na outra categoria, VBR-nrt, sendo a natureza não de tempo real, não há restrições quanto a atrasos (CTD) ou variações no atraso (CDV). Porém, é requerida uma baixa taxa de perda de células (CLR). Portanto, serviços VBR-nrt podem ser amplamente enfileirados. A diferença no tamanho das filas impacta diretamente na largura de banda alocada. Aplicações como correio eletrônico com imagem e vídeo sob demanda são exemplos para essa categoria [GIR 99].

As fontes do serviço VBR geram tráfego, durante períodos de atividade, na taxa máxima PCR. A duração dos períodos de transmissão e de silêncio é caracterizada pelo SCR. A alternância randômica entre períodos de atividade e silêncio das fontes VBR minimiza a probabilidade de duas fontes colidirem nos períodos de pico de transmissão. A variação na taxa de transmissão permite diversas fontes VBR serem multiplexadas estatisticamente sobre os mesmos recursos, otimizando a utilização da rede. Dessa forma, os serviços VBR-rt e VBR-nrt são projetados para os usuários compartilharem enlaces ATM de forma a propiciar o ganho estatístico¹ na utilização do meio [GIR 99].

Multiplexação Estatística

Engenheiros de tráfego chamam o efeito da combinação das saídas de várias fontes com taxas variáveis dentro de um canal de dados de multiplexação estatística. A figura 3.11 ilustra exemplos da atividade de fontes heterogêneas². Cada uma tem uma taxa de pico, uma taxa média e uma duração da rajada. O último gráfico mostra o resultado das taxas de bits agregados resultante da junção da saída das várias fontes [MCD 2000].

Redes ATM oferecem um ganho eficiente através da multiplexação estatística do tráfego dinâmico dos usuários. Multiplexação estatística permite que conexões múltiplas compartilhem a mesma largura de banda de uma porta de saída. É este compartilhamento da largura de banda que oferece o impulso na eficiência e vantagens de economia. O provedor do serviço pode suportar muitos mais assinantes pela venda da mesma porção de largura de banda muitas vezes. Para a empresa, o compartilhamento da largura de banda resulta em uma melhor utilização e redução do custo por bit da capacidade de transmissão. Considere, por exemplo, duas conexões que chegam em uma fila de saída de um switch ATM. A largura de banda requerida para cada conexão flutua, produzindo picos e vales em uma demanda agregada de largura de banda. Para oferecer um ganho eficiente o switch ATM aloca uma largura de banda agregada que é menor que a soma das taxas de picos de células. Para evitar perda de células, uma memória (buffer) é oferecida para armazenar o excesso de células. A porção de largura de banda e memória é oferecida na porta de saída de tal forma que a probabilidade de exceder a memória está dentro do nível aceitável.

O ganho potencial promovido pela multiplexação estatística está relacionado com o tamanho da memória. Com uma memória pequena, a alocação da largura de

¹ Ganho estatístico é definido como a razão entre o número de conexões admitidas pela taxa de pico de transmissão, chamado de multiplexação determinística, pelo número de conexões admitidas com multiplexação estatística.

² Fontes heterogêneas são geradores de fluxos independentes com diferentes parâmetros de tráfego e requisitos de QoS.

banda fica próximo da taxa de pico de célula requerida para garantir QoS. O resultado é um ganho de eficiência pequeno ou zero. Por outro lado, uma memória grande, reduz o requerimento de largura de banda e oferece um ganho de eficiência maior.

Memórias grandes podem aumentar a eficiência da largura de banda. Mas, se não for gerenciada corretamente, memórias grandes podem criar grandes atrasos e jitters no fluxo individual de células. Tais condições podem prejudicar a capacidade de oferecer a cada conexão individual o QoS negociado.

O gerenciamento da alocação de memória e largura de banda no switch é tarefa do gerenciamento de tráfego. O gerenciamento de tráfego é também responsável por realizar o impulso na eficiência sem impactar negativamente na qualidade do serviço.

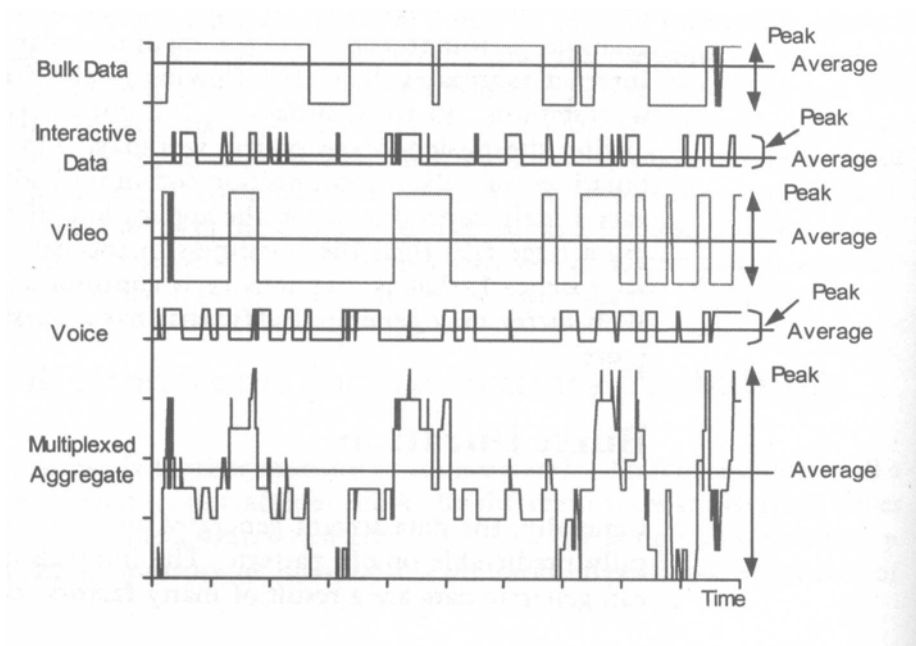


FIGURA 3.11 - Multiplexação estatística de fontes heterogêneas [MCD 2000]

Vídeo

Uma seqüência de vídeo consiste numa série de quadros no quais o movimento de uma cena é refletido em pequenas mudanças nos quadros mostrados seqüencialmente. Os quadros são mostrados em um terminal a uma taxa constante, como 24 ou 30 quadros por segundo¹, permitindo ao olho integrar as diferenças nos quadros com o movimento da cena [ONV 95]. Se um sinal de televisão é codificado de forma digital, uma largura de banda de aproximadamente 160 Mbits/s é requerida para transmiti-lo. Uma variedade de algoritmos de compressão é usada para reduzir a taxa de dados [SIL 2000].

Uma das propriedades de fluxos de vídeo é que alguns são mais difíceis de codificar eficientemente do que outros. Nos codificadores mais usados, os quadros são codificados usando aproximadamente o mesmo número de bits. Isto significa que o fluxo pode ser transmitido eficientemente sobre um canal de largura de banda constante.

¹ O número de quadros por segundo varia de país para país [ONV 95].

Tais codificadores são muitas vezes referenciados como codificadores CBR. Nesta abordagem alguns quadros são codificados com uma qualidade melhor do que outros [ROS 93].

Uma solução para este problema é fazer com que o número de bits por quadro de vídeo varie de forma a manter a qualidade de vídeo constante. Codificadores que se comportam desta maneira são conhecidos como codificadores VBR [ROS 93].

Voz

Uma fonte de voz alterna entre períodos ativos e períodos de silêncio. Voz CBR transmite ambos os períodos, o que é uma maneira ineficiente de usar os recursos da rede. Para atingir taxas altas de utilização de recursos, a detecção da atividade da fala deve ser usada na fonte de voz VBR. Assim pacotes de voz são gerados apenas quando a fonte está ativa, aumentando a eficiência da transmissão [ONV 95].

Os serviços de voz exigem um atraso máximo para evitar a sobreposição de diálogos e uma variação máxima do atraso para manter a continuidade (sincronismo) da voz, enquanto que uma perda de células moderada pode ser tolerada [STA 94].

O ITU-T formou um grupo para investigar e definir características de performance e objetivos para selecionar um único algoritmo de codificação, que irá manter os requisitos de várias aplicações de voz codificada. Uma vez finalizados estes trabalhos, deverá ser possível reduzir a taxa de bits requerida para conexões de voz [ONV 95].

Alocação de Recursos para Tráfego VBR

Redes ATM de alta velocidade são dimensionadas para suportar uma grande quantidade de conexões com as mais diversas características. O controle de admissão de chamadas tenta impedir o congestionamento da rede e, por conseqüência, a perda de QoS, limitando o número de conexões que serão admitidas na rede. Muitos esquemas de admissão de chamadas são baseados no conceito de Banda Efetiva ou Capacidade Equivalente ou Banda Equivalente. A Banda Efetiva de uma fonte de tráfego é definida como a largura de banda requerida para garantir determinada qualidade de serviço (QoS). Em tais esquemas, a chamada é admitida pelo comutador se a banda efetiva da fonte é menor ou igual à banda disponível no enlace de saída da conexão. A banda efetiva de uma conexão está entre seus valores de taxa média e de taxa de pico de transmissão da fonte, sendo que a banda efetiva de "n" conexões agregadas é geralmente menor que "n" vezes a banda efetiva de uma conexão simples. Esse fenômeno é conhecido como ganho da multiplexação estatística [GON 2001].

Alocação pela Banda Equivalente

Visto que a taxa máxima de transmissão de uma conexão é o PCR, ignorando o CDV adquirido, a rede não necessita alocar mais do que o PCR para satisfazer o QoS da conexão. Entretanto, desde que a conexão não envie células continuamente, é possível alocar uma banda menor quando diversas conexões são multiplexadas num ponto de fila. Esse valor resultante da multiplexação estatística que satisfaz o QoS é chamado de ***banda equivalente***. A banda equivalente se encontra entre a taxa de pico e a taxa média de transmissão da conexão (fig. 3.12) [GON 2001].

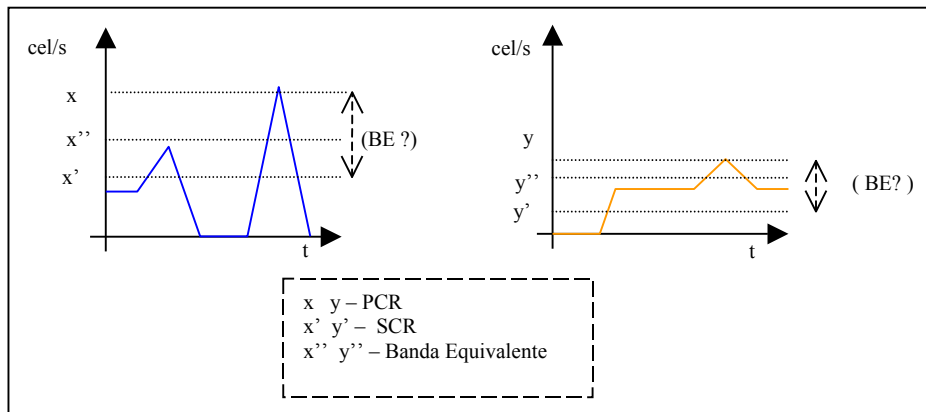


FIGURA 3.12 - BE: Banda Equivalente, $SCR \leq BE \leq PCR$ [GON 2001]

A soma das taxas de picos de todas as conexões é maior que a capacidade do enlace, enquanto que a soma de todas as bandas equivalentes é menor ou igual à capacidade do enlace. A eficiência na utilização da banda do enlace aumenta quando a banda equivalente das conexões se aproxima de suas taxas médias, e piora quando se aproxima dos picos das conexões. Em geral, a alocação pela banda equivalente permite um número maior de conexões do que a alocação pelo pico e, portanto, um melhor aproveitamento dos recursos da rede. O ganho de multiplexação estatística (*SGM - Statistical Multiplexing Gain*) no enlace pode ser definido pela seguinte expressão [GON 2001]:

$$SGM = \frac{\sum PCR}{\sum BE}$$

Onde, $\sum PCR$ é a soma das taxas de pico de todas as conexões no enlace;

$\sum BE$ é a soma das bandas equivalentes de todas as conexões no enlace.

O ganho de multiplexação estatística é determinado por características das conexões. Geralmente quando o número de conexões aumenta sobre um enlace, o fluxo agregado resultante tende a ser mais constante (com menor explosividade de rajadas). Isso porque os fluxos de rajada das conexões individuais estatisticamente tendem a se intercalar. Nesse caso, a banda equivalente do fluxo agregado se aproxima da taxa média de utilização do enlace, aumentando o ganho da multiplexação estatística, ou seja, a eficiência na utilização do enlace. Esse efeito não é alcançado quando se multiplexam conexões com pequenas rajadas, ou seja a taxa de pico é muito próxima da taxa média de transmissão. Há pouca área para o aproveitamento da multiplexação estatística e a banda equivalente se aproxima da taxa de pico. O mesmo acontece com a multiplexação de fontes de alta taxa de transmissão com fontes de baixa taxa. O efeito de compensação das fontes de baixa sobre as de alta para tornar a banda equivalente próxima da média de transmissão pode ser insignificante. O ganho é geralmente uma função do tamanho da fila, das características do tráfego e os objetivos de QoS das conexões que estão sendo multiplexadas. A banda equivalente também é denominada de banda efetiva ou capacidade equivalente [GON 2001].

A alocação pela banda equivalente é um dos métodos propostos na literatura

para o CAC. Ela consiste em determinar, através do estudo preciso da multiplexação estatística das fontes, a capacidade efetiva (capacidade estatística) requerida para cada uma delas. Na prática, uma nova conexão só é aceita se a rede dispuser de recursos para suportar a QoS requerida em cada porta de comutador (enlace) desde a origem até o destino por onde passa a conexão [GON 2001].

3.3.1.3 Serviço ABR (Available Bit Rate)

Para a categoria de serviço ABR a largura de banda é alocada dinamicamente, conforme a disponibilidade da rede, ou seja, a largura de banda dos enlaces físicos não utilizada pelas demais conexões fica automaticamente disponível para os serviços ABR. Portanto, a taxa de transmissão nessa categoria de serviço é dinâmica conforme a demanda da aplicação e a disponibilidade da rede. Por isso, este serviço é referenciado como serviço com largura de banda pela demanda. O principal objetivo deste serviço é conseguir obter o mais rápido possível uma parte da banda que é dinamicamente disponibilizada pela rede [GIR 99]. A rede utiliza células chamadas RM para notificar a fonte de tráfego ABR sobre o estado da rede. Quando a fonte recebe uma célula RM indicando congestionamento, essa deve imediatamente reduzir sua taxa de transmissão. Recebendo células RM indicando não haver congestionamento, a fonte aumenta a utilização da banda. Essa interação entre as células RM e a fonte ABR resulta em um ajuste apropriado na taxa de transmissão para evitar congestionamentos. Para ter certeza que a fonte está obedecendo a requisição de redução ou aumento das células RM é necessário o policiamento do tráfego por parte da rede [GON 2001]. O serviço ABR garante uma taxa mínima de transmissão (MCR) e é limitado por uma taxa máxima (PCR). O mecanismo de controle de fluxo na fonte de tráfego minimiza o potencial de perda de células. Esse serviço foi projetado para acomodar tráfego de dados que requer uma garantia específica em relação à taxa de perda de células (CLR), no entanto não possui requisitos quanto ao atraso de transferência (CTD) ou sua variação (CDV). Exemplo de aplicação que pode utilizar esse serviço é a transferência de arquivos [GIR 99].

Dados

Este tipo de tráfego representa os serviços de comunicação entre computadores, onde há uma grande sensibilidade à perda de células, mas as exigências quanto ao atraso e sua variação são menores. Estes serviços incluem transferência de arquivos, emulação de terminal, entre outros [STA 94].

A despeito das décadas que já se utiliza redes para transmissão de dados, as características de tráfego de fontes de dados ainda não são bem entendidas. A principal dificuldade deve-se ao fato de que não existem conexões típicas de dados. É bastante variado o tipo de aplicação que gera este tipo de tráfego, desde a transmissão de grandes quantidades de bytes em uma transferência de arquivo a poucas centenas de bytes gerados por um e-mail. Algumas conexões requerem que conexões fim-a-fim sejam estabelecidas, enquanto outras são mais bem servidas no modo sem conexão. Serviços de dados interativos requerem troca de dados bidirecional, enquanto atualização de bancos de dados pode usar apenas conexões unidirecionais [ONV 95].

Alocação de Recursos para Tráfego ABR

O tráfego para conexões de categoria ABR tem a garantia de uma taxa mínima de banda dada pelo parâmetro de tráfego MCR. Portanto, a alocação de recursos para tráfego ABR é baseada na taxa mínima [GIR 99].

A banda necessária alocada em uma porta de comutador para um grupo de conexões ABR é dado por [GON 2001]:

$$BA_{ABR} = \sum_{i=1}^n MCR_i$$

Onde,

BA_{ABR} é a banda alocada para o conjunto das conexões ABR;

MCR_i é a taxa mínima de transmissão da conexão "i";

n é o número de conexões do tipo ABR nessa porta.

A categoria ABR não oferece garantia de CLR, no entanto a taxa de perda de células deve ser mantida baixa quando a conexão transmite dentro de valores do MCR. Da mesma forma, tráfego ABR não possui restrições quanto ao CTD e CDV, o que possibilita ter *buffers* grandes para acomodar a chegada simultânea de células e rajadas de células causadas pela multiplexação de conexões. O dimensionamento da capacidade de *buffer* para tráfegos de dado (categoria de serviço ABR e UBR) tem como objetivo restringir a probabilidade de perda de células dentro de valores desejáveis pela rede. A seguinte expressão é uma aproximação para o dimensionamento de *buffer* em comutadores servindo fluxos de tráfego Markoviano (modelado como M/M/1/B) em função da carga do enlace e da desejada probabilidade de perda [MCD 2000]:

$$BU_{ABR} \cong \frac{-\ln(\varepsilon)}{1-\rho} \quad \rho = \frac{\sum_{i=1}^n MCR_i}{BD}$$

Onde,

BU_{ABR} é o tamanho do buffer necessário para o conjunto das conexões ABR;

ε é o valor de probabilidade de perda desejável. Exemplo: 10^{-4} ;

ρ é o fator de carga na utilização da banda para as conexões ABR. Varia de 0 a 1, sendo 1 correspondente a 100% de utilização da banda disponível para as conexões ABR;

MCR_i é a taxa mínima de transmissão da conexão "i";

n é o número de conexões do tipo ABR nessa porta;

BD é a banda disponível para as conexões ABR. É dada como a diferença entre a capacidade do enlace e o total de banda alocada pelas conexões CBR e VBR.

Dessa forma podemos alocar um tamanho de *buffer* para garantir a probabilidade de perda ε para as conexões ABR com uma taxa mínima de transmissão.

Alocação de Banda pela Taxa Mínima

A alocação pela taxa mínima é adequada para serviços de dados com fluxos controlados. Nesse caso, a rede aloca para a conexão a banda correspondente à taxa mínima de transmissão de células. O tráfego excedente ainda pode ser carregado pela rede desde que haja banda disponível no momento. A qualidade de serviço é garantida para as células que são transmitidas até a taxa mínima. Portanto, o tráfego excedente deve ser controlado pela fonte com algum mecanismo de informação baseado no estado de congestionamento da rede. A vantagem desse esquema de alocação é seu potencial para a disponibilização dinâmica dos recursos da rede, tornando-os menos ociosos. A desvantagem é que o tráfego excedente ao mínimo alocado não tem garantia de QoS.

3.3.1.4 Serviço UBR (Unspecified Bit Rate)

Para esta categoria de serviço não existe previsão de taxas de tráfego e de QoS. O controle de tráfego é muito simples. Conexões UBR compartilham a largura de banda sem utilizar qualquer mecanismo de controle de congestionamento. O usuário não informa nenhum tipo de parâmetro do descritor de tráfego. Portanto, a rede não garante nenhuma qualidade de serviço. A banda disponível para serviços UBR é a restante na rede depois de atender todas as demais conexões e não possui qualquer controle de fluxo. Protocolos superiores ao nível ATM, como TCP, devem se preocupar em retransmitir as informações perdidas. Células transmitidas acima da banda disponível são descartadas. As informações são transmitidas pelas fontes UBR entendendo que a rede fará o melhor para levá-las ao destino, porém isso pode não acontecer. Esse comportamento caracteriza o serviço UBR como sendo de melhor esforço. O serviço UBR é atrativo porque é muito simples de ser suportado na rede, não há reserva de recursos e as fontes não necessitam conhecer suas características de tráfego. Exemplo de aplicação de serviços UBR são: transferência de arquivos e correio eletrônico [GIR 99].

3.3.2 Serviços definidos na arquitetura IntServ/RSVP

3.3.2.1 Serviço de QoS Garantido

O serviço garantido é uma classe de QoS proporcionado pelo modelo de serviços integrados que oferece um nível assegurado de largura de banda, um limite rígido de atraso fim a fim e uma proteção contra a perda de pacotes nas filas, para os pacotes que estiverem obedecendo o perfil de tráfego contratado. É direcionado para aplicações com requisitos rígidos de tempo real, como certas aplicações multimídia intolerantes, que precisam de uma garantia firme de que um pacote não irá chegar no destino depois de um tempo maior que um limite especificado. Esse serviço não oferece garantia mínima da variação do atraso, ele simplesmente garante um atraso máximo gerado pelas filas [KAM 2000].

A obtenção de um limite máximo para o atraso exige que todos os roteadores no caminho suportem o serviço garantido. O comportamento fim a fim oferecido por uma série de roteadores que implementam o serviço garantido é um nível assegurado de largura de banda para um determinado fluxo que, quando utilizado por um fluxo que está sendo policiado, produz um serviço com atraso limitado para todos os pacotes que

estejam dentro do perfil [KAM 2000].

Para ter acesso a esse serviço, as aplicações conformam os seus fluxos através de um mecanismo de balde de fichas e a partir dos valores de taxa da rajada, cada roteador calcula vários parâmetros descrevendo como ele tem que tratar os pacotes desses fluxos. Combinando os parâmetros dos vários roteadores em um caminho, é possível calcular o atraso máximo que um pacote irá experimentar quando transmitido por aquele caminho. Uma vez que as aplicações podem controlar os valores de taxa e rajada dos fluxos, elas conseguem obter uma garantia provada matematicamente sobre o atraso máximo dos seus pacotes [KAM 2000].

O serviço garantido necessita de controle de admissão para operar de acordo com as especificações. Teoricamente, ele pode ser utilizado com qualquer protocolo de reserva de recursos, mas apenas a sua utilização em conjunto com o RSVP foi especificada [KAM 2000].

3.3.2.2 Serviço de Carga Controlada

O serviço de carga controlada não oferece garantias quantitativas rígidas, como o serviço garantido. O comportamento fim a fim oferecido para uma aplicação por uma série de roteadores que implementa esse serviço se assemelha ao comportamento visto por aplicações que estão recebendo o serviço de melhor esforço em uma rede com carga leve, ou seja, sem nenhuma situação grave de congestionamento. As garantias que as aplicações têm são [KAM 2000]:

- Um percentual muito alto de pacotes transmitidos chegará com sucesso no receptor (deve se aproximar da taxa básica de erros do meio de transmissão, ou seja, pouquíssimos descartes em filas ocorrem).
- O atraso sofrido por um alto percentual dos pacotes não deverá exceder muito o atraso mínimo sofrido por um pacote dentro de um fluxo, ou seja, a maior parte dos pacotes deve ter um atraso muito próximo do atraso mínimo.

Para assegurar que essas condições serão válidas, aplicações que requisitam o serviço de carga controlada devem oferecer aos roteadores uma estimativa do tráfego de dados que elas irão gerar, chamada de Tspec, que é baseada em um balde de fichas. Como resposta, o serviço assegura que a aplicação terá a sua disposição recursos dos roteadores suficientes para processar adequadamente todos os pacotes que estiverem de acordo com a especificação contido no Tspec. Por outro lado, pacotes introduzidos na rede fora das especificações, poderão ser descartados, ou enfrentar um atraso mais significativo [KAM 2000].

O objetivo do serviço de carga controlada é suportar uma ampla classe de aplicações que tem sido desenvolvida para a Internet atual, mas que não funcionam em situações de carga alta na rede. Alguns membros dessa classe são as aplicações de tempo real adaptáveis, atualmente sendo oferecidas inclusive comercialmente. Essas aplicações têm mostrado que funcionam bem com redes com carga leve, mas a qualidade se degrada rapidamente em condições de congestionamento. Um serviço que imita redes com carga leve é útil para essas aplicações [KAM 2000].

As aplicações podem solicitar o serviço de carga controlada antes de iniciar as transmissões, ou então quando elas detectam que o serviço de melhor esforço não está oferecendo um desempenho aceitável. A primeira estratégia oferece uma maior garantia

de que o nível de QoS não irá mudar enquanto durar a sessão. A segunda estratégia é mais flexível e barata, pois o serviço com tarifação mais alta não é utilizado durante todo o tempo de duração da sessão [KAM 2000].

3.4 Implementação de QoS

A implementação de QoS em redes de computadores está associada à existência de mecanismos de condicionamento de tráfego, políticas de filas (disciplinas de serviços), reserva de recursos e controle de admissão. O condicionamento de tráfego está relacionado com o policiamento realizado para garantir que o tráfego entre usuário e provedor esteja dentro dos perfis definidos em um contrato de serviço. O condicionamento envolve a classificação de pacotes, medição de tráfego e uma subsequente ação, para os pacotes que não estão dentro do perfil de tráfego contratado.

3.4.1 O Fenômeno congestionamento

Congestionamento é a condição encontrada quando a demanda por um recurso excede a disponibilidade deste recurso. Congestionamento é inevitável quando ocorrem expansões do sistema. O tráfego aumenta e os recursos são mantidos estáticos, uma sobrecarga eventualmente pode ocorrer, levando a uma situação de congestionamento. Este padrão se repete por todo o mundo. O congestionamento é um fenômeno dinâmico causado por eventos imprevisíveis. Em redes, IP e/ou ATM, esse fenômeno causa atrasos e perdas, ou ambos. Atrasos e perdas reduzem a qualidade do serviço oferecido para o usuário da aplicação final, muitas vezes reduzindo sua produtividade e algumas vezes tornando a aplicação não viável.

Embora a melhor solução para o congestionamento é simplesmente evitar situações, onde e quando congestionamento pode ocorrer, esta estratégia nem sempre é possível. Infelizmente, congestionamento ocorre em muitos ambientes de redes reais porque existe sempre um gargalo – um computador lento, um link de baixa velocidade, uma sobrecarga no servidor ou switch ou roteador intermediário com baixa vazão ou atraso excessivo.

3.4.2 Controle de Tráfego e Congestionamento

A recomendação I.371 define congestionamento como sendo um estado da rede que, se atingido, leva a degradação da qualidade de serviço das conexões estabelecidas. A rede não é mais capaz de “honrar” os compromissos de desempenho assumidos no momento em que as conexões foram estabelecidas. A tentativa de tratar congestionamentos pode ser feita de forma preventiva, sendo, nesse caso, denominada controle de tráfego, onde todos os esforços estão concentrados em evitar o estado de congestionamento, ou de forma reativa, sendo denominada controle de congestionamento, quando a rede reage a indícios de congestionamento tentando aliviar o tráfego na rede. Técnicas de controle de tráfego e controle de congestionamento podem ser associadas de forma a contribuir para a garantia da qualidade de serviço das conexões na rede [SOA 95]. Logo, o principal objetivo do controle de tráfego é proteger a rede e seus usuários para alcançar seus objetivos de desempenho predefinidos na rede

em termos de taxa de perda, atraso e variação do atraso das célula/pacotes. Basicamente, o controle de tráfego refere-se a um conjunto de ações tomadas pela rede para evitar congestionamento. Um objetivo adicional do controle de tráfego é otimizar o uso dos recursos da rede com o propósito de atingir uma eficiência realística da rede [PRY 95].

Podemos dizer então, que o controle de tráfego tem como objetivos:

- Conformar o tráfego para auxiliar a obtenção da qualidade de serviço;
- Minimizar congestionamento na rede;
- Maximizar o uso eficiente dos recursos da rede.

Roteador/Switch genérico em um arquitetura com QoS

Roteadores IP e Switches ATM empregam técnicas similares para oferecer QoS. Como ilustrado na figura 3.13, o roteador ou switch é uma coleção de portas interconectadas por um sistema de comutação. O sistema pode ser um barramento simples ou uma matriz complexa de elementos de comutação capazes de suportar uma capacidade agregada de trilhões de bits por segundo ou mais. O sistema conecta um número de portas, o qual na figura 3.13 está dividido em portas de entrada e de saída. Primeiro, iniciando do lado de entrada de cada porta, a função classificadora determina o QoS e o perfil de controle de tráfego requerido por cada fluxo ou conexão que as funções de enfileiramento e escalonamento irão empregar. Depois, uma função de policiamento opcional assegura conformidade do fluxo IP ou conexão ATM em relação aos parâmetros de tráfego. O sistema de comutação deve também realizar enfileiramento e escalonamento antes de enviar os pacotes ou células para a destinada porta de saída [MCD 2000.]

Na porta de saída, a função classificadora determina se invoca ou não a função de reformatação. O classificador também controla o enfileiramento e o serviço de escalonamento oferecido para cada pacote ou célula no lado de saída. Juntas, as funções do lado de entrada, de dentro do sistema e as do lado de saída, asseguram que cada nodo de rede IP ou ATM atinge o desempenho requerido necessário para alcançar o QoS fim-a-fim para um fluxo ou uma conexão[MCD 2000].

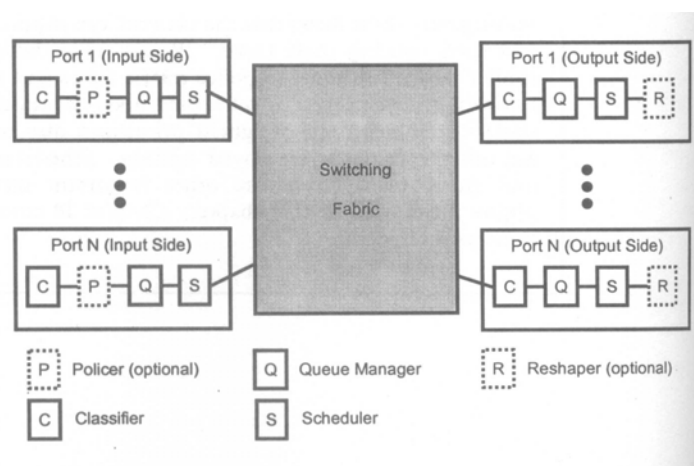


FIGURA 3.13 - Roteador/Switch genérico em um arquitetura com QoS [MCD 2000]

Controle de Admissão em IP e ATM

Controle de admissão definitivamente decide se admite ou rejeita a requisição para adicionar um o novo fluxo ou conexão baseado sobretudo se o recém-chegado violará o QoS oferecido dos fluxos ou conexões já existentes.

ATM, IntServ/RSVP, MPLS e DiffServ, todos operam efetivamente no paradigma orientado a conexão, e por isto eles tem políticas similares de controle de admissão. Controle de admissão envolve que cada nodo verifique toda requisição contra capacidade disponível e capacidades de QoS atual. O nodo admite a requisição somente se ele poder oferecer o QoS requerido depois de ter adicionado o tráfego correspondente para as conexões existentes [MCD 2000].

Em ATM, o CAC é uma função comumente implementada por software dentro dos swiches ATM que determinam se admite ou rejeita requisição de conexão. A figura 3.14 ilustra o algoritmo mais simples do CAC que é pela taxa de pico [MCD 2000].

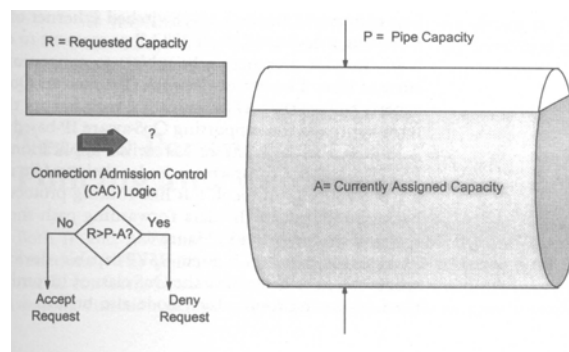


FIGURA 3.14 - Algoritmo do CAC pela taxa de pico [MCD 2000]

Em IP, aplicações destinatárias utilizam RSVP para indicar aos nodos no sentido contrário (destino para origem) seus requerimentos de tráfego. RSVP não é um protocolo de roteamento, mas como eles, ele opera em segundo plano. A figura 3.15 ilustra a operação do RSVP em hosts e roteadores. O classificador de pacotes em cada equipamento com capacidade RSVP ativa utiliza a especificação de filtro para determinar a classe de QoS dos pacotes de dados que chegam, e depois selecionar a rota. Cada nodo também utiliza um selecionador de pacotes, empregando métodos tais como formatação do tráfego no nível de pacotes, filas de prioridades, WFQ para alcançar o QoS requerido. A função de controle de admissão, residente em nodos RSVP ao longo do caminho entre destino e origem, interpreta a especificação do fluxo nos pacotes de controle RSVP e determina se o nodo tem recurso suficiente para suportar o fluxo de tráfego requerido. O nodo também deve executar uma política de controle. Tanto se o processo de controle de admissão quanto o da política de controle falhar, o processo RSVP no nodo retorna uma mensagem de erro para a aplicação requisitora [MCD 2000].

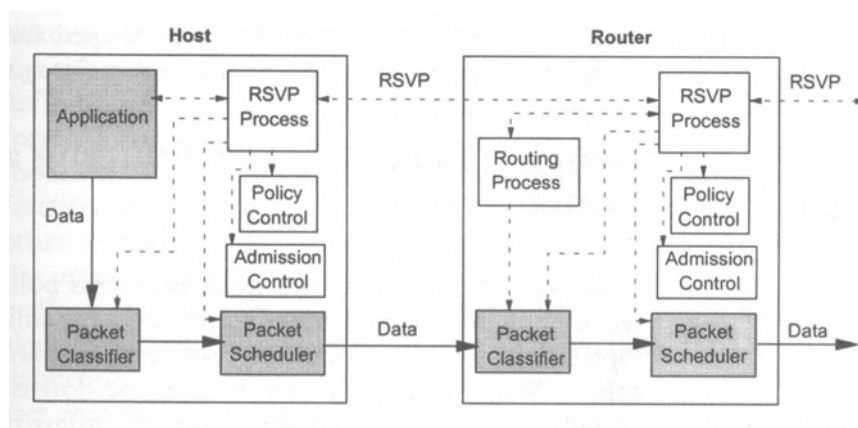


FIGURA 3.15 - Operação do RSVP em hosts e roteadores [MCD 2000]

Policimento e Formatação em IP e ATM

Tanto elementos de redes IP como de redes ATM realizam funções de policiamento e formatação.

A função de policiamento verifica para ver se o tráfego está chegando conforme o contrato de tráfego específico [MCD 2000]. Consiste no conjunto de ações tomadas para monitorar e controlar o tráfego em relação aos parâmetros estabelecidos e negociados durante a fase de estabelecimento (ou renegociação de parâmetros) de uma conexão. O principal objetivo é proteger os recursos da rede contra a má utilização, tanto a maliciosa como a não intencional, de fontes de tráfego que excedam os limites especificados pelos descritores de tráfego e possam, potencialmente, ocasionar congestionamentos [SOA 95]. Uma técnica de policiamento conhecida em redes ATM é o “leaky bucket”, denominação essa originada da analogia que se faz com o escoamento de um líquido através de um furo no fundo de um balde.

Já a função de conformação adia a saída de células ou pacotes para assegurar conformidade com o contrato de tráfego [MCD 2000]. Fontes com característica de rajada, ou seja, com grande diferença entre a média e o pico de transmissão, têm longos períodos de baixa atividade seguidos por transmissões à taxa de pico. Aproveitando o comportamento desse tipo de tráfego é possível reduzir a taxa de pico através do enfileiramento (*bufferização*) das células/pacotes antes delas entrarem na rede, de forma que a taxa de saída dessa fila seja menor que a taxa de pico. Esse procedimento é chamado de conformação de tráfego (*traffic shaping*) e é implementado em um ponto de acesso à rede. São dois os propósitos do *traffic shaping*: economizar largura de banda e conformar o tráfego para evitar descarte de células/pacotes. Para ter certeza que as células/pacotes de uma fonte de tráfego estarão conformes, um usuário pode utilizar um software formatador de tráfego antes que esse dê ingresso na rede [GON 2001].

Correspondência entre os parâmetros do tráfego em IP e ATM

A RFC 2381 [GAR 98] do IETF define o mapeamento de vários parâmetros e os procedimentos para invocá-los. A tabela 3.2 ilustra o mapeamento dos protocolos de sinalização, algoritmos e parâmetros. Como pode ser visto na tabela, a maioria dos parâmetros tem correspondência um para um.

TABELA 3.2 - Correspondência entre parâmetros e protocolos em IP e ATM

Parâmetros/Protocolo	IP	ATM
Protocolo de sinalização	RSVP	SVC (Q.2931)
Parâmetros do tráfego	Token Bucket Pico(p), taxa (r), capacidade (b) tamanho max pacote (M) unidade min policiada (m)	Leaky Bucket PCR, SCR, MBS
Parâmetros de QoS	Atraso	CLR, CTD, CDV (Perda, Atraso, Jitter)
Classes de serviços	Melhor esforço Carga controlada Serviço garantido	UBR nrt-VBR, ABR CBR, rt-VBR
Conformidade	Trata excesso como melhor esforço	Marca excesso usando CLP

Fonte: [MCD 2000]

3.4.3 Parâmetros de QoS em IP e ATM

A tabela 3.3 e 3.4 mapeiam as métricas para os parâmetros de QoS nas rede IP e ATM.

TABELA 3.3 - Parâmetros de QoS em redes ATM

Acrônimo QoS	Nome do parâmetro de QoS	Negociável?
p2pCDV	Variação do atraso de transferência da célula	Sim
MaxCTD	Máximo atraso de transferência da célula	Sim
CLR	Razão de perda de células	Sim
CER	Razão de células erradas	Não
SECBR	Razão de blocos de células severamente errados	Não
CMR	Taxa de células inseridas incorretamente	Não

Fonte: [MCD 2000]

TABELA 3.4 - Parâmetros de QoS em redes IP

Parâmetro de QoS	Carga controlada	QoS garantido
Varição máxima do atraso	Não especificado	Medido automaticamente usando Adspec
Taxa de pacotes perdidos	Pequena ou sem perda por congestionamento Deve ser limitado erro	Suficiente memória (buffer) requerida para não ter perda por congestionamento
Mínimo atraso	Alta porcentagem de pacotes não excede o atraso mínimo	Não especificado
Atraso médio	Pequeno ou nenhum atraso em filas	Não especificado

Fonte: [MCD 2000]

3.5 Modelagem de Tráfego

Até algum tempo atrás, o tráfego de rede era caracterizado predominantemente por fluxos de pacotes de dados. O advento das redes de alta velocidade e das aplicações multimídia, com características diferentes das aplicações tradicionais, ensejou o uso de novos modelos de distribuição de tráfego de rede. Estes novos modelos têm a capacidade de cobrir estatísticas que representam melhor as características observadas no tráfego multimídia.

Na modelagem de tráfego é importante definir precisamente o que está sendo modelado, e deve estar claro que aspectos de desempenho da rede se quer prever. Neste contexto, não importa diretamente se o modelo é um modelo de tráfego preciso, mas que os aspectos do tráfego que têm um impacto no desempenho da rede sejam modelados suficientemente bem para que os resultados sejam úteis. Um outro critério que deve ser considerado na modelagem é a tratabilidade analítica do modelo. Embora isto seja de menor prioridade do que a precisão da estimativa do desempenho da rede, é importante ter modelos disponíveis para os quais parâmetros como atraso e probabilidade de perda de células/pacotes possam ser calculados [SIL 2000].

3.5.1 Caracterização de Tráfego

"Para projetar e desenvolver novas funções de rede, é necessário compreender as características e os requisitos do tráfego que nela passará [ONV 95]."

Visto a afirmação acima, fica claro que conhecer a natureza do tráfego que vai passar em uma determinada rede vai influenciar toda a sua estrutura, seu funcionamento e os mecanismos que efetivam o nível de QoS demandado pelas aplicações, pois é exatamente baseado na caracterização deste tráfego que ocorrerá toda uma definição de parâmetros que depois serão utilizados para analisar seu desempenho, seja de forma direta (por medições) ou indireta (pelo sentimento dos seus usuários).

De acordo com a recomendação I.311 do ITU-T, os parâmetros relevantes para a caracterização de tráfego são [SIL 2000]:

- p Taxa de pico de células no período ativo, ou intervalo mínimo entre duas células consecutivas de uma mesma conexão.
- m Taxa média de células. Pode ser caracterizada pela taxa média real de chegada de células (número total de células de uma conexão dividido pelo tempo de duração) ou pela taxa média estimada de chegada de células (número de células geradas em um período longo dividido por este período).
- β *Burstiness* ou explosividade, que é a razão entre a taxa de pico de células e a taxa média de células (p / m), podendo ser interpretado como uma medida da duração do período ativo.
- t_{on} Duração média do período ativo

Padrões de tráfegos em redes não são geralmente prognosticáveis. Enquanto muito se sabe sobre o padrão de tráfego de voz, tipos de tráfego em rajadas, tais como, vídeo, dados, e voz comprimida, são menos conhecidos. Pesquisas recentes indicam que

o tráfego de rede tem propriedades estatísticas similares em escalas de intervalo de tempo em: milissegundos, segundos, minutos e horas, igualmente em dias e semanas. Estas características são referenciadas como auto-similaridade. Aparece em tráfegos de LAN, tráfegos ATM, fluxos de vídeo digital comprimido, e tráfego WEB.

3.5.2 Modelos de Tráfego

3.5.2.1 O Processo de Poisson

Este modelo é o mais simples e foi usado durante décadas na análise e projeto de redes de pacotes. Tem sido um dos mais utilizados na modelagem de chegada de pacotes ou células em redes de dados [SIL 2000].

Um processo estocástico $\{N(t), t \geq 0\}$ é chamado de *processo de contagem* se $N(t)$ representa o número total de eventos que ocorreram até o tempo t . Um processo de contagem $N(t)$ deve satisfazer [SIL 2000]:

- $N(t) \geq 0$
- $N(t)$ assume somente valores inteiros
- Se $s < t$, então $N(s) \leq N(t)$
- Para $s < t$, $N(t) - N(s)$ fornece o número de eventos que ocorreram no intervalo $(s, t]$

Um processo de contagem $\{N(t), t \geq 0\}$ é chamado de *processo de Poisson* com taxa média λ ($\lambda > 0$) se [SIL 2000]:

- $N(0) = 0$, ou seja, nenhum evento ocorre exatamente no tempo 0
- processo possui *incrementos independentes*, ou seja, o número de eventos que ocorrem em intervalos de tempos diferentes são independentes.
- número de eventos que ocorrem em qualquer intervalo de tempo t é distribuído de acordo com uma variável aleatória do tipo Poisson, com média λt . Isto é, para todo $s, t \geq 0$:

$$P[N(t+s) - N(s) = n] = e^{-\lambda t} (\lambda t)^n / n!, \quad n = 0, 1, \dots \quad (3.1)$$

Um processo de contagem possui *incrementos estacionários* se a distribuição do número de eventos que ocorrem em qualquer intervalo de tempo depende somente do tamanho do intervalo de tempo. Portanto, o processo tem incrementos estacionários se o número de eventos no intervalo $(t_1 + s, t_2 + s]$ tem a mesma distribuição que o número de eventos no intervalo $(t_1, t_2]$ para todo $t_1 < t_2$ e $s > 0$ [SIL 2000].

Relação entre o processo de Poisson e a distribuição exponencial

Seja uma variável aleatória t , que representa o *tempo entre chegadas adjacentes* em um sistema de fila, por exemplo, e cuja FDP (função de distribuição de probabilidade) e fdp (função de densidade de probabilidade) são dadas por $A(t)$ e $a(t)$, respectivamente. $A(t)$ é por definição a probabilidade de que o tempo entre chegadas

seja menor ou igual à t [SIL 2000].

$$A(t) = 1 - P[\tau > t]$$

$P[\tau > t]$ é a probabilidade de que não ocorra nenhuma chegada em $(0,t)$, ou seja, $P[\tau > t] = P[N(t) = 0]$. Da equação (3.1) tem-se que $P[\tau > t] = e^{-\lambda t}$. Logo,

$$A(t) = 1 - e^{-\lambda t}, \quad t \geq 0 \quad (3.2)$$

Da derivação da equação obtém-se a fdp

$$a(t) = \lambda e^{-\lambda t}, \quad t \geq 0 \quad (3.3)$$

conhecida como distribuição *exponencial*.

Das equações (3.2) e (3.3) vê-se, que para um processo de Poisson o tempo entre chegadas é segundo a distribuição exponencial. O processo de chegadas de Poisson tem tempo entre chegadas exponenciais [SIL 2000].

A distribuição exponencial possui uma característica conhecida como *falta de memória*. Os valores futuros de uma variável aleatória, distribuída exponencialmente, não dependem da sua história passada. A distribuição exponencial é a única distribuição contínua com essa propriedade [OLI 97]. Por isso é largamente utilizado para modelar o processo de chegada de pacotes de dados independentes.

Nesta dissertação, baseado em [ROB 96], [SIL 2000], [ONV 95] e [OLI 97], para todos os fluxos multimídia, o gerador de fontes utilizado para modelar o tráfego foi o VBR Poisson. Os períodos de atividade/silêncio (On/Off) do tráfego são exponencialmente distribuídos através dos valores médios informados. A escolha do VBR Poisson nos experimentos é devido à simplicidade de caracterização das fontes e ao fato de que modelos On/Off com os períodos distribuídos exponencialmente têm sido muito usados em estudos de avaliação de desempenho e projeto de redes e por serem genéricos o suficiente para modelar tanto voz quanto dados e vídeo [OLI 97].

3.5.2.2 O Modelo de Tráfego Auto-Similar

Motivados pelo estudo de interconexão de redes locais através de ATM, Leland e Wilson [LEL 91] obtiveram medidas de tráfego de redes locais de alta qualidade, sem perdas e com precisão no tempo de chegada dos pacotes de 100ms. Uma análise estatística dessas medidas [LEL 91] mostrou que o tráfego apresenta *explosividade* em várias escalas de tempo. Num artigo posterior, Leland e al. [LEL 94] mostraram que os dados obtidos nessas medidas podem ser modelados através de um processo estocástico *auto-similar* ou *fractal*, que apresenta características bastante diferentes daquelas dos processos de Poisson e dos modelos Markovianos.

O trabalho de Leland e Wilson [LEL 94] foi pioneiro no estudo da auto-similaridade do tráfego de redes. Eles analisaram rigorosamente as medidas de tráfego Ethernet descritas em [LEL 91]. Essas medidas foram realizadas em redes locais de um laboratório da Bellcore entre agosto de 1989 e fevereiro de 1992. Para cada pacote observado no barramento Ethernet o monitor gravava um tempo de chegada. Investigaram o tráfego interno, o tráfego externo (com a Internet) e o externo TCP. Concluíram que todos eles são auto-similares. Visualmente, a auto-similaridade do tráfego manifesta-se como uma invariância em relação à escala de tempo. Se forem

construídos gráficos "Pacotes/unidade de Tempo" x "Unidade de Tempo", para diversas unidades de tempo, pode-se verificar que o tráfego parece o mesmo na escala de horas, minutos, segundos e milissegundos (fig. 3.16). Em particular, não se identifica um tamanho natural de rajadas, que estão presentes em todas as escalas de tempo [PEA 96]. Estimativas do parâmetro de auto-similaridade das amostras obtidas mostraram que à medida que aumenta a utilização da rede Ethernet (o número de fontes ativas) a *explosividade* do tráfego intensifica. Isso revela uma forte discordância entre o tráfego medido numa rede real e os modelos de tráfego utilizados correntemente, tais como Poisson e processos de Poisson modelados por Markov. Nesses modelos, o tráfego agregado torna-se mais suave (menos explosivo) com o aumento da quantidade de fontes. Isso indica que mecanismos de controle de tráfego de redes devem ser reavaliados [OLI 97].

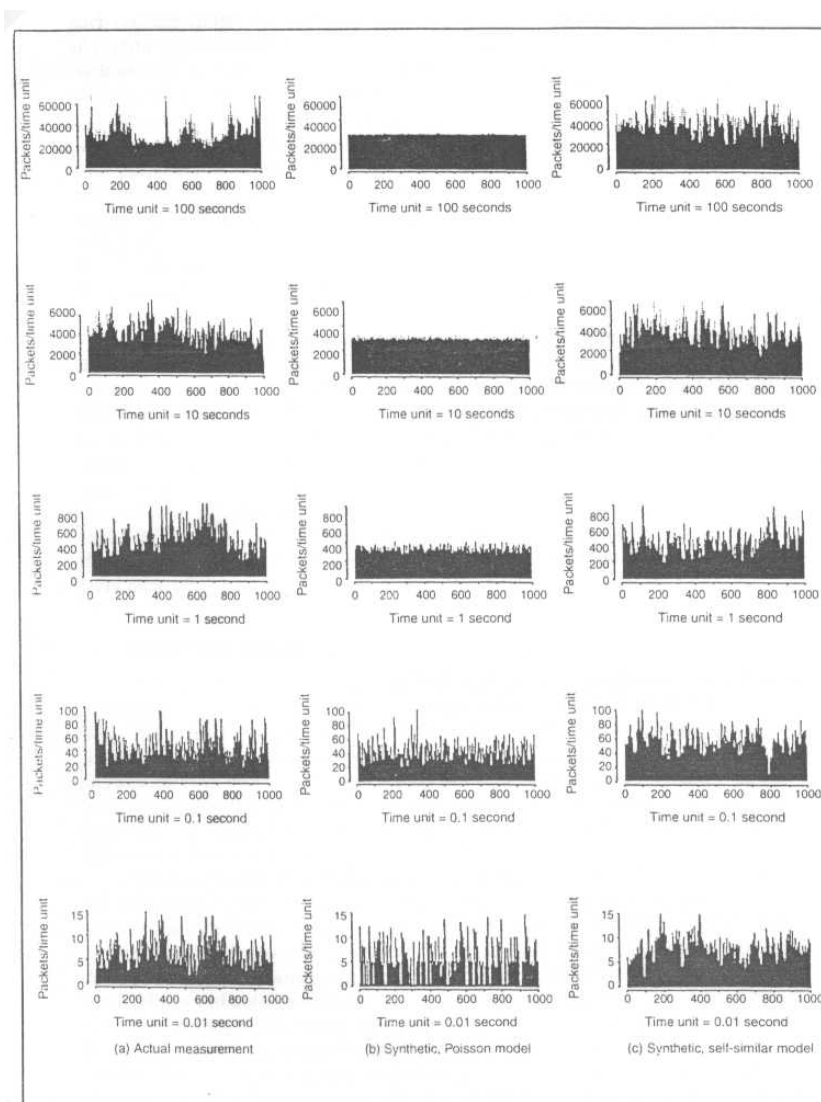


FIGURA 3.16 - Pacotes/unidade de Tempo" x "Unidade de Tempo [STA 98]

A partir desse estudo pioneiro vários tráfegos de diferentes serviços têm sido investigados visando identificar neles o comportamento auto-similar. Desde 1993, estudos reportados na literatura têm documentado que o padrão de tráfego de dado é

mais bem modelado por processos auto-similares em uma vasta variedade de situações do mundo real das redes, tais como, tráfego Ethernet, tráfego WEB, tráfego TCP, FTP e TELNET e vídeo digitais [STA 98].

A propriedade de auto-similaridade pode ser observada através de uma comparação, por meio de gráficos, entre o processo de Poisson e tráfego medido em redes Ethernet. A amostra de tráfego é resultante das medidas descritas em [LEL 91] e analisadas em detalhes em [LEL 94]. Os gráficos, figura 3.16, são muito similares entre si (no sentido da distribuição), isto é, o tráfego Ethernet parece ter o mesmo aspecto em várias escalas de tempo.

Definição Matemática de Processos Auto-Similares

Seja X um processo estocástico estacionário, representando o número de pacotes por unidade de tempo, que possui uma função autocorrelação $r(k)$ que decai hiperbolicamente e que exhibe *dependência de longo alcance* definida pela seguinte fórmula:

$$r(k) \cong |k|^{-\beta} \quad (3.4)$$

onde $k \rightarrow \infty$, $0 < \beta < 1$

Seja:

$$X^{(m)} = X_k^{(m)}$$

para ($k = 1, 2, 3, \dots$) um novo processo estocástico obtido tomando a média aritmética da série original $X^{(m)}$ em blocos de tamanho m que não se sobreponham, isto é,

$$X_k^{(m)} = \frac{1}{m} \sum_{i=km-(m-1)}^{km} x_i$$

Para cada m , $X^{(m)}$ define um novo processo estocástico estacionário.

O processo X é dito *exatamente auto-similar de segunda ordem* com *parâmetro de auto-similaridade (parâmetro de Hurst)* $H = 1 - (\beta / 2)$ se todos os processos agregados correspondentes $X^{(m)}$ tiverem a mesma função de autocorrelação [LEL 94],

$$R_x^{(m)}(k) = R_x(k), \quad \text{para todo } m = 1, 2, 3, \dots \text{ (} k = 1, 2, 3, \dots \text{)}$$

Considerando a definição de dependência de longo alcance (eq. (3.4)) e a relação $H = 1 - (\beta / 2)$, temos que, para um processo com dependência de longo alcance, $0.5 < H < 1$. Para valores maiores de H a autocorrelação $R_x(k)$ (eq. (3.4)) decai mais lentamente. O parâmetro de Hurst, H , fornece-nos a *intensidade* da dependência de longo alcance. Quanto maior for o valor de H mais forte é essa dependência. Os processos que só apresentam dependência de curto alcance possuem $H = 0.5$. Em medidas de tráfego realizadas [LEL 94, WIL 97], o parâmetro de Hurst estimado ficou entre 0.7 e 0.9.

4 Técnica desenvolvida

4.1 Método para avaliação de desempenho escolhido

O projeto, a análise e a implementação de redes são tarefas complexas e trabalhosas. Em geral, três soluções podem ser utilizadas para auxiliar na execução dessas tarefas: a experimentação com redes reais, a utilização de métodos analíticos e a simulação computacional. Cada uma dessas soluções possui vantagens e limitações, e embora não sejam mutuamente exclusivas, cada uma delas é melhor aplicável a uma determinada situação particular [JAI 91].

4.1.1 Simulação - Vantagens e Limitações

A simulação computacional é a mais flexível das soluções mencionadas. Entre as vantagens da simulação de redes está a modelagem realista de fontes de tráfego, a facilidade de realização de experimentos, a comparação de mecanismos de controle de tráfego, a modelagem com diferentes níveis de detalhes e escalas de tempo de observação, a estimação precisa de probabilidade de perda e atrasos de informação, etc. Porém, todas essas vantagens têm seu preço: a simulação computacional demanda um alto desempenho de processamento. Simulação é adequada para resolver problemas envolvendo questões de tempo, comportamentos de fluxos heterogêneos e estados da rede que podem acontecer e são importantes de se entender. A quantidade de elementos simulados aumenta a quantidade de eventos a serem tratados pelo simulador. Portanto, para viabilizar a execução da simulação em tempos razoáveis é necessário limitar a topologia de rede a ser simulada [JAI 91] [GON 2001].

Alguns produtos de simulação utilizam a técnica orientada a eventos estocásticos discretos. Estocástico significa randômico no tempo. Através de métodos matemáticos para gerar seqüência de números pseudo-randômicos, é possível imitar variações aleatórias que ocorrem em sistemas reais. A habilidade permitida pela randomização é uma das grandes vantagens da simulação. Eventos discretos significam que nós visualizamos o sistema com o tempo progredindo de um evento importante para outro, em vez das mudanças de tempo ocorrerem de forma contínua. Isso permite a representação computacional eficiente do tempo, sendo a grande vantagem da simulação orientada a eventos [JAI 91].

4.1.2 Simulação da rede ATM - Simulador ATM-HFC

Para a parte da simulação da rede ATM foi utilizado para implementar o modelo a ferramenta de simulação orientada a eventos ATM/HFC [GOL98] desenvolvida no *National Institute of Standards and Technology* (NIST) dos Estados Unidos da América, desde 1995. A ferramenta é composta por diferentes modelos de elementos, cada qual com seus atributos, que podem ser combinados para caracterizar determinadas configurações de rede que se deseja avaliar. A carga de serviço na simulação é provida por elementos modeladores que caracterizam diferentes tipos de aplicações geradoras de tráfego, permitindo arranjos de serviços CBR, VBR, ABR e UBR utilizando seus respectivos parâmetros descritores [GOL 98].

A escolha da ferramenta de simulação foi baseada no estudo feito no trabalho [GON 2001], onde foi identificada uma certa semelhança de objetivos, como estudo das mesmas métricas, e sendo, em alguns aspectos, quase uma complementação do trabalho [GON 2001]. Como as alterações feitas no simulador NIST no trabalho [GON 2001] são bastante afins com as necessárias para o presente trabalho, a escolha foi feita e outras alterações foram implementadas permitindo a sua utilização com os propósitos dessa dissertação. Todas alterações acrescidas estão relacionadas no anexo 1 desse texto.

Com o objetivo de ser utilizado como ferramenta de apoio para estudo de desempenho de redes ATM, esse produto é capaz de simular diferentes configurações de topologia e carga de tráfego obtendo estatísticas sobre estimativas de utilização de enlaces, taxa de perda de células e outras medidas de desempenho relevantes. Também é possível, através da inclusão de componentes, utilizar o simulador NIST ATM/HFC para avaliar o desempenho de novos mecanismos de controle de tráfego [GOL 98].

Conceitos do Simulador

O simulador é implementado por um conjunto de componentes que trocam eventos entre si para caracterizar a transmissão de células pela rede. O produto possui um gerenciador de eventos que providencia as funções gerais para o escalonamento dos eventos. Os eventos são organizados em uma fila pela ordem crescente de tempo de seu acontecimento (tempo previsto para o "disparo"). Portanto, a execução das ações de determinado evento acontece do primeiro para o último na fila. O gerenciador de eventos mantém um relógio interno que é alterado constantemente conforme o tempo do primeiro evento da fila. Dessa forma, o tempo desse relógio, após a execução de um evento, salta para o tempo do próximo evento na fila. Esse método caracteriza um simulador orientado a eventos [GON 2001].

O relógio interno da simulação utiliza uma unidade de tempo denominada *ticks* que podem representar, através da configuração do produto, qualquer grandeza de tempo em segundos. A conversão de *ticks* para segundos utilizada foi a padrão encontrada no produto, um *tick* é igual a 10 nanossegundos. O relógio interno é mantido em uma variável sem sinal com 8 *bytes*, o que possibilitaria, na teoria, a execução de simulações com tempo simulado de enorme grandeza¹. O tempo de execução computacional da simulação é geralmente maior que o tempo simulado. O tempo gasto para executar uma simulação aumenta conforme a quantidade de componentes e eventos que devem ser tratados na simulação [GON 2001].

Componentes do Simulador

Componentes são os blocos básicos de construção do simulador. O processo de simulação consiste em vários componentes que são combinados entre si para formarem o ambiente a ser simulado. Os tipos de componentes do NIST ATM/HFC estão agrupados em 4 classes básicas, conforme segue [GOL 98]:

- Comutadores ATM;

¹ Utilizando a unidade de ticks de 10 nanossegundos e sabendo que a quantidade máxima de ticks que podem ser representados por uma variável sem sinal de 8 bytes é dada por $2^{32} - 1$, o tempo simulado máximo é aproximadamente de 5.849,42 anos.

- Equipamentos Terminais dos usuários ou BTE;
- Aplicações ATM;
- Enlaces de transmissão.

Cada componente consiste de um programa em C onde estão definidas as ações que devem ser executadas para os eventos previstos para o componente em questão. Cada instância¹ de um componente tem sua própria estrutura de dados que é utilizada para armazenar informações que caracterizam o componente [GON 2001].

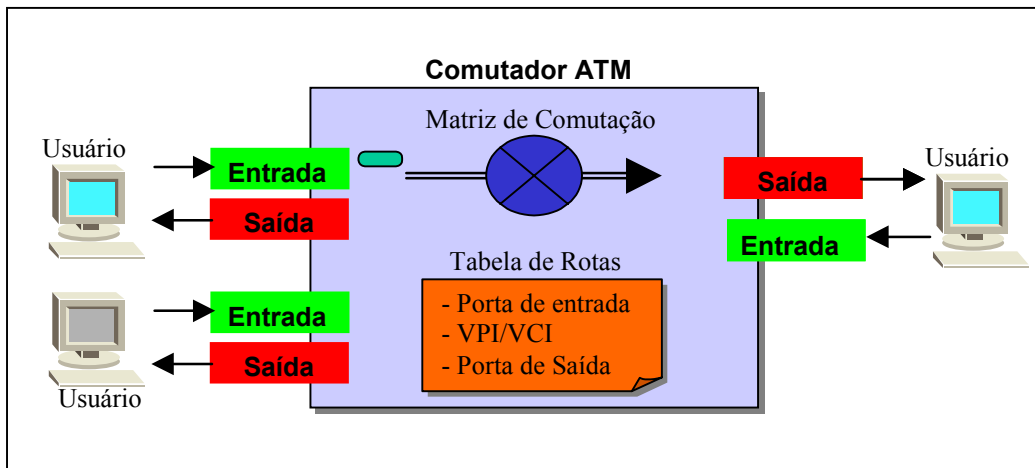


FIGURA 4.1 - Ações simuladas de um comutador ATM [GON 2001]

O funcionamento do comutador ATM é baseado no encaminhamento das células que chegam para as portas de saída. Cada comutador possui uma tabela local de roteamento contendo basicamente: uma identificação de conexão e o número da porta de saída. Para exemplificar o funcionamento da simulação de um comutador ATM, consideremos uma célula chegando a partir de um enlace de transmissão (fig. 4.1). No próximo período de tempo de comutação, após algum atraso² configurado no simulador, a célula é encaminhada para a porta de saída. O encaminhamento é baseado na identificação de conexão no cabeçalho da célula e na tabela de rotas do comutador. Se a porta de saída não possui um intervalo de tempo (*slot time*) para a transmissão da célula, ela é colocada em um fila de espera (buffer) para ser transmitida. O comutador do NIST ATM/HFC implementa uma estrutura de filas por grupo baseada em prioridade de atendimento conforme a categoria de serviço (fig. 4.2). A fila que recebe a maior prioridade no atendimento de suas células é a CBR/VBR, seguida pelas filas ABR e UBR, nessa ordem, significando que somente quando a fila CBR/VBR estiver vazia, a fila ABR será atendida. Portanto, a fila UBR que possui a menor prioridade só será atendida para transmitir suas células quando as outras duas filas estiverem vazias. Se o número de células na fila excede determinado valor configurado para controle de tráfego, um indicador de congestionamento é acionado para a porta. O indicador de congestionamento é desligado quando as três filas estiverem abaixo do valor predeterminado. O tamanho de cada fila (capacidade de armazenar células) é

¹ Terminologia de OOP (Programação Orientada a Objetos) que significa criar um objeto a partir de uma classe.

² Atraso interno simulado para caracterizar o atraso fixo de comutação de células que todo comutador possui intrinsecamente.

configurado para cada comutador. Ultrapassando seu limite de armazenamento, as células excedentes são descartadas e contabilizadas para cada fila na porta [GON 2001].

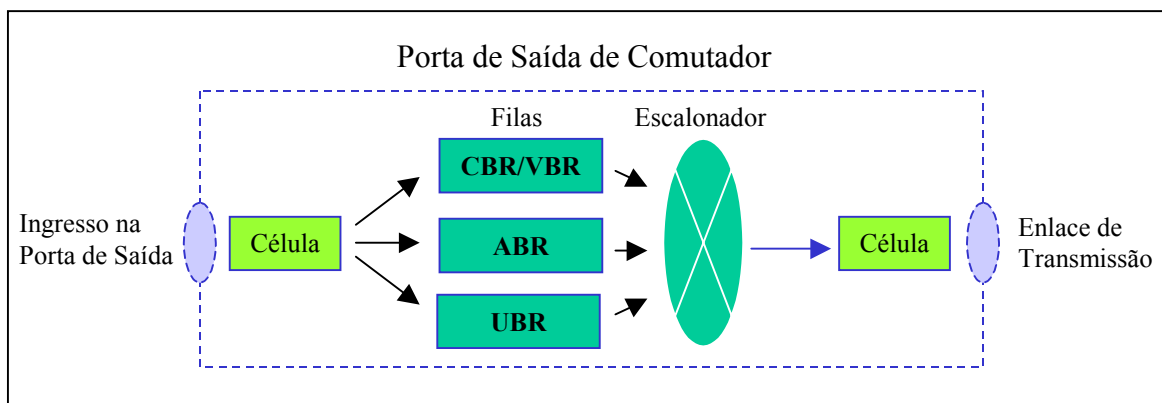


FIGURA 4.2 - Estrutura de filas e escalonador por prioridade de categoria de serviço [GON 2001]

A classe de equipamentos terminais de usuários simula dispositivos físicos que requisitam conexões da rede como computadores, centrais telefônicas, codificadores de sinais de vídeo, e outros. Sobre cada equipamento são executadas uma ou mais aplicações [GOL 98].

As ações simuladas dos equipamentos terminais são: receber as células das aplicações e enviá-las pelo enlace de transmissão (fig. 4.3). Se a porta de saída não possui um intervalo de tempo (*slot time*) para a transmissão da célula, essa é colocada em uma das três filas de prioridade (CBR/VBR, ABR ou UBR). O tamanho máximo de cada fila de saída é configurável e, da mesma forma que nos comutadores, as células que excederem o limite são descartadas [GOL 98].

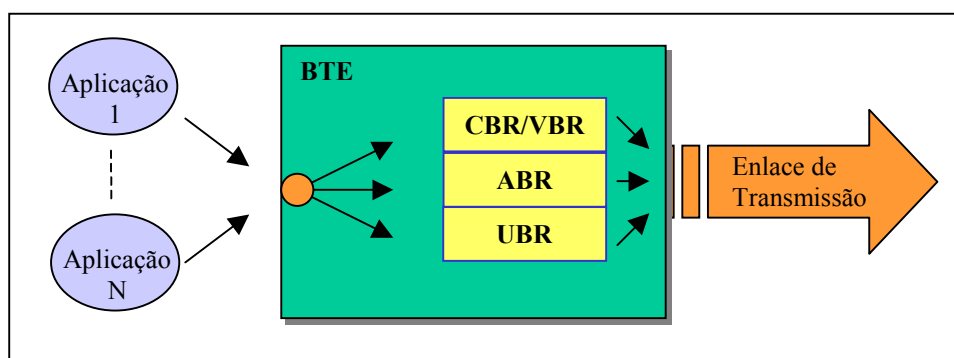


FIGURA 4.3 - Simulação do acesso dos equipamentos terminais à rede [GON 2001]

A classe de aplicações ATM tem a função de gerar o tráfego de células sobre os equipamentos terminais. A fonte de tráfego gerada por esses componentes pode ser basicamente constante ou variável. Porém, com a utilização de diferentes métodos de geração de tráfego, essa classe totaliza 12 diferentes tipos de componentes. Os métodos de geração de tráfego podem ser agrupados em: Constante, Poisson, Batch, Auto-similar, TCP/IP e MPEG. Detalhes sobre os métodos e suas utilizações podem ser encontrados em [GOL98].

A última classe de componentes que resta para completar o ambiente simulado é

a de enlaces de transmissão. Essa classe possui apenas um tipo de componente. Esse componente simula o meio físico por onde as células são transmitidas de um equipamento para outro. A caracterização desses componentes permite considerar a taxa de transmissão do enlace e o comprimento do enlace para cálculo dos atrasos de transmissão e propagação, respectivamente [GOL 98].

4.1.3 Simulação da rede IP - Simulador NS-2

Para a parte da simulação da rede IP foi utilizado para implementar o modelo a ferramenta de simulação orientada a eventos NS-2 [CLA 2001] desenvolvida pela UC Berkeley, acrescido dos módulos para RSVP (RSVP/ns) desenvolvido por Marc Greis da Universidade de Bonn [GRE 2001], sendo que atualmente, apenas o serviço de carga controlada é suportado pelo RSVP/ns e as garantias de largura de banda são forçadas por WFQ nos enlaces [GRE 2001].

NS começou como uma variante do simulador de redes REAL em 1989 e tem evoluído substancialmente ao longo dos últimos anos. Em 1995 o desenvolvimento do NS estava sendo suportado pelo DARPA através do projeto VINT da LBL, Xerox PARC, UCB, and USC/ISI. Atualmente o desenvolvimento do NS é suportado via DARPA com SAMAN e via NSF com CONSER, ambos em colaboração com outros pesquisadores incluindo ACIRI. NS tem incluído substanciais contribuições de outros pesquisadores, incluindo os projetos UCB Daedalus and CMU Monarch e a Sun Microsystems.

Conceitos e componentes do NS-2

NS (versão 2) é um simulador com orientação a objetos, escrito em C++ e Otcl, que permite uma variedade de simulações em redes IP. Ele implementa protocolos de redes tais como TCP e UDP, comportamentos de fontes de tráfegos de aplicações simuladas tais como FTP e Telnet, Web ou fontes de tráfegos gerados de acordo com alguma distribuição tais como exponencial, pareto ou de acordo com um arquivo de trace, e mecanismos de gerenciamento de filas dos roteadores tais como DropTail e RED entre outros [CLA 2001].

A figura 4.4 ilustra uma visão simplificada de como os seus elementos se relacionam.

Para usar o NS, deve-se se programar na linguagem script OTcl. Para inicializar e executar uma simulação, um usuário deve escrever um script na linguagem OTcl que iniciará o escalonador de eventos [CLA 2001].

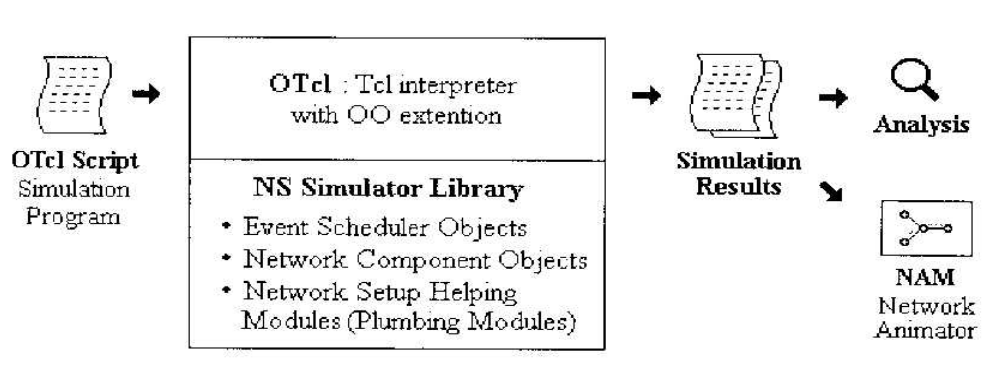


FIGURA 4.4 - NS-2: visão simplificada [CLA 2001]

4.1.4 Validação

Validação do modelo refere-se a garantia de que as hipóteses utilizadas para o desenvolvimento do modelo são razoáveis e que, se corretamente implementadas, o modelo poderá produzir resultados muito próximos aos observados no mundo real [JAI 91].

As hipóteses, os dados de entrada e os dados de saída são os três aspectos que devem ser validados. As seguintes possibilidades são utilizadas para validar esses aspectos [JAI 91]:

1. Validação por intuição de especialistas;
2. Validação por experimentação real;
3. Validação matemática.

Porém, no caso dessa dissertação, como o modelo utilizado nas simulações é extremamente simples, por essa razão não estamos utilizando nenhum tipo de validação acima referida.

Quanto a validação das ferramentas utilizadas, os simuladores NIST e NS-2, estamos considerando que ambos simuladores são bastante conhecidos e reconhecidos no mundo acadêmico não necessitando de nenhum tipo de validação.

4.2 Método da Simulação

Esse trabalho apresenta uma análise comparativa de desempenho em redes ATM e IP que pode ser utilizado como apoio ao projeto ou manutenção de rede. Está baseado na avaliação de desempenho obtida pela simulação, orientada a eventos, do comportamento de tráfego tradicional acrescido do tráfego multimídia sobre a tecnologia ATM e IP com reserva de recursos via RSVP utilizando serviço com carga controlada. O modelo utilizado na simulação é caracterizado por elementos fundamentais que atuam no desempenho dos serviços, tais como: tamanho do *buffer*, modelo de tráfego, quantidade de conexões e largura de banda do enlace.

As medidas de desempenho ou métricas obtidas nas simulações são valores estatísticos médios:

- da razão de perda de células na rede ATM e pacotes na rede IP (CLR);
- do atraso de transferência de células na rede ATM e pacotes na rede IP (CTD);
- e da variação do atraso de células na rede ATM e pacotes na rede IP (CDV).

Para abranger as duas fases de uma rede, a de projeto e a de manutenção, foram elaborados dois conjuntos de cenários:

- com banda variável (BV);
- com banda fixa (BF).

Considerou-se o primeiro caso, BV, pertinente para uma rede em fase de projeto, pois como a rede propriamente dita ainda não está operacional, não existe ainda definição rígida quanto à capacidade do enlace, tamanho dos *buffers*, podendo assim, experimentar inúmeras execuções variando seus recursos, visualizando o desempenho em cada uma das tecnologias simuladas referente às métricas acima citadas e ainda conseguir estabelecer um melhor aproveitamento destes recursos exercitados. Nessa dissertação, a variação na capacidade do enlace foi feita baseada nos valores apresentados em [GON 2001] e colocados na tabela 4.1. Em [GON 2001] é apresentado um método para determinação da banda efetiva que satisfaz o QoS dos fluxos VBR agregados. Sabendo-se a banda efetiva utilizada pelos fluxos multimídia, e variando a largura do enlace nos experimentos simulados em função da taxa de pico (PCR agregado) destas mesmas aplicações, tem-se a sobra de banda para outros tipos de fluxos, e que, nessa dissertação, é o fluxo auto-similar que representa o tráfego tradicional de uma rede (bulk traffic). Assim podemos prever o seu comportamento e antecipar quanto de tráfego será possível compartilhar na rede modelada e ainda manter os seus níveis de QoS.

TABELA 4.1 - Alocação de Recursos para Serviço de Vídeo Conferência1

Conexões Agregadas	PCR Agregado ² (bit/s)	Banda Alocada ³ (bit/s)	Ganho de Multiplexação Estatística
1	1.160.967,50	1.160.967,50	1,00
2	2.321.935,00	1.915.596,38	1,21
3	3.482.902,50	2.263.886,63	1,54
4	4.643.870,00	2.554.128,50	1,82
5	5.804.837,50	2.902.418,75	2,00
6	6.965.805,00	3.134.612,25	2,22
8	9.287.740,00	3.715.096,00	2,50
10	11.609.675,00	4.353.628,13	2,67
15	17.414.512,50	5.659.716,56	3,08
25	29.024.187,50	7.981.651,56	3,64

Fonte: [GON 2001]

Para o segundo caso, BF, considerou-se pertinente para a fase de manutenção de rede. Saber antes do acréscimo de serviços, como será o comportamento da rede evitando possíveis incomodações futuras para os usuários da mesma e poder exercitar cenários de crescimento para a mesma, sem prejuízo para os futuros clientes. Nessa dissertação, a largura fixa de enlace escolhida foi de 10Mbit/s, por ser maior que o PCR agregado nos 3 primeiros cenários (5, 6 e 8 conexões agregadas) e menor que o PCR agregado no último cenário (com 10 conexões agregadas), conforme mostra a tabela 4.1, podendo-se, desta forma, chegar aos piores casos e assim verificar como cada tecnologia se comporta frente a situações de alta carga ou saturação do canal.

Como resultado das simulações são apresentados gráficos com três curvas de desempenho em função do acréscimo no número de conexões VBR simultâneas, tráfego multimídia, e do fator tamanho do *buffer*, que variará entre 100, 1.000 e 10.000 células ou pacotes. Diferentes configurações de redes e cargas de serviços podem ser exercitadas para revelar limites necessários de alocação de recursos que evitem dificuldades de desempenho. A figura 4.5 mostra o exemplo de um cenário de simulação para três determinadas conexões no qual desejamos descobrir se para uma determinada quantidade de banda disponível, é mantido os níveis de QoS de todas as aplicações que estão compartilhando o enlace.

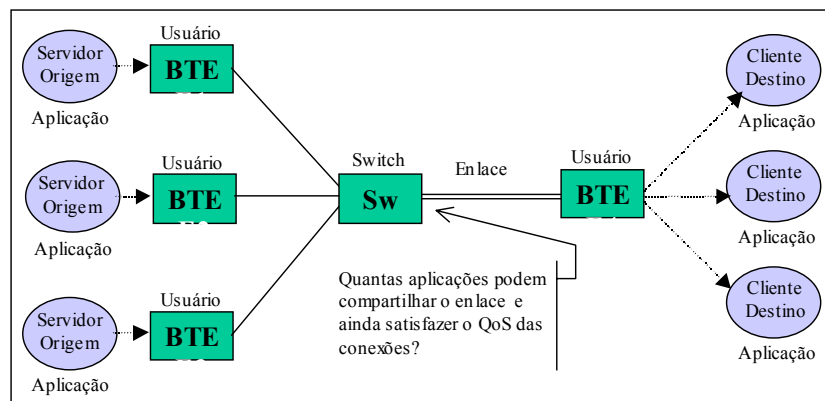


FIGURA 4.5 - Cenário de Simulação

¹ Serviço de vídeo conferência descrito conforme tabela 4.6 dessa dissertação.

² Valores utilizados como largura de banda para os cenários de banda variável.

³ Valores utilizados para as reservas nas simulações com NS-2.

4.3 Cenários simulados

Nessa dissertação foram elaborados dois conjuntos de cenários, um com banda variável e outro com banda fixa. Cada conjunto foi simulado em ambos simuladores, NIST e NS-2.

No primeiro caso, banda variável (BV), foram simulados 10 cenários, com a variação do fator tamanho do buffer entre 100, 1.000 e 10.000 células/pacotes. Conforme explicado no capítulo 2 desse trabalho, cada cenário foi replicado 10 vezes para fins estatísticos, o que computou um total de 300 simulações para esse conjunto em cada um dos simuladores¹. A largura de banda desses cenários variou conforme o PCR agregado do serviço de vídeo conferência apresentado em [GON 2001] e mostrado na tabela 4.1. De acordo com o referido trabalho, a banda alocada por esse serviço é inferior a sua taxa de pico, ocasionado sobra de banda para utilização de outro serviço, permitindo a multiplexação estatística das fontes e um melhor aproveitamento dos recursos da rede. Com esse conjunto de cenários pode-se verificar o quão bem isso ocorre em cada uma das tecnologias simuladas nessa dissertação.

No segundo caso, banda fixa (BF), foi simulado 4 cenários, com a mesma variação do fator tamanho do buffer entre 100, 1.000 e 10.000 células/pacotes. Também da mesma forma foram feitas 10 replicações para cada cenário, o que totalizou 120 simulações em cada um dos simuladores¹. Nesse caso, a largura de banda escolhida foi de 10Mbit/s, por ser maior que o PCR agregado nos 3 primeiros cenários (5, 6 e 8 conexões VBR simultâneas) e menor que o PCR agregado no último cenário (com 10 conexões VBR simultâneas). Esse conjunto de cenários tem como objetivo a visualização das possibilidades de crescimento de uma rede já existente, com recursos definidos, permitindo antecipar o provável desempenho frente a necessidade de expansão na carga de tráfego na rede.

Em todas as simulações foram utilizadas uma fonte de tráfego auto-similar, que representa a carga de uma rede Ethernet tradicional, com suas transferências de arquivos, serviço de correio eletrônico, serviço WWW, etc. No caso no simulador NIST, já estava implementada esse tipo de fonte, sendo somente necessário fazer as adaptações para os cálculos das métricas de interesse desse trabalho, que estão descritos no anexo 1. No caso do simulador NS-2, foi utilizado um gerador de tráfego auto-similar² com os mesmos parâmetros utilizados no NIST, conforme tabela 4.5.

A fonte de tráfego que representa uma vídeo conferência é descrita conforme tabela 4.4 em ambos simuladores, sendo que o número de fontes vai variando em cada um dos cenários simulados, sendo na verdade outro fator de variação das simulações desse trabalho.

A tabela 4.2 mostra cenários simulados para o primeiro caso citado, banda variável.

A tabela 4.3 mostra os cenários para o segundo caso citado, banda fixa.

¹ Para verificação do melhor tamanho de pacote a ser utilizado no simulador NS-2, foi gerado mais uma variação para estes mesmos conjuntos de cenários. Foi executado 600 simulações para o primeiro caso: 300 com pacote com 240 bytes e mais 300 com pacote de 48 bytes e foi executado 240 simulações para o segundo caso, sendo 120 para pacotes de 240 bytes e 120 para pacotes com 48 bytes. Ao final, verificou-se que os valores ficavam muito próximos, sendo escolhido para a apresentação dos dados, os resultados com tamanho de pacote de 48 bytes, mesma carga útil de uma célula ATM.

² Encontrado em http://www.csif.cs.ucdavis.edu/~kramer/code/trf_gen2.html

O computador utilizado nas execuções das simulações tem a seguinte configuração: processador Intel Pentium-II 333 MHz, 128 MB, sistema operacional Linux Conectiva version 2.2.17-14cl e GNU bash, version 2.04.0(1) - release (i386-conectiva-linux-gnu).

TABELA 4.2 - Cenários Simulados - Banda Variável (PCR Agregado)

Descrição dos fluxos	Cenário	Conexões VBR	Buffers (células/pacotes)
1 Auto-similar + 1 Vídeo Conferência	1	1	100
1 Auto-similar + 1 Vídeo Conferência	2	1	1.000
1 Auto-similar + 1 Vídeo Conferência	3	1	10.000
1 Auto-similar + 2 Vídeo Conferências	4	2	100
1 Auto-similar + 2 Vídeo Conferências	5	2	1.000
1 Auto-similar + 2 Vídeo Conferências	6	2	10.000
1 Auto-similar + 3 Vídeo Conferências	7	3	100
1 Auto-similar + 3 Vídeo Conferências	8	3	1.000
1 Auto-similar + 3 Vídeo Conferências	9	3	10.000
1 Auto-similar + 4 Vídeo Conferências	10	4	100
1 Auto-similar + 4 Vídeo Conferências	11	4	1.000
1 Auto-similar + 4 Vídeo Conferências	12	4	10.000
1 Auto-similar + 5 Vídeo Conferências	13	5	100
1 Auto-similar + 5 Vídeo Conferências	14	5	1.000
1 Auto-similar + 5 Vídeo Conferências	15	5	10.000
1 Auto-similar + 6 Vídeo Conferências	16	6	100
1 Auto-similar + 6 Vídeo Conferências	17	6	1.000
1 Auto-similar + 6 Vídeo Conferências	18	6	10.000
1 Auto-similar + 8 Vídeo Conferências	19	8	100
1 Auto-similar + 8 Vídeo Conferências	20	8	1.000
1 Auto-similar + 8 Vídeo Conferências	21	8	10.000
1 Auto-similar + 10 Vídeo Conferências	22	10	100
1 Auto-similar + 10 Vídeo Conferências	23	10	1.000
1 Auto-similar + 10 Vídeo Conferências	24	10	10.000
1 Auto-similar + 15 Vídeo Conferências	25	15	100
1 Auto-similar + 15 Vídeo Conferências	26	15	1.000
1 Auto-similar + 15 Vídeo Conferências	27	15	10.000
1 Auto-similar + 25 Vídeo Conferências	28	25	100
1 Auto-similar + 25 Vídeo Conferências	29	25	1.000
1 Auto-similar + 25 Vídeo Conferências	30	25	10.000

TABELA 4.3 - Cenários Simulados - Banda Fixa (10Mbit/s)

Descrição dos fluxos	Cenário	Conexões VBR	Buffers (células/pacotes)
1 Auto-similar + 5 Vídeo Conferências	1	5	100
1 Auto-similar + 5 Vídeo Conferências	2	5	1.000
1 Auto-similar + 5 Vídeo Conferências	3	5	10.000
1 Auto-similar + 6 Vídeo Conferências	4	6	100

1 Auto-similar + 6 Vídeo Conferências	5	6	1.000
1 Auto-similar + 6 Vídeo Conferências	6	6	10.000
1 Auto-similar + 8 Vídeo Conferências	7	8	100
1 Auto-similar + 8 Vídeo Conferências	8	8	1.000
1 Auto-similar + 8 Vídeo Conferências	9	8	10.000
1 Auto-similar + 10 Vídeo Conferências	10	10	100
1 Auto-similar + 10 Vídeo Conferências	11	10	1.000
1 Auto-similar + 10 Vídeo Conferências	12	10	10.000

4.4 Valores Iniciais

O simulador NIST ATM/HFC utilizado em [GON 2001], foi modificado¹ para apresentar os resultados de CTD, CDV e CLR utilizando o método de divisão em subintervalos no gerador de tráfego auto-similar. As alterações atuaram no cálculo e na forma de apresentação dos resultados. A parte central e principal do simulador responsável pela geração de tráfego e tratamento de eventos não sofreu modificações, permanecendo conforme a versão original.

O gerador de fontes utilizado para modelar o tráfego de aplicações com requisitos temporais foi o VBR Poisson. Os períodos de atividade/silêncio (*on/off*) do tráfego são exponencialmente distribuídos através dos valores médios informados. A escolha do VBR Poisson nos experimentos é devido à simplicidade de caracterização das fontes, no entanto outros geradores de tráfego disponíveis no NIST ATM também podem ser utilizados, assim como no NS-2. Utilizamos exemplos de aplicações de transmissão de vídeo (tab. 4.4). Os dados são baseados em medições realizadas pelo Instituto de Ciência da Computação de Würzburg (Alemanha) usando um codificador MPEG-I com as seguintes características: 25 quadros por segundo, quadro de 384 por 288 *pixels* com 12 bits para informação de cor e 40.000 quadros observados por evento (aproximadamente 30 minutos de vídeo).

TABELA 4.4 - Serviço VBR simulado

Serviço	Evento	Tamanho Médio do Quadro (bit)	Relação Pico/Média
Vídeo Conferência	Desktop	6.031	7.7

Fonte: [ROB96]

O gerador de fontes utilizado para modelar o tráfego de aplicações elásticas foi o Auto-similar. No NIST este gerador já existia. No NS, como esta aplicação não está disponível, então foi utilizada uma aplicação do tipo TraceFile que utiliza um trace gerado por um gerador de tráfego externo com os mesmos parâmetros do gerador utilizado no NIST, descrito na tabela 4.5.

¹ As modificações feitas nos programas que compõem o NIST ATM/HFC estão relacionadas no anexo 1.

TABELA 4.5 - Serviço Auto-similar simulado

Serviço	Taxa	Hurst
Tráfego Auto-similar, tais como, ethernet, web, ftp, ...	10 Mbit/s	0.8

4.4.1 Caracterização do tráfego VBR para os simuladores

O gerador de tráfego VBR Poisson exponencialmente distribuídos necessita três parâmetros para caracterização do tráfego:

- PCR, dado em Mbit/s;
- Tamanho médio da rajada, dado em microssegundos (μ s);
- Tamanho médio do intervalo entre rajadas, dado em μ s.

Para converter as informações da tabela 4.4 para os parâmetros mencionados anteriormente, devemos utilizar os cálculos a seguir.

Informação Inicial:

- TQM - Tamanho Quadro Médio, dado em bits;
- TQ - Taxa de Quadros, dado em quadros/s
- PM - Razão Pico/Média.

Calculamos então,

$$TTM = TQM \times TQ$$

$$PCR = TTM \times PM$$

$$Ton = \frac{TQM}{PCR}$$

$$Toff = \frac{1}{TQ} - Ton$$

Onde,

- TTM é a taxa de transmissão média, dada em bit/s;
- TQM é o tamanho médio do quadro;
- TQ é a taxa de transmissão de quadros por segundo;
- PCR é a taxa de transmissão de pico, dada em bit/s;
- PM é a razão entre a taxa máxima de transmissão e a taxa média;
- Ton é o tempo médio de duração da rajada (transmissão na taxa de pico);
- $Toff$ é o tempo médio entre as rajadas (inatividade da transmissão).

Dessa forma, temos as informações necessárias para gerar o tráfego com o

modelo VBR Poisson exponencialmente distribuídos. Seguem os valores calculados para o serviço utilizado nessa dissertação.

TABELA 4.6 - Parâmetros Descritores VBR Poisson

Serviço	Evento	PCR (bit/s)	Ton (μ s)	Toff (μ s)
Vídeo Conferência	Desktop	1.160.967,5	5.190	34.810

4.5 Métricas utilizadas

Devido à maneira diferente de processamento das informações geradas em cada um dos dois simuladores envolvidos nessa dissertação, o procedimento dos cálculos das métricas de interesse foi feito em momentos diferentes, sendo que se manteve a mesma fórmula de cálculo.

No caso do NIST, para rede ATM, as informações são processadas durante a própria simulação permitindo que no final de cada execução, já se tenha em mãos dados médios de perda, atraso e variação do atraso. Para a apresentação gráfica dos resultados finais dessas simulações foram desenvolvidos alguns scripts apresentados no anexo 2.

No caso do NS, para rede IP, as informações precisam ser processadas após o término de cada execução do simulador, pois apenas é gerado um trace contendo todos os eventos ocorridos a cada instante na simulação. Nesse caso, para padronização dos cálculos e formatação dos resultados, foi utilizado um software feito em JAVA¹ para analisar os dados desse trace de saída. Posteriormente, para manipulação desses dados e geração dos gráficos finais dessas simulações foram desenvolvidos alguns scripts apresentados no anexo 2.

4.5.1 Estimativa de CTD - Atraso

O atraso para a transferência de células/pacotes da origem até o destino da conexão é calculado como a diferença entre o tempo de chegada e o tempo de partida da célula/pacote. Portanto, o cálculo é feito na chegada de cada célula/pacote conforme a expressão:

$$CTD_y = T_{chegada} - T_{partida}$$

No final de cada subintervalo da simulação², é calculada a média de CTD entre todas as células/pacotes recebidas(os) na conexão, conforme expressão:

$$\overline{CTD}_{sub_x} = \frac{\sum_{y=1}^m CTD_y}{m}$$

¹ Software em JAVA desenvolvido e cedido por Luis Fernando Pollo, aluno do PGCC.

² Cada subintervalo possui valor de 500 milisegundos conforme motivos apresentados em [GON 2001], sendo que os mais significativos são: o tipo de fluxo ser variável ao longo do tempo, o número de amostras para o cálculo do IC e a limitação do próprio simulador NIST.

Onde,

\overline{CTDsub}_x é a média do CTD do subintervalo "x";
 CTD_y é o valor do CTD para a célula/pacote "y";
 m é a quantidade de células/pacotes recebidas no subintervalo "x".

No final da simulação após "n" subintervalos, é calculado o CTD médio para a replicação em questão, conforme expressão:

$$\overline{CTD rep}_z = \frac{\sum_{x=1}^n \overline{CTD sub}_x}{n}$$

Onde,

\overline{CTDrep}_z é o CTD médio para a replicação "z", sendo que z varia de 1 a 10;
 n é o número de subintervalos da simulação.

Para calcular o CTD final da simulação, para cada cenário proposto, consideramos a média entre todas as replicações, conforme expressão:

$$CTD_{final} = \frac{\sum_{z=1}^p \overline{CTDrep}_z}{p}$$

Onde,

CTD_{final} é o CTD médio entre todas as replicações estabelecidas na execução do simulador para um determinado cenário;
 p é o número de replicações feitas.

Também no final da simulação, é calculado o intervalo de confiança para o CTD médio final utilizando as seguintes expressões:

$$s^2 = \frac{n \sum_{i=1}^n (\overline{CTD rep}_i)^2 - \left(\sum_{i=1}^n \overline{CTD rep}_i \right)^2}{n(n-1)}$$

$$s = \sqrt{s^2}$$

Onde,

\overline{CTDrep}_i é a média do CTD da replicação "i";
 n é o número de amostras de CTD. Cada amostra de CTD corresponde ao valor calculado em uma replicação;
 s^2 é a variância das amostras de CTD calculado em cada replicação;
 s é o desvio padrão das amostras de CTD calculado em cada replicação.

O intervalo de confiança para o valor final é dado pelas seguintes expressões:

$$\text{Erro}_z = 3.25 * \frac{s}{\sqrt{n}}$$

$$IC_z = \left[\overline{CTD}_{\text{final}_z} - \text{Erro}_z, \overline{CTD}_{\text{final}_z} + \text{Erro}_z \right]$$

Onde,

$\overline{CTD}_{\text{final}_z}$ é o CTD médio para o cenário "z";

IC_z é o intervalo de confiança do cenário "z";

Erro_z é o valor que determina o limite superior e inferior do intervalo de confiança para o cenário "z". A constante "3.25" da expressão refere-se à probabilidade de 99 % de acerto com 9 graus de liberdade de que o CTD estará dentro dos limites do intervalo de confiança (tabela da distribuição $t_{[0.995,9]}$);

s é o desvio padrão das amostras de CTD calculado em cada replicação;

n é o número de replicações para cada cenário.

4.5.2 Estimativa de CDV - Variação do Atraso

O tempo de permanência em filas (*buffers*) varia de uma célula/pacote para outra(o). Essa variação é referida como CDV (*Cell Delay Variation*), também chamada de *jitter*. Um fluxo de células/pacotes com taxa de transmissão constante chegando em uma fila congestionada para transmissão pode resultar em uma seqüência de rajadas de células/pacotes na saída.

O CDV tem um impacto negativo no desempenho das aplicações que não se adaptam bem às modificações na taxa recebimento das células/pacotes. Por exemplo, um decodificador que está mostrando um vídeo com 30 quadros por segundo espera receber da rede um fluxo consistente. Entretanto, se a taxa de recebimento decrementa, o decodificador deverá reduzir a quantidade de quadros por segundo, prejudicando a qualidade do vídeo. A tabela 4.7 mostra valores de atraso e de variação de atraso tolerados para diferentes aplicações [GON 2001].

TABELA 4.7 - Exemplos de Requisitos de CTD e CDV

Aplicação	Taxa de Pico	Atraso (CTD)	Variação do Atraso (CDV)
Voz Comprimida	16 kbit/s	30 ms	130 ms
Vídeo Conferência	64 kbit/s	250 ms	130 ms
Vídeo Conf. MPEG	256 kbit/s	7 ms	9,1 ms
Vídeo NTSC MPEG	1,5 Mbit/s	5 ms	6,5 ms
Vídeo HDTV	20 Mbit/s	0,8 ms	1 ms

Fonte: [ONV 95]

Conceitualmente, o CDV é dado como a diferença entre o atraso experimentado por duas células/pacotes consecutivas(os), como mostra a expressão:

$$CDV = (D + W_{i+1}) - (D + W_i)$$

Onde,

D é a constante que inclui os atrasos de propagação, transmissão e comutação. É um valor igual para todas as células/pacotes da conexão em questão.

W_i é o valor de atraso resultante das filas por onde passou a célula/pacote "i" da conexão.

W_{i+1} é um componente do atraso variável da célula/pacote que segue a de número "i". É um valor independente do W_i .

A variação do atraso para a transferência de células/pacotes da origem até o destino da conexão é baseada no método de CDV de dois pontos que é dado pela seguinte expressão:

$$CDV2P_y = |CTD_y - CTDref|$$

Onde,

$CDV2P_y$ é o CDV de dois pontos da célula/pacote "y". É sempre um valor positivo;

CTD_y é o valor do CTD para a célula/pacote "y";

$CTDref$ é o CTD de referência que é definido como sendo o CTD calculado para a primeira célula/pacote da conexão.

O método de CDV de dois pontos utiliza como $CTDref$ o atraso da primeira célula/pacote da conexão. O mais adequado seria utilizar como $CTDref$ a média de CTD da conexão e assim medir a variação em torno desse valor padrão. Como o cálculo do CDV é feito célula/pacote a célula/pacote, não podemos utilizar a média como referência pois não temos como calculá-la antecipadamente. Entretanto, como dividimos a simulação em subintervalos, podemos utilizar o CTD médio do subintervalo anterior como $CTDref$ para o subintervalo corrente. Dessa forma, a expressão para o cálculo do CDV para cada célula/pacote que chega é a seguinte:

$$CDV2P_y = |CTD_y - \overline{CTDsub_{i-1}}|$$

Onde,

$CDV2P_y$ é o CDV de dois pontos da célula/pacote "y". É sempre um valor positivo;

CTD_y é o valor do CTD para a célula/pacote "y";

$CTDsub_{i-1}$ é a média do CTD do subintervalo "i-1";

i é o número do subintervalo corrente.

Baseado nessa expressão, o simulador não calcula o CDV para as células/pacote do primeiro subintervalo. Nos subintervalos subsequentes, o CDV é calculado normalmente conforme a expressão acima.

No final da simulação, após "n" subintervalos, é calculado o CDV médio para a replicação, conforme expressão:

$$\overline{CDVrep}_z = \frac{\sum_{x=1}^n CDVsub_x}{n}$$

Onde,

\overline{CDVrep}_z é o CDV médio para a replicação "z";
 n é o número de subintervalos da simulação.

Para calcular o CDV final da simulação consideramos a média entre todas as replicações, conforme expressão:

$$CDV_{final} = \frac{\sum_{z=1}^p \overline{CDVrep}_z}{p}$$

Onde,

CDV_{final} é o CDV médio entre todas as replicações estabelecidas na execução do simulador;
 p é o número de replicações feitas.

Também no final da simulação, é calculado o intervalo de confiança para o CDV final utilizando as seguintes expressões:

$$s^2 = \frac{n \sum_{i=1}^n (\overline{CDVrep}_i)^2 - \left(\sum_{i=1}^n \overline{CDVrep}_i \right)^2}{n(n-1)}$$

$$s = \sqrt{s^2}$$

Onde,

$CDVrep_i$ é o CDV da replicação "i";
 n é o número de amostras de CDV. Cada amostra de CDV corresponde ao valor calculado em uma replicação;
 s^2 é a variância das amostras de CDV calculado em cada replicação;
 s é o desvio padrão das amostras de CDV calculado em cada replicação.

O intervalo de confiança para o valor final é dado pelas seguintes expressões:

$$Erro_z = 3.25 * \frac{s}{\sqrt{n}}$$

$$IC_z = \left[\overline{CDVfinal}_z - Erro_z, \overline{CDVfinal}_z + Erro_z \right]$$

Onde,

\overline{CDV}_{final_z} é o CDV médio para o cenário "z";

IC_z é o intervalo de confiança do cenário "z";

$Erro_z$ é o valor que determina o limite superior e inferior do intervalo de confiança para o cenário "z". A constante "3.25" da expressão refere-se à probabilidade de 99 % de acerto com 9 graus de liberdade de que o CDV estará dentro dos limites do intervalo de confiança (tabela da distribuição $t_{[0,995,9]}$);

s é o desvio padrão das amostras de CDV calculado em cada replicação;

n é o número de replicações para cada cenário.

O resultado final apresentado pelo simulador inclui o intervalo de confiança para o CDV (IC_{final}). A figura 4.6 mostra um resumo da seqüência de cálculos utilizados pelo simulador para estimar tanto o CTD como o CDV.

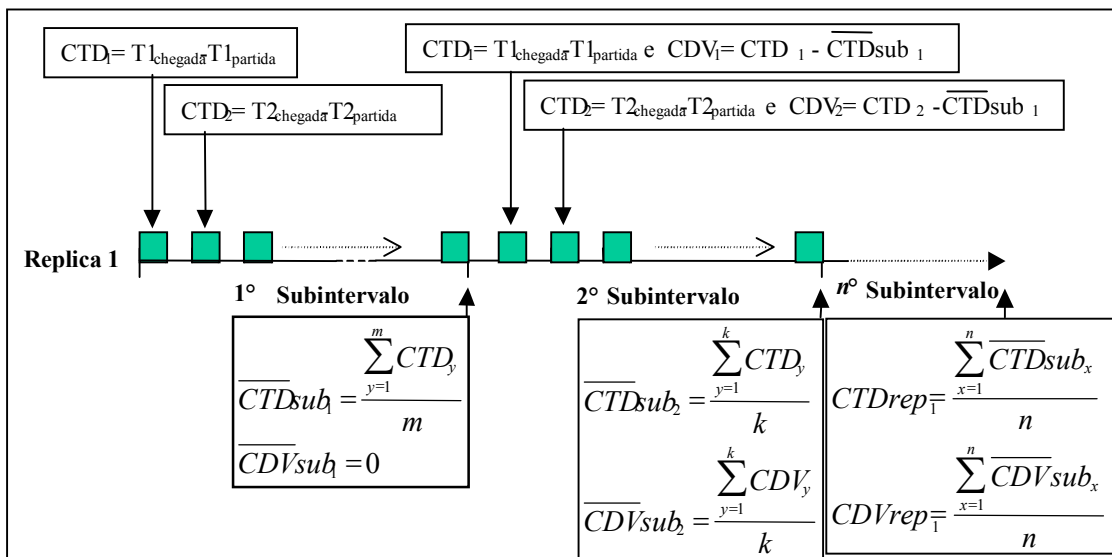


FIGURA 4.6 - Estimativa do CTD e CDV para uma replicação [GON 2001]

4.5.3 Estimativa de CLR - Perda/Descarte

A razão de células/pacotes perdidas(os) de uma conexão é dada por:

$$CLR = \frac{Cel_Desc}{Cel_Env}$$

Onde,

Cel_Desc é a quantidade de células/pacotes que foram descartadas por transbordo de buffer no comutador/roteador;

Cel_Env é a quantidade de células/pacotes enviadas pela fonte geradora do fluxo.

O intervalo de tempo de medição não é padronizado, mas geralmente é representando como o período de duração da conexão. Seguindo o método de divisão da simulação em subintervalos, utilizamos o tempo de duração do subintervalo como

período de contabilização das células/pacotes descartadas(os) e das células/pacotes enviadas(os). Portanto, no final de cada subintervalo da simulação é feito o cálculo da razão de perda de células/pacotes. Segue expressão utilizada:

$$CLR_{sub_x} = \frac{Cel_Desc}{Cel_Env}$$

Onde,

CLR_{sub_x} é a razão de perda de células/pacotes do subintervalo "x";

Cel_Desc é a quantidade de células/pacotes descartadas no subintervalo "x";

Cel_Env é a quantidade de células/pacotes enviadas no subintervalo "x".

A quantidade de células/pacotes geradas(os) depende do tamanho do subintervalo utilizado (nessa dissertação 500 ms) e da taxa de transmissão da fonte que varia de um subintervalo para outro (fontes VBR e SS).

No final da simulação, é calculado o CLR de cada replicação como sendo a média entre todos os valores encontrados em cada subintervalo conforme expressão a seguir:

$$\overline{CLRrep_z} = \frac{\sum_{x=1}^n CLR_{sub_x}}{n}$$

Onde,

$\overline{CLRrep_z}$ é o CLR médio para a replicação "z";

CLR_{sub_x} é a razão de perda de células do subintervalo "x";

n é o número de subintervalos da simulação.

Para calcular o CLR final da simulação consideramos a média entre todas as replicações, conforme expressão:

$$CLR_{final} = \frac{\sum_{z=1}^p \overline{CLRrep_z}}{p}$$

Onde,

CLR_{final} é o CLR médio entre todas as replicações feitas na execução do simulador para um dado cenário;

p é o número de replicações feitas.

Também no final da simulação, é calculado o intervalo de confiança para o CLR final utilizando as seguintes expressões:

$$s^2 = \frac{n \sum_{i=1}^n (\overline{CLRrep}_i)^2 - \left(\sum_{i=1}^n \overline{CLRrep}_i \right)^2}{n(n-1)}$$

$$s = \sqrt{s^2}$$

Onde,

$CLRrep_i$ é o CLR da replicação "i";

n é o número de amostras de CLR. Cada amostra de CLR corresponde ao valor calculado em uma replicação;

s^2 é a variância das amostras de CLR calculado em cada replicação;

s é o desvio padrão das amostras de CLR calculado em cada replicação.

O intervalo de confiança para o valor final é dado pelas seguintes expressões:

$$Erro_z = 3.25 * \frac{s}{\sqrt{n}}$$

$$IC_z = \left[\overline{CLRfinal}_z - Erro_z, \overline{CLRfinal}_z + Erro_z \right]$$

Onde,

$\overline{CLRfinal}_z$ é o CLR médio para o cenário "z";

IC_z é o intervalo de confiança do cenário "z";

$Erro_z$ é o valor que determina o limite superior e inferior do intervalo de confiança para o cenário "z". A constante "3.25" da expressão refere-se à probabilidade de 99 % de acerto com 9 graus de liberdade de que o CLR estará dentro dos limites do intervalo de confiança (tabela da distribuição $t_{[0,995,9]}$);

s é o desvio padrão das amostras de CLR calculado em cada replicação;

n é o número de replicações para cada cenário.

O resultado final apresentado pelo simulador inclui o CLR médio de todas as conexões (CLR_{final}). A figura 4.7 mostra um resumo da seqüência de cálculos utilizados pelo simulador para estimar o valor de CLR.

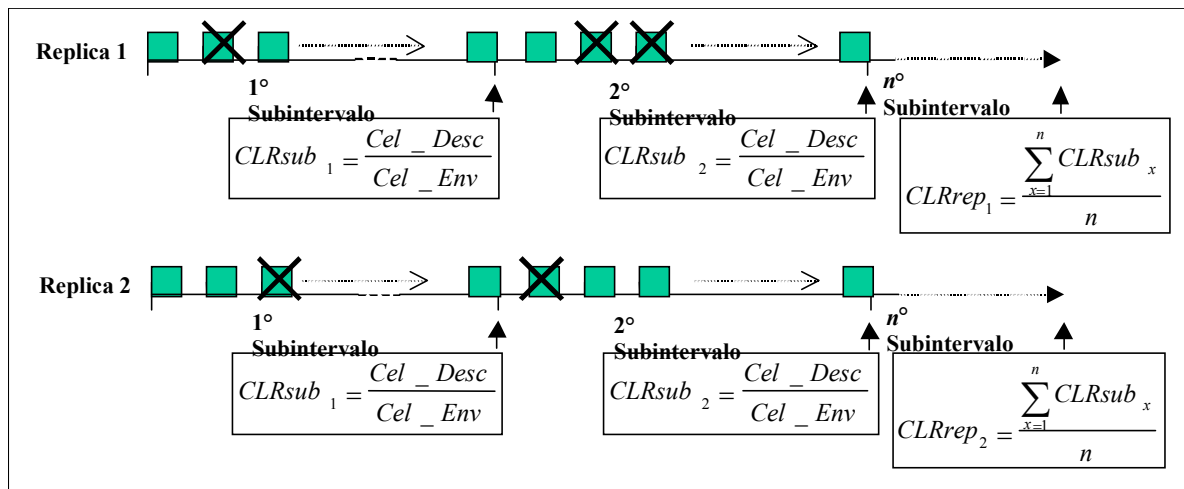


FIGURA 4.7 - Estimativa do CLR para duas replicações [GON 2001]

4.6 Processo das simulações

A saída da execução de uma simulação é uma seqüência de medidas que dependem da seqüência de números aleatórios utilizada. Repetindo a execução da simulação com outra seqüência de números aleatórios obteremos um resultado diferente do primeiro. Portanto, para chegarmos a um valor estimado do comportamento do sistema real, devemos avaliar estatisticamente os resultados das "n" execuções da simulação. No desenvolvimento clássico do intervalo de confiança para a média, assume-se que as amostras são independentes e identicamente distribuídas. Pelo teorema do limite central [SOA 90], a distribuição da soma de "n" amostras independentes de "x" se aproxima da distribuição normal para valores grandes de "n" (normalmente n=10 a 15 já é suficiente). A maioria dos procedimentos utilizados na estimativa dos parâmetros ou testes de hipóteses se baseia na média amostral. Se uma variável aleatória provém de uma amostra normalmente distribuída, então a média amostral também é normalmente distribuída. Se a variável aleatória provém de uma amostra que não é normalmente distribuída, mas de tamanho suficientemente grande ($n \geq 10$), então o teorema do limite central mostra que a média amostral tende à curva normal [SOA 90].

Se assumirmos que a média de "n" amostras é normalmente distribuída ($n \geq 10$), então podemos afirmar, com o nível de confiança $1 - \alpha$, que o parâmetro da população que nos dá a média de uma variável aleatória X, encontra-se dentro do intervalo de confiança a seguir [SOA 90] [GON 2001] [JAI 91]:

$$\left(\bar{X}_n - t_{[1-\alpha/2, n-1]} * \frac{S}{\sqrt{n}}, \bar{X}_n + t_{[1-\alpha/2, n-1]} * \frac{S}{\sqrt{n}} \right)$$

Onde,

\bar{X}_n é média das amostras. Cada amostra corresponde ao resultado de uma execução de simulação;

n é o número de amostras. Corresponde ao número de replicações (execuções) da simulação;

$t_{[1-\alpha/2;n-1]}$ é o valor dado pela tabela da distribuição t^1 ;

α é o nível de significância utilizado para calcular o nível de confiança.

O nível de confiança é igual a $100 * (1 - \alpha) \%$ ou, em outras palavras, um α de 0,01 indica um nível de confiança de 99% e $n-1$ indica qual é o grau de liberdade;

s é o desvio padrão da média das amostras.

O desvio padrão das amostras é calculado com a seguinte expressão:

$$s = \sqrt{\frac{n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2}{n(n-1)}}$$

Onde,

x_i é o valor da amostra "i";

n é o número de amostras.

Para a avaliação do intervalo de confiança para a média de uma variável aleatória é importante que tenhamos várias amostras independentes resultantes da execução do simulador. Há vários métodos para estimar o intervalo de confiança [SOA90], entre eles citamos dois métodos clássicos:

- *Replicação* - que emprega execuções separadas do simulador, com cada execução sendo considerada uma rajada para o cálculo de uma única amostra da variável aleatória " X_i ", definida como a média na execução "i". A vantagem desse método é a independência entre cada amostra. Como desvantagens desse método podemos citar: o fato de que cada replicação contém um segmento inicial de instabilidade significativo, e o fato de que apenas uma amostra é obtida em cada replicação, o que pode significar um grande tempo gasto em execuções pelo simulador [GON 2001].
- *Subintervalos* - que divide uma execução em iguais subintervalos (rajadas), e calcula um valor de " X_i " como uma média em cada subintervalo de "i". Assume-se que " X_i " sejam independentes. A vantagem do uso de subintervalos na estimativa dos intervalos de confiança é que uma única execução pode ser feita na obtenção da estimativa, além disso, apenas um período transiente é excluído dos resultados. A desvantagem do procedimento é a necessidade do estabelecimento da largura do subintervalo, de forma a torná-los independentes [GON 2001].

¹ Se o numero de amostras for maior que 30, então utilizar a tabela de distribuição normal (tabela $Z_{1-\alpha/2}$) [JAI 91].

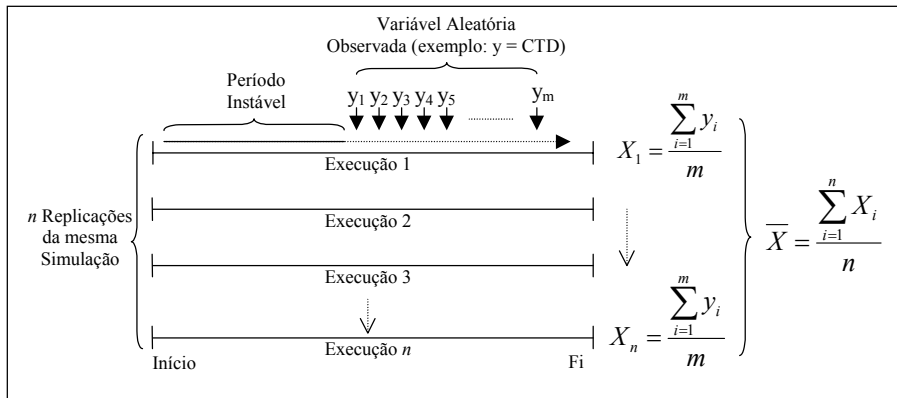


FIGURA 4.8 - Intervalo de Confiança: Método de Replicação da Simulação

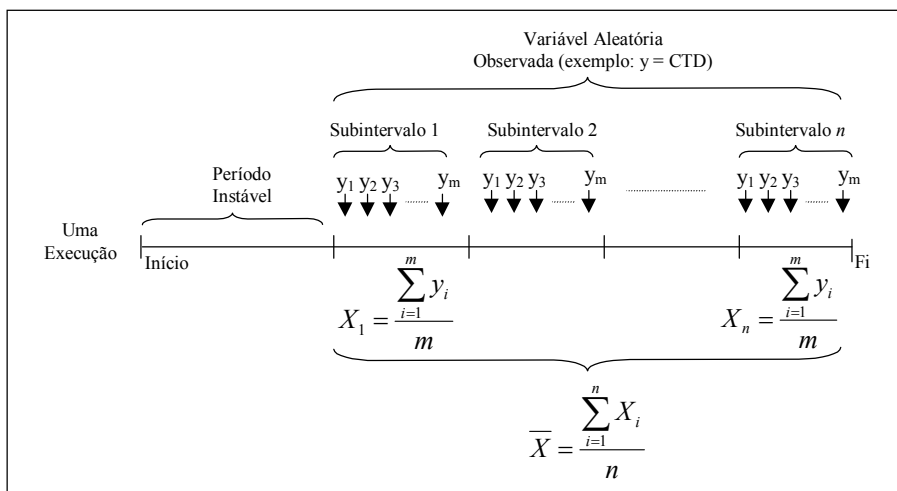


FIGURA 4.9 - Intervalo de Confiança: Método de Subintervalo da Simulação

Nessa dissertação ambos métodos foram utilizados, sendo que para avaliação do IC em ambos simuladores o método de replicação foi aplicado para que os valores finais fossem médios de valores médios, podendo-se, assim, utilizar o cálculo do intervalo de confiança baseados no teorema do limite central.

O nível de confiança utilizado para cálculo do intervalo de confiança em todas as simulações desse trabalho é de 99%, ou seja, α é igual a 0.01.

4.7 Análise dos Resultados

A seguir são apresentados os gráficos obtidos nas simulações feitas com o simulador ATM (NIST) e IP (NS-2) respectivamente. Para melhor visualização e análise dos resultados estão dispostos lado a lado os resultados em cada um dos simuladores em situações análogas e será apresentada a análise por métrica. No total foram gerados 68 gráficos¹.

¹ No NS-2: 3 métricas (CLR, CTD e CDV) * 2 fluxos (VBR e SS) * 2 tipos de gráficos (c/ IC e s/ IC) * 2 conjuntos de cenários (BV e BF) * 2 tamanhos de pacotes (240 bytes e 48 bytes) = 48 gráficos; No NIST: 2 métricas (CTD e CDV) * 2 fluxos (VBR e SS) * 2 tipos de gráficos (c/ IC e s/ IC) * 2 conjuntos de cenários (BV e BF) + 1 métrica (CLR) * 1 fluxo (SS) * 2 tipos de gráficos (c/ IC e s/ IC) * 2 conjuntos de cenários (BV e BF) = 20 gráficos.

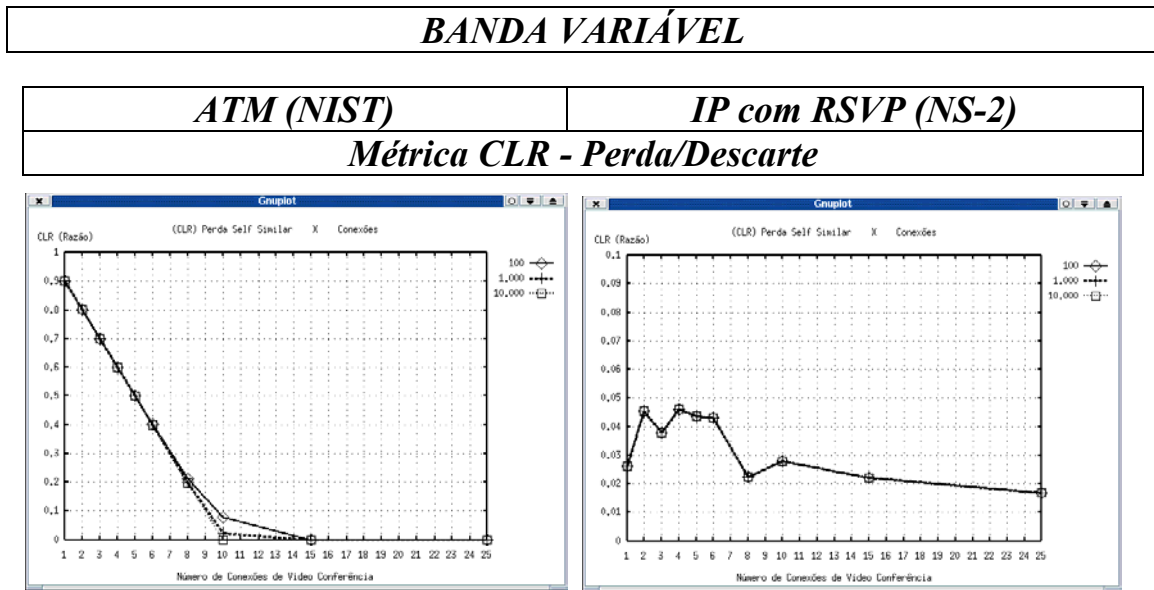


FIGURA 4.10 - CLR para o fluxo auto-similar com banda variável

Nas simulações ATM (NIST) não houveram perdas para o fluxo VBR agregados com banda variável.

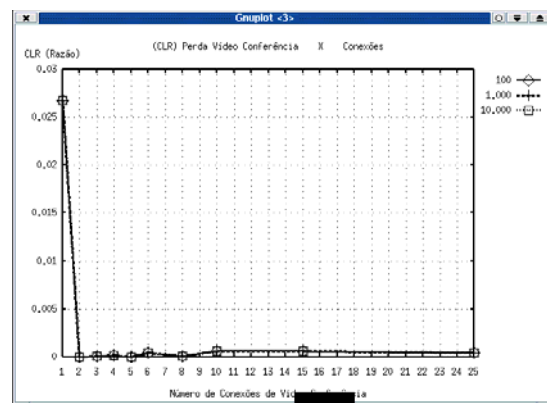


FIGURA 4.11 - CLR para os fluxos VBR agregados com banda variável

Observando os resultados obtidos, notamos que as perdas para os fluxos modelados como auto-similar, aplicações elásticas, são muito mais significativas na rede ATM do que na rede IP. Justamente o contrário se estabelece quando se trata dos fluxos VBR, que representam os tráfegos multimídia, chamadas de tempo real ou não elástica. Para a rede ATM, nesse caso não houve perda de célula mostrando o tratamento prioritário que a rede presta para esta classe de serviço, enquanto que na rede IP houve uma baixíssima perda de pacotes, tendo um valor maior quando o recurso largura do enlace é menor e se estabilizando em torno de um mesmo valor nos demais cenários.

Também podemos observar que o tamanho do *buffer* não apresentou influência para esta métrica, mostrando-se desnecessário investir nesse recurso para se obter menos perdas.

BANDA VARIÁVEL

ATM (NIST)	IP com RSVP (NS-2)
Métrica CTD - Atraso	

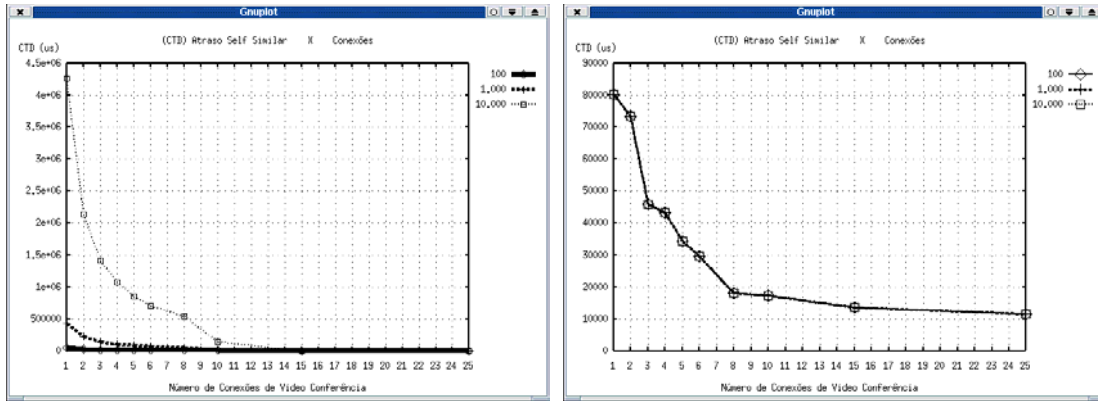


FIGURA 4.12 - CTD para o fluxo auto-similar com banda variável

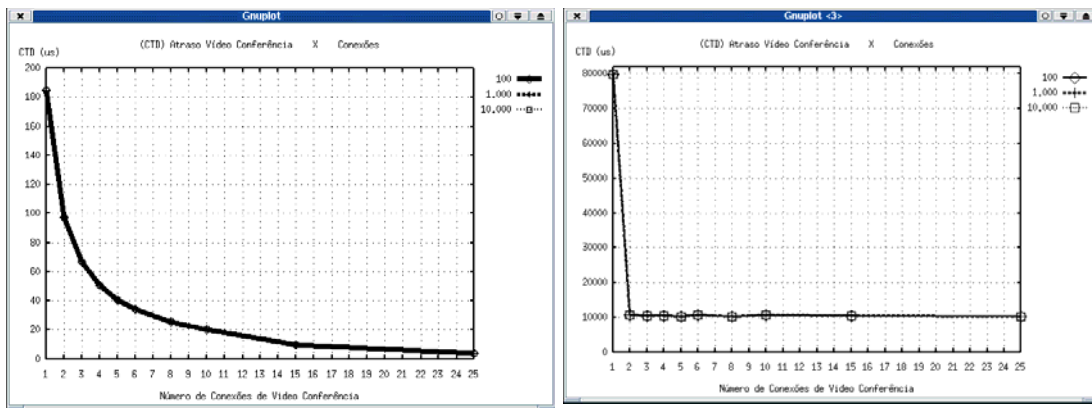


FIGURA 4.13 - CTD para os fluxos VBR agregados com banda variável

Observando os gráficos acima, notamos que os atrasos ocorridos na rede ATM entre os dois tipos de fluxos mostram também a diferença de tratamento em relação a classes de serviços; para os fluxos VBR os atrasos computados são ótimos e para os fluxos auto-similares, os atrasos são maiores, mas ainda bons, ocorrendo influência do fator tamanho do *buffer*. Na rede IP, se observa uma menor diferenciação nos valores de atrasos entre os dois tipos de fluxos, ficando em ambos os casos com valores bons, independentes do fator tamanho do *buffer*, também mostrando o tratamento diferenciado para suas classes de serviço. Observamos também, na rede IP, para os fluxos VBR, um valor maior quando o recurso largura do enlace é menor e se estabilizando em torno de um mesmo valor, 10 milisegundos, nos demais cenários, ou seja, na medida que aumenta a disponibilidade de banda.

BANDA VARIÁVEL

ATM (NIST)

IP com RSVP (NS-2)

Métrica CDV - Variação do Atraso

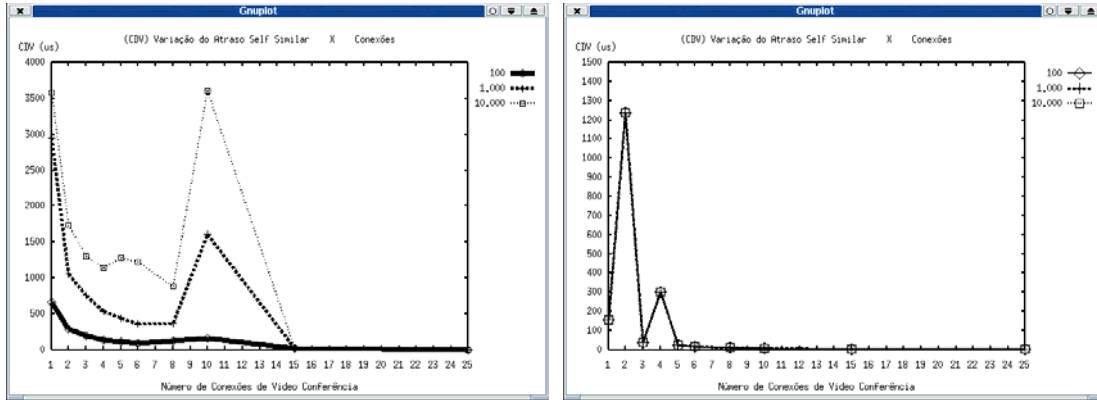


FIGURA 4.14 - CDV para o fluxo auto-similar com banda variável

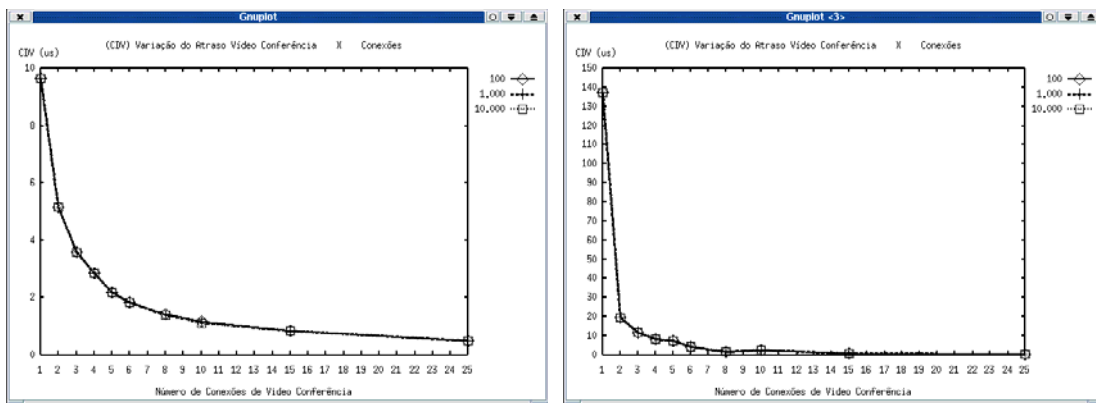


FIGURA 4.15 - CDV para os fluxos VBR agregados com banda variável

Observando os resultados dos gráficos acima notamos que a variação do atraso na maioria dos casos apresenta comportamento compatível com os das outras métricas em ambas as redes, computando valores considerados aceitáveis para estes dois tipos de fluxos. Todavia, para o tráfego auto-similar, em ambas redes, ocorreram alguns picos que poderiam ser mais bem investigado, ficando como sugestão para pesquisa.

BANDA FIXA

<i>ATM (NIST)</i>	<i>IP com RSVP (NS-2)</i>
<i>Métrica CLR - Perda/Descarte</i>	

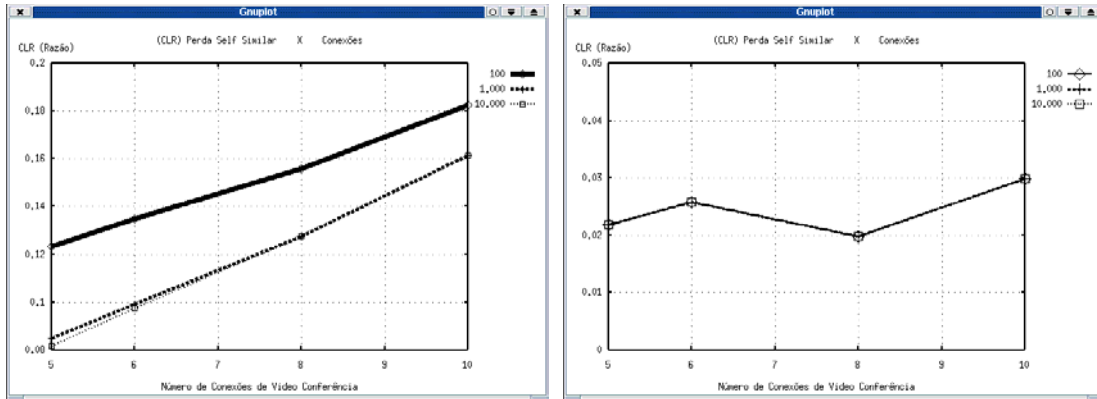


FIGURA 4.16 - CLR do fluxo auto-similar com banda fixa

Nas simulações ATM (NIST) não houveram perdas do fluxo VBR agregado com banda fixa

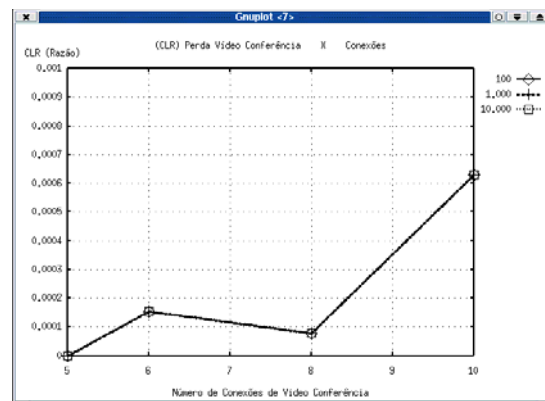


FIGURA 4.17 - CLR do fluxo VBR agregado com banda fixa

Lembrando que com a largura de banda fixa e com o aumento de conexões simultâneas compartilhando o enlace as aplicações vão tendo menos recurso. Assim observamos os resultados obtidos e verificamos que na rede ATM, para fluxos auto-similares, a tendência no aumento dos valores para perda de células se confirma, existindo ainda uma pequena influência do fator tamanho do buffer, e para os fluxos VBR se mantém com perda zero. Na rede IP, os valores mostram também uma variação, mas são ficam mais ou menos constantes. Em ambos os casos todos os valores computados são bastante bons.

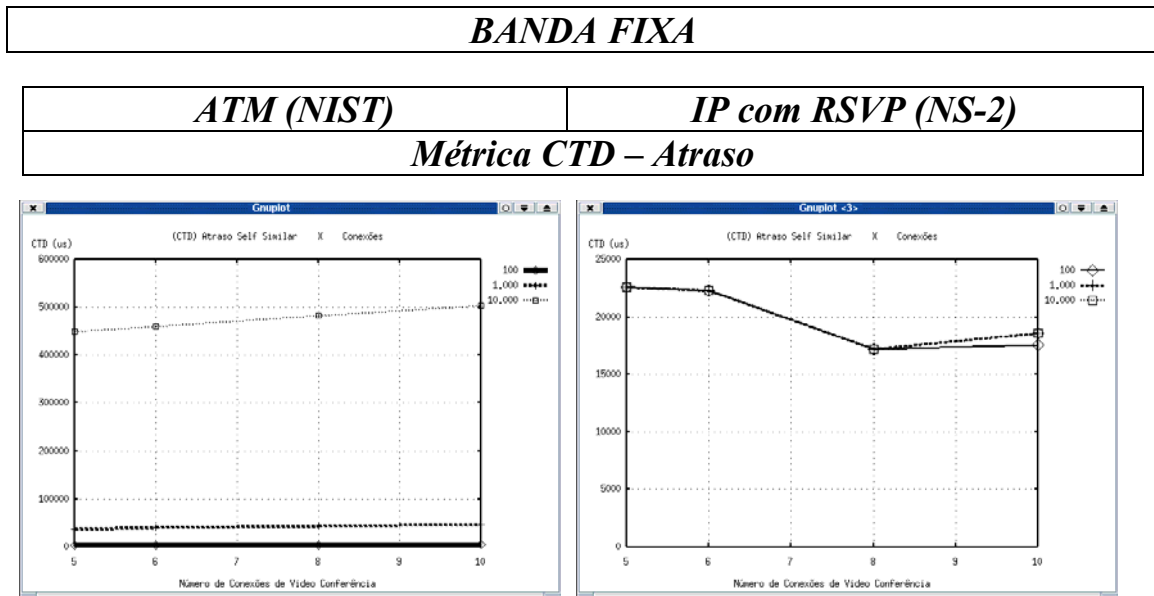


FIGURA 4.18 - CTD para o fluxo auto-similar com banda fixa

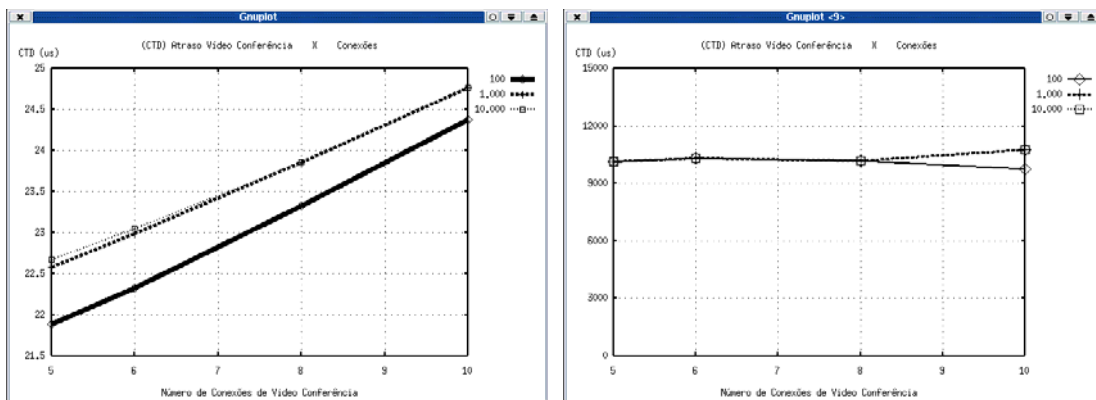


FIGURA 4.19 - CTD para o fluxo VBR agregado com banda fixa

Considerando a mesma relação citada na análise anterior sobre a largura de banda versus o aumento de conexões simultâneas compartilhando o enlace, observamos novamente a mesma tendência nas redes, ou seja, na rede ATM ocorre aumento nos valores de atraso e na rede IP os valores se mantêm mais constantes. Na rede ATM, na medida que a banda fica mais estressada, a curva cresce para os valores de atraso do fluxo mais prioritário, sendo que para o auto-similar a curva fica mais linear. Já na rede IP, o padrão de comportamento se mantém constante para ambos os fluxos e visualmente também apresenta a mesma curva que no caso com banda variável. Em ambos os casos todos os valores computados são bastante bons.

BANDA FIXA

<i>ATM (NIST)</i>	<i>IP com RSVP (NS-2)</i>
<i>Métrica CDV - Variação do Atraso</i>	

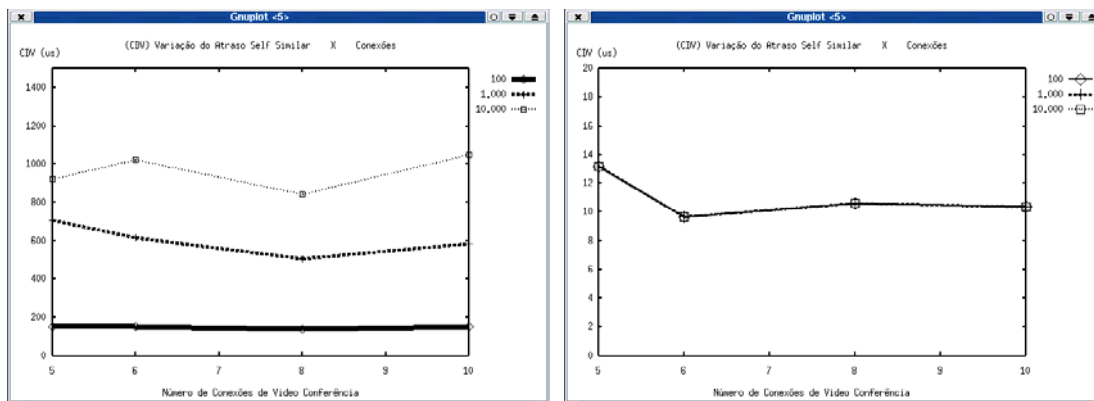


FIGURA 4.20 - CDV para o fluxo auto-similar com banda fixa

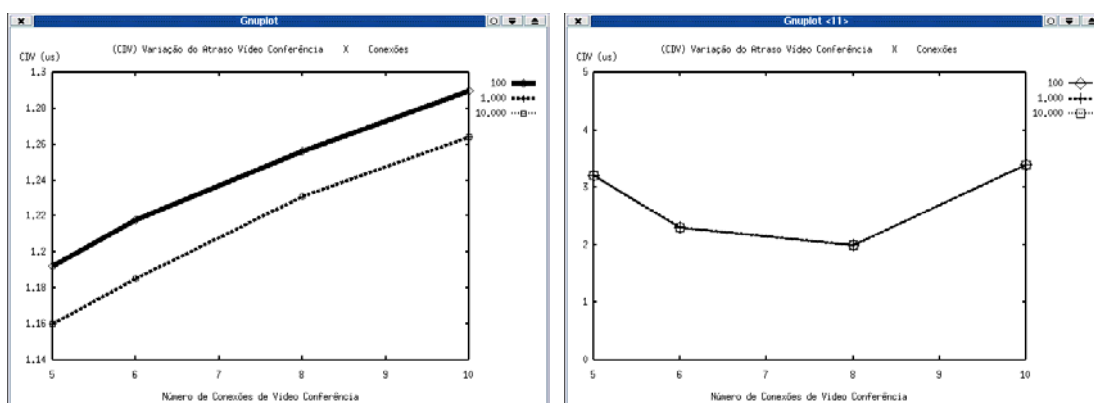


FIGURA 4.21 - CDV para o fluxo VBR agregado com banda fixa

Observando os gráficos acima, podemos notar que a variação do atraso em todos os casos apresentam comportamentos compatíveis com os das outras métricas em ambas as redes, computando valores considerados aceitáveis para estes dois tipos de fluxos.

5 Considerações Finais

5.1 Conclusões

Originalmente em redes IP não existia compromisso com QoS. Entretanto, como o seu uso teve um crescimento exponencial com a Internet, fez-se necessário um envolvimento para suportar os requerimentos de QoS das diversas aplicações com mídias diferenciadas tal como as aplicações multimídia solicitam. Estes esforços podem ser vistos com o desenvolvimento de conceitos de reserva de recursos, diferenciação de serviços, MPLS, Engenharia de tráfego e Internet2.

Por outro lado, o aparecimento da rede ATM, que já nasceu com claras intenções de oferecer diferenciação de classes de serviço, assim como, garantir o QoS especificado nas diversas aplicações, veio impulsionar a pesquisa nesta área para que o mercado das comunicações de dados pudessem continuar se expandindo e suportando de forma eficiente as necessidades de seus usuários. É bem verdade, que hoje em dia, ainda está difícil de se dizer quando tudo, que foi cuidadosamente planejado pelos órgãos padronizadores como ATM Forum e ITU-T, venha a ser viável na prática, mas o impulso inicial já foi dado e é, sem dúvida alguma, a base para todo esse novo panorama na área das redes de comunicações de dados.

Cada um dos enfoques apresentados nessa dissertação tem suas vantagens e desvantagens. Uma importante diferença gira em torno do fato de que ATM tem um enfoque “*hard-state*” permitindo que a rede garanta QoS e reserva de largura de banda recusando tentativas de conexões se admitindo um tráfego adicional poder degradar a qualidade das conexões existentes. A rede ATM tenta outras rotas, bloqueando a requisição de serviços somente se todas as rotas estiverem cheias. De forma contrária, o enfoque “*soft-state*” não pode realizar esta função facilmente, uma vez que o RSVP confia nos protocolos de roteamento das camadas inferiores para descobrir para onde poderá emitir os pedidos de reserva. Quando um fluxo RSVP tem seu caminho alterado, o módulo de roteamento notifica o módulo RSVP sobre a mudança da rota. Assim, o RSVP pode ajustar rapidamente as reservas de recursos para as novas rotas. Em outras palavras, em IP, reservas e roteamento operam independentemente, enquanto em ATM, operam concorrentemente durante o estabelecimento da conexão.

A utilização desse trabalho como base para modelagem e simulação do comportamento de determinada rede sobre os efeitos de diferentes cargas de tráfegos e posterior avaliação e comparação do desempenho das alternativas, poderá ajudar no projeto de novas redes, como também, na manutenção de redes existentes baseadas nas tecnologias apresentadas. E, diferentes configurações de redes, modelos e cargas de tráfego podem ser exercitadas para verificar previamente as dificuldades de desempenho. Comparar as medidas de desempenho é o caminho para dizer qual é a mais eficiente. A comparação utiliza valores numéricos que representam as quantidades alcançadas pelas atividades do sistema. A comparação é feita considerando medidas que quantificam a mesma característica de desempenho para um mesmo serviço. O mínimo tolerável para as características de desempenho é determinado pelas necessidades básicas que viabilizam o serviço no sistema. Comparando as medidas de desempenho entre sistemas, considerando a finalidade, podemos dizer qual é o mais eficiente, mas não qual é o mais apropriado. A eficiência não é o único fator determinante na seleção de sistemas computacionais. O fato de um sistema ser mais eficiente do que o outro não

garante que será o escolhido para ser utilizado. Outros fatores de benefício, tais como, facilidade operacional, escalabilidade, disponibilidade, padronização e custo de implementação e instalação, influenciam na decisão. Sistemas podem ter desempenhos extraordinários e grandes facilidades, mas terem implantação inviável devido ao custo. Os sistemas mais eficientes podem ser descartados por não haver necessidade prática de atingir os níveis de desempenho apresentados. Como em qualquer processo de escolha, deve ser pesada a relação entre os custos e os benefícios.

Pelos resultados numéricos podemos verificar as seguintes condições nos cenários modelados:

- Cenário Banda Variável: se a estimativa inicial para a banda da rede for pelo menos igual ao PCR agregado das aplicações multimídia, então ambas tecnologias se mostraram eficientes e suportam os serviços;
- Cenário Banda Fixa: se o PCR agregado das aplicações multimídia for menor ou igual a 110% da banda fixa, então ambas tecnologias se mostraram eficientes e suportam os serviços;

Na medida que se aumentar os números de nodos pelos quais o tráfego vai passar, e natural que os valores também aumentem, e se considerarmos as premissas apresentadas em [MCD 2000], que coloca como sendo esperado um crescimento linear para as métricas CLR e CTD e um crescimento no mínimo na razão da raiz quadrada no número de nodos atravessados pela informação para a métrica CDV, então um crescimento na topologia da rede na ordem de 10 a 20 vezes o simulado ainda estaria dentro de limites aceitáveis, podendo ser esse um exemplo de um campus universitário.

Assim sendo, olhando os valores apresentados nos gráficos na sessão anterior e as exigências mínimas dos serviços modelados, analisamos o parâmetro de QoS de interesse para ambas tecnologias que é o atraso ou CTD, conforme tabela 3.2, e concluímos que:

- Se aumentássemos em 20 vezes o cenário simulado, banda variável, os atrasos na rede IP, para os fluxos auto-similares, que variaram de 80 ms a 10 ms, ficariam entre 1600 ms a 200 ms, podendo ainda atender a vários tipos de serviços de dados, tais como, e-mail, transferência de arquivo e outros, conforme [STA 2000];
- Se aumentássemos em 20 vezes o cenário simulado, banda variável, os atrasos na rede IP, para os fluxos das vídeo conferências, que ficaram em torno de 10 milissegundos, passariam para 200 milissegundos, ficando dentro dos limites aceitáveis, que são de 150 ms a 350 ms, conforme [STA 2000];
- Se aumentássemos em 20 vezes o cenário simulado, banda variável, os atrasos na rede ATM, para os fluxos auto-similares, que ficaram abaixo de 500 ms, com buffer de 100 células, passariam para 10000 ms ou 10 s, o que fica ainda dentro dos limites apresentados em [STA 2000] para muitos serviços de dados;
- Se aumentássemos em 20 vezes o cenário simulado, banda variável, os atrasos na rede ATM, para os fluxos das vídeo conferências, que ficaram abaixo de 200 μ s, passariam a computar valores abaixo de 4 ms, o que representa um desempenho ótimo para esses fluxos.
- Se aumentássemos em 20 vezes o cenário simulado, banda fixa, os atrasos

na rede IP, para os fluxos auto-similares, que variaram de 22 ms a 18 ms, ficariam entre 440 ms a 360 ms, podendo ainda atender a vários tipos de serviços de dados, tais como, e-mail, transferência de arquivo e outros, conforme [STA 2000];

- Se aumentássemos em 20 vezes o cenário simulado, banda fixa, os atrasos na rede IP, para os fluxos das vídeo conferências, que ficaram em torno de 10 milisegundos, passariam para 200 milisegundos, ficando dentro dos limites aceitáveis, que são de 150 ms a 350 ms, conforme [STA 2000];
- Se aumentássemos em 20 vezes o cenário simulado, banda fixa, os atrasos na rede ATM, para os fluxos auto-similares, que ficaram abaixo de 100 ms, com buffer de 100 células, passariam para 2000 ms ou 2 s, o que fica ainda dentro dos limites apresentados em [STA 2000] para muitos serviços de dados;
- Se aumentássemos em 20 vezes o cenário simulado, banda variável, os atrasos na rede ATM, para os fluxos das vídeo conferências, que ficaram abaixo de 25 μ s, passariam a computar valores abaixo de 500 μ s, o que representa um desempenho ótimo para esses fluxos.

Com respeito a garantia de QoS, ATM esta à frente de qualquer enfoque baseado em IP. Entretanto, muitos especialistas acreditam que IP eventualmente emulará ou melhorará o gerenciamento de tráfego do ATM e as funções de QoS, permitindo ainda melhores resultados dos que os computadores nessa dissertação.

Não se pode deixar de falar sobre o problema econômico que gira em torno destas duas tecnologias. No caso do IP, uma rede amplamente implantada e utilizada, com interfaces amigáveis e existentes em qualquer loja de informática, faz com que na hora da decisão pela sua implantação, isto torne a escolha muitas vezes mais atraente. Já no caso do ATM, embora conhecidamente oferece nativamente garantias de QoS, acaba por perder mercado em função das suas dificuldades operacionais, tais como, hardware mais caros, e ausência de aplicações para os usuários finais.

Como produtos resultantes dessa dissertação temos:

- a nova versão do simulador NIST ATM/HFC que pode ser utilizada como ferramenta para modelar e exercitar outros cenários de rede;
- os scripts desenvolvidos que podem ser utilizados como base para criação de outras simulações;

Importante salientar que as simulações finais dessa dissertação levaram em torno de 50 horas de execução, e que, na medida que a complexidade do ambiente simulado aumentar o tempo necessário para a execução da simulação também aumentará, sendo esse um inconveniente que será cada vez mais minimizado com a disponibilização de computadores mais poderosos.

Esta página foi deixada em branco de propósito.

5.2 Trabalhos Futuros

Durante o desenvolvimento desse trabalho, várias limitações foram sentidas e que talvez possam ser o início de novos trabalhos.

Segue uma pequena lista de sugestões para trabalhos futuros:

- Desenvolvimento dos módulos para RSVP suportar o serviço de QoS garantido;
- Desenvolvimento de um modelo de tráfego auto-similar para o simulador NS-2;
- Utilização de geradores de tráfego TCP/IP no lugar do gerador de tráfego auto-similar e posterior comparação com os resultados dessa dissertação objetivando constatar a real auto-similaridade entre estes dois tipos de fontes;
- Para o simulador NIST, utilização do modelo de switch com algoritmos de controle de tráfego para serviços ABR, objetivando verificar as diferenças nos valores de perdas de células para serviços menos prioritários;

Anexo 1 Relação dos programas alterados no simulador NIST

O simulador NIST ATM/HFC foi desenvolvido na linguagem C e está dividido em 99 módulos (programas fontes). Desse total, 5 módulos foram alterados para adequar o simulador a este trabalho. A tabela Anexo 1.1 lista as alterações feitas nos arquivos fontes. É importante salientar que estas alterações partiram da versão do NIST implementada em [GON 2001], sendo, portanto, uma complementação aos vários módulos já modificados.

TABELA Anexo 1.1 - Notas das Alterações Efetuadas

Nome do Arquivo Fonte	Nome da Função	Modificação e comentário
CLOCK.c	draw_digits()	- ARG_inc: Incl. de comentário na tela da interface gráfica : "Milliseconds tat"
SSCONNECTION.h	Define	- ARG_inc: Inclui na estrutura SS Connectiont (tipo de componente) na posição que coincida com a definição Component (Component.h): <pre> Estatistica *cn_estatistica; </pre> A estrutura "Estatistica" está definida no component.h e é utilizada também na estrutura Component.
SSCONNECTION.c	cn_send() SS	- ARG_inc: timestamp de saída da célula na geração da célula: <code>cell->stamp1=ev_now();</code> - Arg_inc: Marca na célula qual o componente que vai recebe-la (aplicação) no outro extremo da conexão. - ARG_inc: incrementa nas estatísticas o número de células enviadas.
	cn_reset()	- ARG_inc: reseta campos de estatísticas: <code>Estat_reset(cn->cn_estatistica);</code>
	cn_create()	- ARG_inc: inicialização do parâmetro Células Enviadas para este destino - ARG_inc: inicialização do parâmetro Células Recebidas - ARG_inc: inicialização do parâmetro Tamanho de Subintervalo - ARG_inc: inicialização do parâmetro Número do Subintervalo - ARG_inc: inicialização do parâmetro Perda Média - ARG_inc: inicialização do parâmetro Delay Médio - ARG_inc: inicialização do parâmetro Jitter Médio - ARG_inc: inicialização do parâmetro

		Intervalo de Confiança do Jitter - ARG_inc: inicialização da área de estatística
	cn_receive()	- ARG_inc: Chama rotina para calcular estatísticas e depois imprime. ARG_inc: Atualiza variáveis da tela e imprime estatísticas.
	cn_route()	- ARG_alt: correção do nome do bte do outro extremo. Troquei l tail por l head
	cn_send()	- ARG_inc: incrementa nas estatísticas o número de células enviadas
	cn_hops()	- ARG_alt: adicionei o teste de SWITCH_CLASS, para permitir conexão da aplicação direto no switch. Para diminuir quantidade de eventos gerados
	cn_neighbor())	- ARG_alt: adicionei no teste SWITCH_CLASS, para permitir conexão da aplicação direto no switch. Para diminuir quantidade de eventos gerados
SUBR.c	calcula_estatisticas()	- ARG_inc: nova rotina para calcular estatísticas a partir da célula recebida
	calcula_estatisticas_intervalo()	- ARG_inc: nova rotina para calcular estatísticas por intervalo de tempo
	estat_init()	- ARG_inc: Rotina para inicializar variáveis de Estatísticas. Aloca espaço para as variáveis e retorna o endereço da estrutura.
	estat_reset())	- ARG_inc: Rotina para reinicializar variáveis de Estatísticas
	contabiliza_descarte()	- ARG_inc: Rotina para contabilizar na aplicação receptora o número de células descartadas pelo switch (transbordo de buffer).
	contabiliza_envio()	- ARG_inc: Rotina para contabilizar envio de células a partir da aplicação geradora de tráfego. Se o período de warmuptime não terminou, não contabiliza.
MAIN.c	Define Main()	- ARG_inc: variáveis para cálculo do tempo decorrido: time_t tempo_de_inicio, tempo_final; double tempo_real; - Antes de começar a simulação, guarda o tempo corrente. Utiliza a definição <time.h> - ARG_inc: Novo argumento de entrada na chamada do simulador para valor do subintervalo em unidades de microssegundos; - ARG_inc: Inicialização do valor do subintervalo caso o campo de entrada não seja utilizado

	<pre> tick t tam_interv = USECS_TO_TICKS(500); - ARG_inc: início do primeiro intervalo após o término do período de warmuptime - ARG_inc: Durante a simulação, acompanha o tempo simulado dividindo em subintervalos para executar as estatísticas. - ARG_inc: calculo do tempo decorrido na chamada da impressão do arquivo SIM_SNAP (final da simulação = stoptime). - ARG_inc: imprime resultados finais como última linha do arquivo SIM_LOG. No final da simulação, imprime os resultados das medias entre todas as conexões. - ARG_inc: Coloca o tempo simulado e o tempo real decorrido no arquivo SIM_LOG. - ARG_inc: Condicionar final da simulação pelo intervalo de confiança do Delay. </pre>
sim_resultado()	- ARG_inc: rotina para imprimir resultado final da Simulação
coloca_valor_interv()	- ARG_inc: coloca no display da tela de todos as aplicações o valor do intervalo para a execução da simulação. Só para constar no display.
verifica_intervalo_confianca()	- ARG_inc: rotina para verificar se o intervalo de confiança atingiu final da Simulação.

Anexo 2 Listagem dos principais "scripts" e arquivos das simulações para os simuladores NIST e NS-2

A2.1 - Simulador NIST

```
#####
## Data: 13/08/2001 ##
## Autora: Tatiana Rotava Schabbach ##
## Chama o script que gera os arquivos de configuracoes (30) ##
#####
# aplicacao VBR simulada: Video Conferência com PCR=1.16096750, ##
# Ton=5190 e Toff=34810 [ROB 96] ##
# aplicacao self-similar: MBR=10, H=0.8, Timegran=500 ##
# variacao buffers: 100, 1000 e 10000 células ##
#####

cd /home/tatiana/NIST/final-ubr-generic

/home/tatiana/NIST/final-ubr-generic/gerarqconf 10 1.16096750 5190
34810 100

/home/tatiana/NIST/final-ubr-generic/gerarqconf 10 1.16096750 5190
34810 1000

/home/tatiana/NIST/final-ubr-generic/gerarqconf 10 1.16096750 5190
34810 10000

#####
## Data: 13/08/2001 ##
## Autora: Tatiana Rotava Schabbach ##
#####
## Script para: ##
## ===== ##
## ##
## - Gerar "n" arquivos de configuracao VBR + 1 SS ##
## Onde "n" e' igual ao numero de conexoes simultaneas ##
## de 1 a "n". ##
## Exemplo: n=3, sao gerados 3 arquivos: ##
## - o primeiro arquivo com 1 conexao ##
## - o segundo arquivo com 2 conexoes ##
## - o terceiro arquivo com 3 conexoes ##
## ##
#####
## ##
## Entrada: ##
## - Numero de Conjuntos de conexoes (max. 10 conj.) ##
## - PCR (Mbit/s) ##
## - MBS (tempo medio da rajada em us) ##
## - Toff (tempo medio entre rajadas em us) ##
## - Tamanho Buffer ##
## ##
#####

gawk -v tot_con=$1 -v PCR=$2 -v MBS=$3 -v TOFF=$4 -v BUF=$5 '

```

```
BEGIN {
```

```

dir = "/home/tatiana/NIST/final-ubr-generic/modelos/"

# Se alterar a combinacao de conexoes: alterar no script de
execucao simulacao

    comb_cnx[1]=1
    comb_cnx[2]=2
    comb_cnx[3]=3
    comb_cnx[4]=4
    comb_cnx[5]=5
    comb_cnx[6]=6
    comb_cnx[7]=8
    comb_cnx[8]=10
    comb_cnx[9]=15
    comb_cnx[10]=25

if (tot_con > 10)
{
    print "=====> Maximo 10 conjuntos <====="
    exit(1)
}

    prep_vbrserv()
    arq_servico = "conf_sim_vbr"

for (ind = 1; ind <= tot_con; ind += 1)
{

    num_arq = comb_cnx[ind]

    arq_saida = arq_servico "_" num_arq "_" BUF

    gera_switch()

    for (num_fonte = 1; num_fonte <= num_arq; num_fonte +=
1)
        {
            gera_vbrserv()
        }

    for (num_fonte = 1; num_fonte <= num_arq; num_fonte
+= 1)
        {
            gera_vbrclie()
        }

## acrescenta fluxo ss
        comando = "cat " dir "ssserv_model >> " arq_saida
        system(comando)

        print " " >> arq_saida

        comando = "cat " dir "ssclie_model >> " arq_saida
        system(comando)

##

        gera_lk()

        for (num_fonte = 1; num_fonte <= num_arq; num_fonte
+= 1)

```



```

        {
            gera_neighbor()
        }

## acrescenta vizinhos do fluxo ss
        comando = "cat " dir "ss_neighbor_model >> "
arq_saida
        system(comando)
##
        print " " >> arq_saida

1)
        for (num_fonte = 1; num_fonte <= num_arq; num_fonte +=
        {
            gera_rota()
        }

## acrescenta rota do fluxo ss
        comando = "cat " dir "ss_rota_model >> " arq_saida
        system(comando)
##
        close (arq_saida)
    }

}

function prep_vbrserv()
{
    comando = "rm -f vbrserv_model_tmp*"
    system(comando)

    arq_entrada = dir "vbrserv_model"
    arq_saida = "vbrserv_model_tmp"
    troca = "<pcr>"
    por = PCR

    Le_Altera()

    arq_entrada = "vbrserv_model_tmp"
    arq_saida = "vbrserv_model_tmp1"
    troca = "<mbs>"
    por = MBS

    Le_Altera()

    arq_entrada = "vbrserv_model_tmp1"
    arq_saida = "vbrserv_model_tmp2"
    troca = "<toff>"
    por = TOFF

    Le_Altera()

}

function gera_switch()
{

```

```

    arq_entrada = dir "switch_model"
    troca = "<buffer>"
    por = BUF

    Le_Altera()

    print " " >> arq_saida

    }

function gera_bte()
{

    arq_entrada = dir "bte_model"
    troca = "<cx>"
    por = num_fonte

    Le_Altera()

    print " " >> arq_saida

    }

function gera_vbrserv()
{

    arq_entrada = "vbrserv_model_tmp2"
    troca = "<cx>"
    por = num_fonte

    Le_Altera()

    print " " >> arq_saida

    }

function gera_vbrclie()
{

    arq_entrada = dir "vbrclie_model"
    troca = "<cx>"
    por = num_fonte

    Le_Altera()

    print " " >> arq_saida

    }

function gera_lk()
{

    arq_entrada = dir "lk0_model"

## cria link com taxa igual a soma dos PCR das conexoes vbr

```

```

        troca = "<taxa>"
        por = num_arq*PCR

    Le_Altera()

    print " " >> arq_saida

}

function gera_neighbor()
{

    arq_entrada = dir "neighbor_model_v8"
    troca = "<cx>"
    por = num_fonte

    Le_Altera()

}

function gera_rota()
{

    arq_entrada = dir "rota_model_v8"
    troca = "<cx>"
    por = num_fonte

    Le_Altera()

}

function Le_Altera()
{

    eof = getline < arq_entrada

    while (eof != 0)
    {
        alt = gensub(troca,por,"g", $0)
        print alt >> arq_saida
        eof = getline < arq_entrada
    }

    close (arq_entrada)
    close (arq_saida)

}

,

rm -f vbrserv_model_tmp*
```

Modelos usados para a criação dos arquivos de configurações

Modelo para switch

```
# Simulador NIST-ATM Versao: 5.0
# Seed 965781655
# Time of snapshot (ticks) 0
# Tempo Simulado ==> 00:00:00 (0 seg)
# Tempo Execucao ==> 00:00:00 (0 seg)
component 'Switch' SWITCH 250 260
param 'Switch' 32 0 # Switch
param 0 12 0 # Delay to process a cell (uSec): 0
param 155 12 0 # Switching Slot time (Mbit/s): 155
param <buffer> 12 0 # Output q_size (cells, -1=inf): <buffer>
param 0 12 0 # High Threshold, Q Cong. Flag (cells): 0
param 0 12 0 # Low Threshold, Q Cong. Flag (cells): 0
param 1 12 0 # Logging every (ticks) (e.g. 1, 100): 1
pflags 2a 4 #Cells Received: 0
pflags 2a 4 #Cell Drop %: 0
pflags 2a 4 #Cells in VBR Q to LK0: 0
pflags 2a 4 #Cells dropped in VBR Q to LK0: 0
pflags 2a 4 #Cells in ABR Q to LK0: 0
pflags 2a 4 #Cells Dropped in ABR Q to LK0: 0
pflags 2a 1 #Congestion for Link LK0: FALSE
```

```
component 'TE0' BTE 250 150
param 'TE0' 32 0 # TE0
param -1 12 0 # Max Output Queue Size(-1=inf): -1
param 1 12 0 # Logging every (ticks) (e.g. 1, 100): 1
pflags 2a 4 #Cells Received: 0
pflags 2a 4 #Cells in VBR Q to LK0: 0
pflags 2a 4 #Cells dropped in VBR Q to LK0: 0
pflags 2a 4 #Cells in ABR Q to LK0: 0
pflags 2a 4 #Cells Dropped in ABR Q to LK0: 0
```

Modelo para Gerador VBR

```
component 'GER<cx>' VBRCONNECTION 180 420
param 'GER<cx>' 32 0 # GER<cx>
param <pcr> 12 0 # Bit Rate (Mbits/s): <pcr>
param <mbs> 12 0 # Mean Burst Length (usecs): <mbs>
param <toff> 12 0 # Mean Interval Between Bursts (usecs): <toff>
param 1 12 0 # Start time (usecs): 1
param 1500 12 0 # Number of MBits to be sent: 1500
pflags 2a 4 #Celulas Enviadas para ca': 0
pflags 2a 4 #Celulas Recebidas: 0
pflags 2e 4 # Tamanho Subintervalo (ms): 0
pflags 2a 4 # Subintervalo numero: 0
pflags 2a 4 # Perda Media (%): 0
pflags 2a 4 # Atraso Medio (usecs): 0
pflags 2a 4 # Interv. Conf. 99% (-/+ usecs): 0
pflags 2a 4 # Jitter Medio (usecs): 0
pflags 2a 4 # Interv. Conf. 99% (-/+ usecs): 0
pflags 22 0
```

Modelo para Receptor VBR

```
component 'REC<cx>' VBRCONNECTION 186 99
param 'REC<cx>' 32 0 # REC<cx>
param 0 12 0 # Bit Rate (Mbits/s): 0
```

```

param 0 12 0 # Mean Burst Length (usecs): 0
param 0 12 0 # Mean Interval Between Bursts (usecs): 0
param 0 12 0 # Start time (usecs): 0
param 0 12 0 # Number of MBits to be sent: 0
pflags 2a 4 #Celulas Enviadas para ca': 0
pflags 2a 4 #Celulas Recebidas: 0
pflags 2a 4 # Tamanho Subintervalo (ms): 0
pflags 6a 4 # Subintervalo numero: 0
pflags 2a 4 # Perda Media (%): 0
pflags 2a 4 # Atraso Medio (usecs): 0
pflags 2a 4 # Interv. Conf. 99% (-/+ usecs): 0
pflags 2a 4 # Jitter Medio (usecs): 0
pflags 2a 4 # Interv. Conf. 99% (-/+ usecs): 0
pflags 22 0

```

Modelo para Gerador Auto-similar

```

component 'GER_ss' SCONNECTION 175 88
param 'GER_ss' 32 0 # GER_ss
param 10 12 0 # Mean Bit Rate (Mbits/s): 10
param 500 12 0 # Timegran (e.g. 100) (usecs): 500
param 1 12 0 # Start time (usecs): 1
param 0.8 12 0 # Hurst param H (0.5<=H<1): 0.8
pflags 2a 4 #Num Sent: 0
pflags 2a 4 #Output Rate (Mbps): 0
pflags 2a 4 #Celulas Enviadas para ca': 0
pflags 2a 4 #Celulas Recebidas: 0
pflags 2e 4 # Tamanho Subintervalo (ms): 0
pflags 2a 4 # Subintervalo numero: 0
pflags 2a 4 # Perda Media (%): 0
pflags 2a 4 # Atraso Medio (usecs): 0
pflags 2a 4 # Interv. Conf. 99% (-/+ usecs): 0
pflags 2a 4 # Jitter Medio (usecs): 0
pflags 2a 4 # Interv. Conf. 99% (-/+ usecs): 0
pflags 22 0

```

Modelo para Receptor Auto-similar

```

component 'REC_ss' SCONNECTION 173 382
param 'REC_ss' 32 0 # REC_ss
param 0 12 0 # Mean Bit Rate (Mbits/s): 0
param 500 12 0 # Timegran (e.g. 100) (usecs): 500
param 0 12 0 # Start time (usecs): 0
param 0.8 12 0 # Hurst param H (0.5<=H<1): 0.8
pflags 2a 4 #Num Sent: 0
pflags 2a 4 #Output Rate (Mbps): 0
pflags 2a 4 #Celulas Enviadas para ca': 0
pflags 2a 4 #Celulas Recebidas: 0
pflags 2e 4 # Tamanho Subintervalo (ms): 0
pflags 6a 4 # Subintervalo numero: 0
pflags 2a 4 # Perda Media (%): 0
pflags 2a 4 # Atraso Medio (usecs): 0
pflags 2a 4 # Interv. Conf. 99% (-/+ usecs): 0
pflags 2a 4 # Jitter Medio (usecs): 0
pflags 2a 4 # Interv. Conf. 99% (-/+ usecs): 0
pflags 22 0

```

Modelo para Link

```
component 'LK0' LINK 250 210
param 'LK0' 32 0 # LK0
param <taxa> 12 0 # Link Speed (Mbits/sec): <taxa>
pflags 2 0
pflags 2 0
pflags 2 0
pflags 2 0
pflags 2 0
param 1 12 0 # Distance (Km): 1
pflags 2a 4 #Link rate (Mbit/s) to TE0: 0
pflags 2a 4 #Link rate (Mbit/s) to Switch: 0
```

```
neighbor1 'TE0' 'LK0'
neighbor1 'LK0' 'TE0'
neighbor1 'LK0' 'Switch'
neighbor1 'Switch' 'LK0'
```

Modelo para neighbor VBR

```
neighbor1 'GER<cx>' 'Switch'
neighbor1 'TE0' 'REC<cx>'
neighbor1 'REC<cx>' 'TE0'
neighbor1 'Switch' 'GER<cx>'
```

Modelo para neighbor Auto-similar

```
neighbor1 'GER_ss' 'Switch'
neighbor1 'TE0' 'REC_ss'
neighbor1 'REC_ss' 'TE0'
neighbor1 'Switch' 'GER_ss'
```

Modelo para rota VBR

```
routel 'GER<cx>' 'Switch' 'LK0' 'TE0' 'REC<cx>'
```

Modelo para rota Auto-similar

```
routel 'REC_ss' 'TE0' 'LK0' 'Switch' 'GER_ss'
```

**Exemplo do arquivo de configuração gerado para 1 conexão vbr
e buffer de 100 células**

```

# Simulador NIST-ATM Versao: 5.0
# Seed 965781655
# Time of snapshot (ticks) 0
# Tempo Simulado ==> 00:00:00 (0 seg)
# Tempo Execucao ==> 00:00:00 (0 seg)
component 'Switch' SWITCH 250 260
param 'Switch' 32 0 # Switch
param 0 12 0 # Delay to process a cell (uSec): 0
param 155 12 0 # Switching Slot time (Mbit/s): 155
param 100 12 0 # Output q_size (cells, -1=inf): 100
param 0 12 0 # High Threshold, Q Cong. Flag (cells): 0
param 0 12 0 # Low Threshold, Q Cong. Flag (cells): 0
param 1 12 0 # Logging every (ticks) (e.g. 1, 100): 1
pflags 2a 4 #Cells Received: 0
pflags 2a 4 #Cell Drop %: 0
pflags 2a 4 #Cells in VBR Q to LK0: 0
pflags 2a 4 #Cells dropped in VBR Q to LK0: 0
pflags 2a 4 #Cells in ABR Q to LK0: 0
pflags 2a 4 #Cells Dropped in ABR Q to LK0: 0
pflags 2a 1 #Congestion for Link LK0: FALSE

component 'TE0' BTE 250 150
param 'TE0' 32 0 # TE0
param -1 12 0 # Max Output Queue Size(-1=inf): -1
param 1 12 0 # Logging every (ticks) (e.g. 1, 100): 1
pflags 2a 4 #Cells Received: 0
pflags 2a 4 #Cells in VBR Q to LK0: 0
pflags 2a 4 #Cells dropped in VBR Q to LK0: 0
pflags 2a 4 #Cells in ABR Q to LK0: 0
pflags 2a 4 #Cells Dropped in ABR Q to LK0: 0

component 'GER1' VBRCONNECTION 180 420
param 'GER1' 32 0 # GER1
param 1.16096750 12 0 # Bit Rate (Mbits/s): 1.16096750
param 5190 12 0 # Mean Burst Length (usecs): 5190
param 34810 12 0 # Mean Interval Between Bursts (usecs): 34810
param 1 12 0 # Start time (usecs): 1
param 1500 12 0 # Number of Mbits to be sent: 1500
pflags 2a 4 #Celulas Enviadas para ca': 0
pflags 2a 4 #Celulas Recebidas: 0
pflags 2e 4 # Tamanho Subintervalo (ms): 0
pflags 2a 4 # Subintervalo numero: 0
pflags 2a 4 # Perda Media (%): 0
pflags 2a 4 # Atraso Medio (usecs): 0
pflags 2a 4 # Interv. Conf. 99% (-/+ usecs): 0
pflags 2a 4 # Jitter Medio (usecs): 0
pflags 2a 4 # Interv. Conf. 99% (-/+ usecs): 0
pflags 22 0

component 'REC1' VBRCONNECTION 186 99
param 'REC1' 32 0 # REC1
param 0 12 0 # Bit Rate (Mbits/s): 0
param 0 12 0 # Mean Burst Length (usecs): 0
param 0 12 0 # Mean Interval Between Bursts (usecs): 0
param 0 12 0 # Start time (usecs): 0
param 0 12 0 # Number of Mbits to be sent: 0
pflags 2a 4 #Celulas Enviadas para ca': 0

```

```

pflags 2a 4      #Celulas Recebidas: 0
pflags 2a 4      # Tamanho Subintervalo (ms): 0
pflags 6a 4      # Subintervalo numero: 0
pflags 2a 4      # Perda Media (%): 0
pflags 2a 4      # Atraso Medio (usecs): 0
pflags 2a 4      # Interv. Conf. 99% (-/+ usecs): 0
pflags 2a 4      # Jitter Medio (usecs): 0
pflags 2a 4      # Interv. Conf. 99% (-/+ usecs): 0
pflags 22 0

```

```

component 'GER_ss' SCONNECTION 175 88
param 'GER_ss' 32 0      # GER_ss
param 10 12 0      # Mean Bit Rate (Mbits/s): 10
param 500 12 0      # Timegran (e.g. 100) (usecs): 500
param 1 12 0      # Start time (usecs): 1
param 0.8 12 0      # Hurst param H (0.5<=H<1): 0.8
pflags 2a 4      #Num Sent: 0
pflags 2a 4      #Output Rate (Mbps): 0
pflags 2a 4      #Celulas Enviadas para ca': 0
pflags 2a 4      #Celulas Recebidas: 0
pflags 2e 4      # Tamanho Subintervalo (ms): 0
pflags 2a 4      # Subintervalo numero: 0
pflags 2a 4      # Perda Media (%): 0
pflags 2a 4      # Atraso Medio (usecs): 0
pflags 2a 4      # Interv. Conf. 99% (-/+ usecs): 0
pflags 2a 4      # Jitter Medio (usecs): 0
pflags 2a 4      # Interv. Conf. 99% (-/+ usecs): 0
pflags 22 0

```

```

component 'REC_ss' SCONNECTION 173 382
param 'REC_ss' 32 0      # REC_ss
param 0 12 0      # Mean Bit Rate (Mbits/s): 0
param 500 12 0      # Timegran (e.g. 100) (usecs): 500
param 0 12 0      # Start time (usecs): 0
param 0.8 12 0      # Hurst param H (0.5<=H<1): 0.8
pflags 2a 4      #Num Sent: 0
pflags 2a 4      #Output Rate (Mbps): 0
pflags 2a 4      #Celulas Enviadas para ca': 0
pflags 2a 4      #Celulas Recebidas: 0
pflags 2e 4      # Tamanho Subintervalo (ms): 0
pflags 6a 4      # Subintervalo numero: 0
pflags 2a 4      # Perda Media (%): 0
pflags 2a 4      # Atraso Medio (usecs): 0
pflags 2a 4      # Interv. Conf. 99% (-/+ usecs): 0
pflags 2a 4      # Jitter Medio (usecs): 0
pflags 2a 4      # Interv. Conf. 99% (-/+ usecs): 0
pflags 22 0

```

```

component 'LK0' LINK 250 210
param 'LK0' 32 0      # LK0
param 1.16097 12 0      # Link Speed (Mbits/sec): 1.16097
pflags 2 0
pflags 2 0
pflags 2 0
pflags 2 0
pflags 2 0
pflags 2 0
pflags 2 0
param 1 12 0      # Distance (Km): 1
pflags 2a 4      #Link rate (Mbit/s) to TE0: 0

```



```
pflags 2a 4 #Link rate (Mbit/s) to Switch: 0
```

```
neighbor1 'TE0' 'LK0'  
neighbor1 'LK0' 'TE0'  
neighbor1 'LK0' 'Switch'  
neighbor1 'Switch' 'LK0'
```

```
neighbor1 'GER1' 'Switch'  
neighbor1 'TE0' 'REC1'  
neighbor1 'REC1' 'TE0'  
neighbor1 'Switch' 'GER1'
```

```
neighbor1 'GER_ss' 'Switch'  
neighbor1 'TE0' 'REC_ss'  
neighbor1 'REC_ss' 'TE0'  
neighbor1 'Switch' 'GER_ss'
```

```
routel 'GER1' 'Switch' 'LK0' 'TE0' 'REC1'  
routel 'REC_ss' 'TE0' 'LK0' 'Switch' 'GER_ss'
```

```

#####
## Data: 26/07/2001 ##
## Autora: Tatiana Rotava Schabbach ##
#####
## SIMULAÇÕES FINAIS - OBJETIVOS: ##
## 1 - ANALISAR OCORRÊNCIA DE ATRASOS E PERDAS NO FLUXO ABRSS, ##
## AUMENTANDO O NUMERO DE FLUXOS ##
## VBR POISSON (de 1 ate 25 video conferência) E VARIANDO A BANDA##
## (n x PCR, assim sempre sobra banda para os dados - ##
## [GON 2001]) E OS BUFFERS (100, 1000 e 10000). ##
## ##
## 2 - CONTABILIZAR AS PERDAS E ATRASOS DOS FLUXOS VBR, SENDO ##
## POSSIVEL TER (OU NAO) A VERIFICACAO DO TRATAMENTO PRIORITARIO ##
## DA REDE ATM PARA FLUXOS REAL-TIME ##
#####
## ##
## ROTINA PARA QUE A CADA EXECUÇÃO (de todas as simulações) NÃO ##
## SOBRESCREVA NO ARQUIVO DE SAIDA ANTERIOR. ##
## ASSIM EM CADA EXECUCAO SERA GERADO UM NOVO LOG COM O NUMERO ##
## DO PROCESSO (DO LINUX) ##
## NO FINAL DO NOME DO ARQUIVO DE RESULTADO ##
#####

arq_resultado="resultado_simulacao_tat"

if [ -f $arq_resultado ]
then
    arq_resultado="$arq_resultado.$$"
fi

#Arquivos de configurações: 30

#Cenário 1:conf_sim_vbr_1_100, conf_sim_vbr_1_1000,
conf_sim_vbr_1_10000
#Cenário 2:conf_sim_vbr_2_100, conf_sim_vbr_2_1000,
conf_sim_vbr_2_10000
#Cenário 3:conf_sim_vbr_3_100, conf_sim_vbr_3_1000,
conf_sim_vbr_3_10000
#Cenário 4:conf_sim_vbr_4_100, conf_sim_vbr_4_1000,
conf_sim_vbr_4_10000
#Cenário 5:conf_sim_vbr_5_100, conf_sim_vbr_5_1000,
conf_sim_vbr_5_10000
#Cenário 6:conf_sim_vbr_6_100, conf_sim_vbr_6_1000,
conf_sim_vbr_6_10000
#Cenário 7:conf_sim_vbr_8_100, conf_sim_vbr_8_1000,
conf_sim_vbr_8_10000
#Cenário 8:conf_sim_vbr_10_100, conf_sim_vbr_10_1000,
conf_sim_vbr_10_10000
#Cenário 9:conf_sim_vbr_15_100, conf_sim_vbr_15_1000,
conf_sim_vbr_15_10000
#Cenário 10:conf_sim_vbr_25_100, conf_sim_vbr_25_1000,
conf_sim_vbr_25_10000

#####
## Chama o script feito em gawk passando os seguintes parâmetros:##
## cenario, num vbr (banda = (num vbr vezes PCR)), buffers e nome##
## do arquivo de resultados ##
#####

executa_sim_vf 1 1 100 $arq_resultado
executa_sim_vf 1 1 1000 $arq_resultado

```

```
executa_sim_vf 1 1 10000 $arq_resultado

executa_sim_vf 2 2 100 $arq_resultado
executa_sim_vf 2 2 1000 $arq_resultado
executa_sim_vf 2 2 10000 $arq_resultado

executa_sim_vf 3 3 100 $arq_resultado
executa_sim_vf 3 3 1000 $arq_resultado
executa_sim_vf 3 3 10000 $arq_resultado

executa_sim_vf 4 4 100 $arq_resultado
executa_sim_vf 4 4 1000 $arq_resultado
executa_sim_vf 4 4 10000 $arq_resultado

executa_sim_vf 5 5 100 $arq_resultado
executa_sim_vf 5 5 1000 $arq_resultado
executa_sim_vf 5 5 10000 $arq_resultado

executa_sim_vf 6 6 100 $arq_resultado
executa_sim_vf 6 6 1000 $arq_resultado
executa_sim_vf 6 6 10000 $arq_resultado

executa_sim_vf 7 8 100 $arq_resultado
executa_sim_vf 7 8 1000 $arq_resultado
executa_sim_vf 7 8 10000 $arq_resultado

executa_sim_vf 8 10 100 $arq_resultado
executa_sim_vf 8 10 1000 $arq_resultado
executa_sim_vf 8 10 10000 $arq_resultado

executa_sim_vf 9 15 100 $arq_resultado
executa_sim_vf 9 15 1000 $arq_resultado
executa_sim_vf 9 15 10000 $arq_resultado

executa_sim_vf 10 25 100 $arq_resultado
executa_sim_vf 10 25 1000 $arq_resultado
executa_sim_vf 10 25 10000 $arq_resultado
```

```

#####
##  Data: 27/07/2001                                     ##
##  Autora: Tatiana Rotava Schabbach                    ##
#####
##  Script para:                                         ##
##  =====                                             ##
##      - Executar "n" Simulacoes com NIST ATM          ##
##  #####                                              ##
##  Entrada:                                             ##
##      - Numero do cenario simulado (1 a 10)           ##
##      - Numero de conexoes VBR (1, 2, 3, 4, 5, 6, 8, 10, ##
##        15 ou 25)                                     ##
##      - Tamanho do buffer (100, 1000 ou 10000 células) ##
##      - Nome do arquivo de saida dos resultados (acumula ##
##        todas as simulações em um mesmo arquivo para   ##
##        posteriormente gerar gráfico                  ##
#####

gawk -v cenario=$1 -v numvbr=$2 -v buffers=$3 -v saida=$4 '

    BEGIN {
        conf_temp=sprintf("conf_sim_vbr_%s_%s",numvbr,buffers)

# PCR para servico de video conferencia, conforme [GON 2001] e [ROB
96] #

        PCR = 1.16096750
        semente[1] = 963360530
        semente[2] = 963360531
        semente[3] = 963360532
        semente[4] = 963360533
        semente[5] = 963360534
        semente[6] = 963360535
        semente[7] = 963360536
        semente[8] = 963360537
        semente[9] = 963360538
        semente[10] = 963360539
        print "\n\n"
        print "-----X-----"
    -----"
        print "\t\tCenário ("cenario") Simulado: Dados (SS) + "
numvbr " Video Conferência (VBR)"
        print "\t\tConexões: " numvbr + 1
        print "\t\tCapacidade do Enlace: " numvbr * PCR
        print "\t\tCapacidade do Buffer: " buffers
        print "\t\tArquivo de Configuração: " conf_temp
        print "-----X-----"
    -----"

# este arquivo conterá as linhas finais alteradas das 10 replicações
para
# posterior cálculo de médias e IC

        arq_final="final_" cenario "_" buffers

        print "\n\n\n"
        print "====> Resultados das replicações gravados no
arquivo: " arq_final
        print "\n\n\n"

```

```

        for (rep = 1; rep <= 10; rep +=1) {

            Executa_Simulacao()

        }

        Calcula_Medias_IC()

        close (saida)
        close (arq_final)

        print "\n\n\n"
        print "=> Resultado do cálculo das medias finais e IC
gravados no arquivo: " saida
        print "\n\n"

    }

function Executa_Simulacao() {

    # chama o simulador com a versão que gera o sim-log com linhas 888 e
    999 e sem delay de comutação

        comando = "/home/tatiana/NIST/final-ubr-
generic/sim_v3_semdelaycomutacao -x -v -s " semente[rep] " -jitter 1 -
o sim_log -int 500000 -w 8000000 " conf_temp " 48000000"
        system(comando)

    # pega as ultimas 6 linhas do arquivo de log e joga em um arquivo
    temporario

        comando = "tail -5 sim_log > temp"
        system(comando)

    # pega a primeira linha (resultado 888) e a última (resultado 999) e
    joga em arquivo temporário2

        comando = "grep -v \\# temp > temp2"
        system(comando)

        $0 = "vazio"
        eof = getline < "temp2"
        banda = numvbr * PCR
        while (eof != 0) {

            # retira os campos 999 e 888, CLR, CTD e CDV e acrescenta a esta linha
            mais
            # informacoes e grava no arquivo final deste cenário

                linha = $1 " " $8 " " $9 " " $10 " " banda " "
            buffers " " numvbr
                print linha >> arq_final
                eof = getline < "temp2"
            }

            close ("temp")
            close ("temp2")

            comando = "rm -f sim_log"
            system(comando)

```

```

comando = "rm -f temp"
system(comando)

comando = "rm -f temp2"
system(comando)

}

function Calcula_Medias_IC() {

    eof = getline < arq_final
    while (eof != 0) {

# efetua a soma e a soma-quadrada das 3 métricas nas 10 replicações
para cada tipo de fluxo

        if ($1 == "888") {
            CLR_ss=CLR_ss+$2
            CLR_QUA_ss=CLR_QUA_ss+($2**2)
            CTD_ss=CTD_ss+$3
            CTD_QUA_ss=CTD_QUA_ss+($3**2)
            CDV_ss=CVD_ss+$4
            CDV_QUA_ss=CDV_QUA_ss+($4**2)
        }
        else {
            CLR_vbr=CLR_vbr+$2
            CLR_QUA_vbr=CLR_QUA_vbr+($2**2)
            CTD_vbr=CTD_vbr+$3
            CTD_QUA_vbr=CTD_QUA_vbr+($3**2)
            CDV_vbr=CVD_vbr+$4
            CDV_QUA_vbr=CDV_QUA_vbr+($4**2)
        }
        eof = getline < arq_final
    }

# calcula a média das 10 replicações para cada uma das 3 métricas de
interesse
# para cada tipo de fluxo - self-similar e vbr

    CLR_ss_medio=CLR_ss/10
    CTD_ss_medio=CTD_ss/10
    CDV_ss_medio=CDV_ss/10

    CLR_vbr_medio=CLR_vbr/10
    CTD_vbr_medio=CTD_vbr/10
    CDV_vbr_medio=CDV_vbr/10

# calcula a variância e o desvio-padrão de cada métrica para cada tipo
de fluxo

    Var_CLR_ss=((10*CLR_QUA_ss)-(CLR_ss**2))/90
    Desvio_CLR_ss=Var_CLR_ss**0.5
    Var_CTD_ss=((10*CTD_QUA_ss)-(CTD_ss**2))/90
    Desvio_CTD_ss=Var_CTD_ss**0.5
    Var_CDV_ss=((10*CDV_QUA_ss)-(CDV_ss**2))/90
    Desvio_CDV_ss=Var_CDV_ss**0.5

    Var_CLR_vbr=((10*CLR_QUA_vbr)-(CLR_vbr**2))/90

```

```

Desvio_CLR_vbr=Var_CLR_vbr**0.5
Var_CTD_vbr=((10*CTD_QUA_vbr)-(CTD_vbr**2))/90
Desvio_CTD_vbr=Var_CTD_vbr**0.5
Var_CDV_vbr=((10*CDV_QUA_vbr)-(CDV_vbr**2))/90
Desvio_CDV_vbr=Var_CDV_vbr**0.5

# calcula o valor limite para o intervalo de confiança para cada uma
das métricas
# para cada tipo de fluxo

    Erro_CLR_ss=3.25*(Desvio_CLR_ss/10**0.5)
    Erro_CTD_ss=3.25*(Desvio_CTD_ss/10**0.5)
    Erro_CDV_ss=3.25*(Desvio_CDV_ss/10**0.5)

    Erro_CLR_vbr=3.25*(Desvio_CLR_vbr/10**0.5)
    Erro_CTD_vbr=3.25*(Desvio_CTD_vbr/10**0.5)
    Erro_CDV_vbr=3.25*(Desvio_CDV_vbr/10**0.5)

# grava a saída em arquivos distintos para posterior geração de
gráficos

    linhafinal888="888 " CLR_ss_medio " " CTD_ss_medio " "
CDV_ss_medio " " Erro_CLR_ss " " Erro_CTD_ss " " Erro_CDV_ss " "
numvbr*PCR " " buffers " " numvbr
    print linhafinal888 >> saida "_ss"

    linhafinal999="999 " CLR_vbr_medio " " CTD_vbr_medio " "
CDV_vbr_medio " " Erro_CLR_vbr " " Erro_CTD_vbr " " Erro_CDV_vbr " "
numvbr*PCR " " buffers " " numvbr
    print linhafinal999 >> saida "_vbr"

}

,

# para gerar os gráficos foi preciso ordenar os arquivos de saída
fazendo:
# sort -n +8 +9 -o resultado_simulacao_tat_ss resultado_ss
# sort -n +8 +9 -o resultado_simulacao_tat_vbr resultado_vbr

```

Exemplo de arquivo de saída FINAL_1_100 (arq_final no script):

```

888 0.9019 43370.14 6858.58 1.16097 100 1
999 0.0000 180.89 95.67 1.16097 100 1
888 0.8991 42565.66 6291.07 1.16097 100 1
999 0.0000 176.82 87.67 1.16097 100 1
888 0.9018 43411.95 7159.64 1.16097 100 1
999 0.0000 184.30 100.21 1.16097 100 1
888 0.9009 42875.90 6266.14 1.16097 100 1
999 0.0000 190.41 93.96 1.16097 100 1
888 0.8991 42213.45 5810.62 1.16097 100 1
999 0.0000 184.48 99.74 1.16097 100 1
888 0.8996 42432.72 5893.69 1.16097 100 1
999 0.0000 198.77 96.35 1.16097 100 1
888 0.9006 42742.95 6421.26 1.16097 100 1
999 0.0000 195.37 104.16 1.16097 100 1
888 0.9000 42598.16 6040.99 1.16097 100 1
999 0.0000 187.03 91.46 1.16097 100 1
888 0.9017 43258.63 6578.53 1.16097 100 1
999 0.0000 175.11 93.02 1.16097 100 1
888 0.8994 42788.51 6656.06 1.16097 100 1
999 0.0000 177.65 96.58 1.16097 100 1

```

```
# Data: 11/09/2001
```

```
# Autora: Tatiana Rotava Schabbach
```

```
# Batch para chamar os scripts geradores dos gráficos dos resultados
```

```

clr_ss
clr_ss_ic
ctd_ss
ctd_ss_ic
cdv_ss
cdv_ss_ic

```

```

ctd_vbr
ctd_vbr_ic
cdv_vbr
cdv_vbr_ic

```

```

#####
#      Script para chamar gnuplot      #
#      Data: 11/09/2001                #
#      Autora: Tatiana Rotava Schabbach #
#      Chamada: ./clr_ss               #
#####

```

```

arq_saida="resultado_simulacao_tat_ss"
titulo="(CLR) Perda Self Similar      X      Conexões"

```

```

#####
# $1 = 888
# $2 = CLR
# $3 = CTD
# $4 = CDV
# $5 = Erro CLR
# $6 = Erro CTD
# $7 = Erro CDV
# $8 = banda
# $9 = buffers

```



```

# $10 = numvbr
#####

p01='-u ($10):($9>100 ? 1/0 : $9>0 ? $2 : 1/0) ti "100" with
linespoint lw 6.5 ps 2.5'
p02='-u ($10):($9>1000 ? 1/0 : $9>100 ? $2 : 1/0) ti "1.000" with
linespoint lw 4.5 ps 2.0'
p03='-u ($10):($9>10000 ? 1/0 : $9>1000 ? $2 : 1/0) ti "10.000" with
linespoint lw 2.5 ps 1.5'

pl "$p01" "$p02" "$p03" -title "$titulo" -f def_graf_CLR $arq_saida

#####
#      Script para chamar gnuplot      #
#      Data: 11/09/2001                #
#      Autora: Tatiana Rotava Schabbach #
#      Chamada: ./clr_ss_ic            #
#####

arq_saida="resultado_simulacao_tat_ss"
titulo="(CLR) Perda Self Similar com IC      X      Conexões"

#####
# $1 = 888
# $2 = CLR
# $3 = CTD
# $4 = CDV
# $5 = Erro CLR
# $6 = Erro CTD
# $7 = Erro CDV
# $8 = banda
# $9 = buffers
# $10 = numvbr
#####

p01='-u ($10):($9>100 ? 1/0 : $9>0 ? $2 : 1/0) notitle with line lt 1
lw 3'
p02='-u ($10):($9>1000 ? 1/0 : $9>100 ? $2 : 1/0) notitle with line lt
2 lw 2'
p03='-u ($10):($9>10000 ? 1/0 : $9>1000 ? $2 : 1/0) notitle with line
lt 3 lw 1'

p01i='-u ($10):($9>100 ? 1/0 : $9>0 ? $2 : 1/0):($5) ti "100" with
yerrorbars lt 1 lw 20 '
p02i='-u ($10):($9>1000 ? 1/0 : $9>100 ? $2 : 1/0):($5) ti "1.000"
with yerrorbars lt 2 lw 15 '
p03i='-u ($10):($9>10000 ? 1/0 : $9>1000 ? $2 : 1/0):($5) ti "10.000"
with yerrorbars lt 3 lw 10 '

pl "$p01" "$p01i" "$p02" "$p02i" "$p03" "$p03i" -title "$titulo" -f
def_graf_CLR $arq_saida

#####
#      Script com parametros para o gnuplot      #
#      Data: 11/09/2001                #
#      Autora: Tatiana Rotava Schabbach #
#####

set xlabel "Número de Conexões de Video Conferência"

```

```
set ylabel "CLR (Razão)"  
#set yrange [0:1]  
#set xrange [1:25]  
#set nogrid  
set xtics 1  
set key outside  
#set size 0.9,0.9
```

A2.2 - Simulador NS-2

```

# Data: 21/08/2001
# Autora: Tatiana Rotava Schabbach
#####
# SIMULAÇÕES FINAIS - OBJETIVOS:
# 1 - ANALISAR OCORRÊNCIA DE ATRASOS E PERDAS NO FLUXO SS, AUMENTANDO O NUMERO
#     DE FLUXOS VBR POISSON (de 1 ate 25 video conferência) E VARIANDO A BANDA
#     (n x PCR, assim sempre sobra banda para os dados - [GON 00]) E OS
#     BUFFERS (100, 1000 e 10000 pacotes).
#
# 2 - CONTABILIZAR AS PERDAS E ATRASOS DOS FLUXOS VBR, SENDO POSSIVEL TER (OU
#     NAO) A VERIFICACAO DO TRATAMENTO PRIORITARIO DA REDE ATM PARA FLUXOS
#     REAL-TIME
#####

# Cenários simulados: 10
# Variação dentro de cada cenário: 3
# Replicações por cenário: 10
# Total de simulações: 300
# Total de traces gerados: 300

#####
# Batch para chamar o script feito em gawk passando os seguintes parâmetros:
# cenario, num vbr (banda = (num vbr vezes PCR)), buffers
#####

#executa_sim_vp 1 1 100
#executa_sim_vf 1 1 100
#executa_sim_vf 1 1 1000
#executa_sim_vf 1 1 10000

#executa_sim_vf 2 2 100
#executa_sim_vf 2 2 1000
#executa_sim_vf 2 2 10000

#executa_sim_vf 3 3 100
#executa_sim_vf 3 3 1000
#executa_sim_vf 3 3 10000

#executa_sim_vf 4 4 100
#executa_sim_vf 4 4 1000
#executa_sim_vf 4 4 10000

#executa_sim_vf 5 5 100
#executa_sim_vf 5 5 1000
#executa_sim_vf 5 5 10000

#executa_sim_vf 6 6 100
#executa_sim_vf 6 6 1000
#executa_sim_vf 6 6 10000

#executa_sim_vf 7 8 100
#executa_sim_vf 7 8 1000
#executa_sim_vf 7 8 10000

#executa_sim_vf 8 10 100
#executa_sim_vf 8 10 1000
#executa_sim_vf 8 10 10000

executa_sim_vf 9 15 100
executa_sim_vf 9 15 1000
executa_sim_vf 9 15 10000

executa_sim_vf 10 25 100
executa_sim_vf 10 25 1000

```

```

executa_sim_vf 10 25 10000

## Data: 21/08/2001 ##
## Autora: Tatiana Rotava Schabbach ##
#####
## Script para: ##
## ===== ##
## - Executar "n" Simulacoes com NS ##
#####
## Entrada: ##
## - Numero do cenario simulado (1 a 10) ##
## - Numero de conexoes VBR (1, 2, 3, 4, 5, 6, 8, 10, 15 ou 25) ##
## - Tamanho do buffer (100, 1000 ou 10000 pacotes) ##
#####

gawk -v cenario=$1 -v numvbr=$2 -v buffers=$3 '

    BEGIN {

        saida = "result_sim_ns_tat"

# PCR para servico de video conferencia, conforme [GON 2001] e [ROB 96]

        PCR = 1.16096750e6

# Define os valores de banda que o fluxo da video conferência vai precisar
# reservar para atingir qualidade de serviço [GON 01, página 96, Tabela 4.9]
# Conexões VBR: 1, 2, 3, 4, 5, 6, 8, 10, 15 e 25 (10 cenários)

        reserva[1]= 1.16096750e6
        reserva[2]= 1.91559638e6
        reserva[3]= 2.26388663e6
        reserva[4]= 2.55412850e6
        reserva[5]= 2.90241875e6
        reserva[6]= 3.13461225e6
        reserva[7]= 3.71509600e6
        reserva[8]= 4.35362813e6
        reserva[9]= 5.65971656e6
        reserva[10]= 7.98165156e6

        print "\n\n"
        print "-----X-----"
        print "\t\tCenário ("cenario") Simulado: Dados (SS) + " numvbr "
Video Conferência (VBR)"
        print "\t\tConexões: " numvbr + 1
        print "\t\tCapacidade do Enlace: " numvbr * PCR
        print "\t\tCapacidade do Buffer: " buffers " pacotes"
        print "\t\tBanda reservada para vídeos: " reserva[cenario]
        print "-----X-----"

        arq_final="final_" cenario "_" buffers
        resv = reserva[cenario]
        Executa_Simulacao()
        Calcula_Medias_IC()

        close (saida)

        print "\n\n\n"
        print "=====> Resultados gravados no arquivo: " saida
        print "\n\n"

    }

function Executa_Simulacao() {

        for (rep = 1; rep <= 10; rep += 1) {

```

```

        semente = rep
        print "====> Número da replicacao e semente geradora: "
semente
        comando = "ns6_v2 diss_1.tcl " cenario " " numvbr " " buffers
" " semente " " resv
        system(comando)
        trace = "tr_" cenario "_" numvbr "_" buffers "_" rep
        comando = "java AnalisaTrace -src 0 -dst 1 -int 500000 -o
sim_log " trace
        system(comando)
# pega as ultimas 5 linhas do arquivo de log e joga em um arquivo temporario

        comando = "tail -5 sim_log > temp"
        system(comando)

# pega a primeira linha (resultado 888) e a última (resultado 999) e joga em
arquivo temporário2

        comando = "grep -v \\# temp > temp2"
        system(comando)

        $0 = "vazio"
        eof = getline < "temp2"
        while (eof != 0) {

# retira os campos 999 e 888, CLR, CTD e CDV e acrescenta a esta linha mais
# informacoes e grava no arquivo final deste cenário

                linha = $1 " " $2 " " $3 " " $4 " " buffers " " numvbr
                print linha >> arq_final
                eof = getline < "temp2"
        }

        close ("temp")
        close ("temp2")

        comando = "rm -f " trace
        system(comando)

        comando = "rm -f sim_log"
        system(comando)

        comando = "rm -f temp"
        system(comando)

        comando = "rm -f temp2"
        system(comando)
    }

}

function Calcula_Medias_IC() {

        eof = getline < arq_final
        while (eof != 0) {

# efetua a soma e a soma-quadrada das 3 métricas nas 10 replicações para cada
tipo de fluxo

                if ($1 == "888") {
                        CLR_ss=CLR_ss+$2
                        CLR_QUA_ss=CLR_QUA_ss+($2**2)
                        CTD_ss=CTD_ss+$3
                        CTD_QUA_ss=CTD_QUA_ss+($3**2)
                        CDV_ss=CVD_ss+$4
                        CDV_QUA_ss=CDV_QUA_ss+($4**2)
                }
        }
}

```

```

else {
    CLR_vbr=CLR_vbr+$2
    CLR_QUA_vbr=CLR_QUA_vbr+($2**2)
    CTD_vbr=CTD_vbr+$3
    CTD_QUA_vbr=CTD_QUA_vbr+($3**2)
    CDV_vbr=CVD_vbr+$4
    CDV_QUA_vbr=CDV_QUA_vbr+($4**2)
}
eof = getline < arq_final
}

# calcula a média das 10 replicações para cada uma das 3 métricas de interesse
# para cada tipo de fluxo - self-similar e vbr

CLR_ss_medio=CLR_ss/10
CTD_ss_medio=CTD_ss/10
CDV_ss_medio=CDV_ss/10

CLR_vbr_medio=CLR_vbr/10
CTD_vbr_medio=CTD_vbr/10
CDV_vbr_medio=CDV_vbr/10

# calcula a variância e o desvio-padrão de cada métrica para cada tipo de
fluxo

Var_CLR_ss=((10*CLR_QUA_ss)-(CLR_ss**2))/90
Desvio_CLR_ss=Var_CLR_ss**0.5
Var_CTD_ss=((10*CTD_QUA_ss)-(CTD_ss**2))/90
Desvio_CTD_ss=Var_CTD_ss**0.5
Var_CDV_ss=((10*CDV_QUA_ss)-(CDV_ss**2))/90
Desvio_CDV_ss=Var_CDV_ss**0.5

Var_CLR_vbr=((10*CLR_QUA_vbr)-(CLR_vbr**2))/90
Desvio_CLR_vbr=Var_CLR_vbr**0.5
Var_CTD_vbr=((10*CTD_QUA_vbr)-(CTD_vbr**2))/90
Desvio_CTD_vbr=Var_CTD_vbr**0.5
Var_CDV_vbr=((10*CDV_QUA_vbr)-(CDV_vbr**2))/90
Desvio_CDV_vbr=Var_CDV_vbr**0.5

# calcula o valor limite para o intervalo de confiança para cada uma das
métricas
# para cada tipo de fluxo considerando nível de confiança de 99% e 9 (I-
1) graus de
# liberdade (constante tirada da tabela t-students (t0.995,9)=3.25)

Erro_CLR_ss=3.25*(Desvio_CLR_ss/10**0.5)
Erro_CTD_ss=3.25*(Desvio_CTD_ss/10**0.5)
Erro_CDV_ss=3.25*(Desvio_CDV_ss/10**0.5)

Erro_CLR_vbr=3.25*(Desvio_CLR_vbr/10**0.5)
Erro_CTD_vbr=3.25*(Desvio_CTD_vbr/10**0.5)
Erro_CDV_vbr=3.25*(Desvio_CDV_vbr/10**0.5)

# grava a saída em arquivos distintos para posterior geração de gráficos

linhafinal888="888 " CLR_ss_medio " " CTD_ss_medio " "
CDV_ss_medio " " Erro_CLR_ss " " Erro_CTD_ss " " Erro_CDV_ss " " buffers " "
numvbr
print linhafinal888 >> saida "_ss"
linhafinal999="999 " CLR_vbr_medio " " CTD_vbr_medio " "
CDV_vbr_medio " " Erro_CLR_vbr " " Erro_CTD_vbr " " Erro_CDV_vbr " " buffers "
" numvbr
print linhafinal999 >> saida "_vbr"
}

```

```

,

# para gerar os gráficos foi preciso ordenar os arquivos de saida fazendo:
# sort -n +8 +9 -o result_sim_ns_tat_ss resultado_ss
# sort -n +8 +9 -o result_sim_ns_tat_vbr resultado_vbr

# Data: 23/08/2001
# Autora: Tatiana Rotava Schabbach
# Entrada: ns nome.tcl cenario numvbr buffers semente resv ([lindex
$argv 0..4])

#puts "O nome do script tcl é \"$argv0\""
#puts "Existem $argc argumentos: $argv"
#puts "O primeiro (argv 0) argumento é o número do cenário: [lindex $argv 0]"
#puts "O segundo (argv 1) argumento é o número de vbr's: [lindex $argv 1]"
#puts "O terceiro (argv 2) argumento é o tamanho do buffer: [lindex $argv 2]"
#puts "O quarto (argv 3) argumento é o número da replicação e semente
geradora: [lindex $argv 3]"
#puts "O quinto (argv 4) argumento é a banda reservada: [lindex $argv 4]"
#puts ""

# Cria o objeto simulador

set ns [new Simulator]

# Cria arquivo para analise da vazao para o fluxo ss

set f0 [open out0.tr w]

# Cria arquivos para analise da vazao para os fluxos das vídeos

for {set i 1} {$i <= [lindex $argv 1]} {incr i} {
    set f($i) [open out$i.tr w]
}

# Define o nome do trace par ao fluxo auto-similar de acordo com a taxa do
link

set trace_ss "trace_ss[lindex $argv 0].bin"
#puts "Nome do trace do fluxo ss: $trace_ss"

# Define nomes dos arquivos de saida

set tr_file "tr_[lindex $argv 0]_[lindex $argv 1]_[lindex $argv 2]_[lindex
$argv 3]"
#puts "Nome do trace de saida: $tr_file"

set namtr_file "namtr_[lindex $argv 0]_[lindex $argv 1]_[lindex $argv
2]_[lindex $argv 3]"
#puts "Nome do trace para animacao: $namtr_file"

## parametros do link RSVP

# banda em megabyte
set pcr 1.1609675067e6
set banda [expr [lindex $argv 1] * $pcr]
#puts "Capacidade do link: $banda"
# atraso em milisegundos
set atraso_lk10ms
# banda reservable em percentual
set bw_resv 0.8
# mensagens rsvp em bits/s
set mes_rsvp 650
# fila para best-effort em bytes
set fila_be 50000
# admission control algorithm
set adc Param
# estimator adc

```

```

set est Null

## parâmetros para reserva

# taxa em bits/s
set bw_resv      [expr [lindex $argv 4]/[lindex $argv 1]]
#puts "Banda reservada por fluxo de vídeo: $bw_resv"
# tamanho do bucket em bytes
set bucket_tam  50000
# time to live
set ttl 3

Agent/RSVP set noisy_ 3

Agent/RSVP instproc upcall-path { sid rate bucket sender} {
    $self reserve $sid FF $rate $bucket $sender
    puts "upcall-path reserve $sid $rate $bucket $sender"
}

Agent/RSVP instproc upcall-resv { sid rate bucket sender} {
    puts "Reserva aceita $sid $rate $bucket $sender"
}

# rotina de finalização

proc finish {} {
    global ns namtr_file argv
    $ns flush-trace
    # exec ./nam $namtr_file &
    # Call xgraph to display the results
    # exec xgraph out0.tr -geometry 800x400 &
    # for {set i 1} {$i <= [lindex $argv 1]} {incr i} {
    #     puts "O segundo (argv 1) argumento é o número de vbr's:
[lindex $argv 1]"
    #     exec xgraph out$i.tr -geometry 800x400 &
    # }
    exit 0
}

# rotina para criação dos traces

proc create-trace {} {
    global ns tr_file namtr_file
    set trfd [open $tr_file w]
    $ns trace-all $trfd
    # if {$namtr_file != " "} {
    #     $ns namtrace-all [open $namtr_file w]
    # }
    return $trfd
}

# rotina para criação da topologia

proc create-topology {} {
    global ns banda atraso_lk bw_resv mes_rsvp fila_be adc est
    global rot term

    set rot [$ns node]
    set term [$ns node]

    # criacao do link com variação de parametros de acordo com o num de conexoes
    # e de tamanho de buffer

    $ns duplex-rsvp-link $rot $term $banda $atraso_lk $bw_resv $mes_rsvp
    $fila_be $adc $est

```



```

}

# MAIN #

set trfd [create-trace]

create-topology

#Set Queue Size of link (n2-n3) to buffers (expresso em pacotes)
# podera ser 100, 1000 ou 10000 pacotes (passado por argumento [lindex $argv
2])

$ns queue-limit $rot $term [lindex $argv 2]

# diferentes sementes em cada uma das 10 replicações, mas
# semente iguais em diferentes cenários

ns-random [lindex $argv 3]

#ou esta outra forma

#set rng [new RNG]
#$rng seed [lindex $argv 3]

set counter 0
# Start the procedure which logs the rate for the incoming flow on node 0
$ns at 0.0 "record"

# coloca agentes rsvp em cada nodo

set rsvp0 [$rot add-rsvp-agent]
set rsvpl [$term add-rsvp-agent]

# cria o fluxo self-similar a partir de um arquivo de trace gerado

set sender_ss [new Agent/UDP]
$ns attach-agent $rot $sender_ss
set rec_ss [new Agent/LossMonitor]
$ns attach-agent $term $rec_ss
$ns connect $sender_ss $rec_ss
$sender_ss set fid_1
puts "Este é o flow_id do agente sender_ss: [$sender_ss set fid_]"

set tfile [new Tracefile]
$tfile filename $trace_ss
set flow_ss [new Application/Traffic/Trace]
$flow_ss attach-tracefile $tfile
$flow_ss attach-agent $sender_ss

# cria os fluxos vbr

for {set i 1} {$i <= [lindex $argv 1]} {incr i} {

# utilizou-se agente UDP para evitar o controle que o agente TCP tem e que é
implementado
# neste simulador. Versões mais recentes do NS já possibilitam desabilitar este
controle.

    set udp_($i) [new Agent/UDP]
    $ns attach-agent $rot $udp_($i)
    $udp_($i) set fid_ [expr $i+1]
    puts "Este é o flow_id do agente udp do fluxo vbr: [$udp_($i) set fid_]"
    set null_($i) [new Agent/LossMonitor]
    $ns attach-agent $term $null_($i)
    $ns connect $udp_($i) $null_($i)

# os tempos estão expressos em millisegundos

```

```

set flow_vc($i) [new Application/Traffic/Exponential]
$flow_vc($i) attach-agent $udp_($i)

# tamanho do pacote é fixo.
# para facilitar foi definido múltiplo de 48 (área de dados de 1 célula em
ATM)
# 1o. caso:
# 48*5=240 bytes = 1920 bits
# 2o. caso:
# 48 bytes = 384 bits
# os tempos estão expressos em millisegundos

$flow_vc($i) set packetSize_ 240
$flow_vc($i) set burst_time_ 5.190ms
$flow_vc($i) set idle_time_ 34.810ms
$flow_vc($i) set rate_ $pcr

# cria sessões independentes para cada fluxo de video e inicia transmissao vbr
# ver estes parâmetros
# a banda reservada foi dividida aritmeticamente entre os fluxos vbr (partes
iguais)

set sessao_id [$rsvp0 session $term [expr [$udp_($i) set fid_]]]
$ns at 0.0 "$rsvp0 sender $sessao_id $bw_resv $bucket_tam $ttl"
$ns at 0.0 "$flow_vc($i) start"

# para transmissao vbr
# $ns at 48.0 "stop $sessao_id"
# $ns at 48.0 "$flow_vc($i) stop"

}

# inicia transmissao self-similar
$ns at 0.0 "$flow_ss start"

# para transmissao self-similar

$ns at 48.0 "$flow_ss stop"

$ns at 50.0 "finish"

# A simple procedure to log some interesting data. The procedure
# writes the data into the corresponding files, then schedules itself
# a second later.
proc record {} {
    global ns counter rec_ss null_ f0 f argv
    set time 1.0
    set now [$ns now]
    set bw_ss [$rec_ss set bytes_]
    puts $f0 "$now [expr $bw_ss*8/$time]"
    $rec_ss set bytes_ 0
    for {set i 1} {$i <= [lindex $argv 1]} {incr i} {
        set bw($i) [$null_($i) set bytes_]
# para expressar em bits/s
        puts $f($i) "$now [expr ($bw($i)*8)/$time]"
        $null_($i) set bytes_ 0
    }
    $ns at [expr $now+$time] "record"
    puts "Time: $now"
    incr counter
}

proc stop { sid } {
    global rsvp0
    $rsvp0 release $sid
    puts "Reserva cancelada $sid"
}

ns run

```

Exemplo de arquivo de saída FINAL_1_100 (arq_final no script):

```
888 0.000000 0.013041 0.001421 100 1
999 0.000000 0.012802 0.001320 100 1
888 0.000000 0.012440 0.001298 100 1
999 0.000000 0.012276 0.000737 100 1
888 0.000000 0.012732 0.001474 100 1
999 0.000000 0.012186 0.000710 100 1
888 0.000000 0.012687 0.001556 100 1
999 0.000000 0.012393 0.000899 100 1
888 0.068118 0.253529 0.038021 100 1
999 0.197423 0.184545 0.079503 100 1
888 0.000000 0.013296 0.001883 100 1
999 0.000000 0.012516 0.000981 100 1
888 0.073531 0.251702 0.035454 100 1
999 0.182339 0.194843 0.062406 100 1
888 0.000000 0.012508 0.001344 100 1
999 0.000000 0.012407 0.001051 100 1
888 0.083160 0.242194 0.032018 100 1
999 0.156204 0.196519 0.079824 100 1
888 0.000000 0.013532 0.001582 100 1
999 0.000000 0.012540 0.001163 100 1
```

Anexo 3 Execução dos Simuladores

A3.1 – Simulador NIST

O simulador NIST ATM/HFC possui dois modos de execução: interface gráfica ou linha de comando (*background*). A execução da simulação via interface gráfica permite a configuração do cenário de rede a ser simulado, bem como acompanhar interativamente a situação da ocupação dos *buffers*, da taxa de transmissão dos enlaces, do número de células descartadas e de outros valores disponíveis. Para os dois modos podemos utilizar um arquivo texto de entrada para configuração do cenário de rede a ser simulado. A vantagem da execução via linha de comando (sem interface gráfica) é a rapidez na execução da simulação, chegando a ser vinte vezes mais rápido do que no modo gráfico. Os resultados da simulação para os dois modos são apresentados em um arquivo texto de saída, chamado geralmente de "SIM_LOG". No final da execução, também é gerado no formato texto um arquivo com a configuração do cenário de rede. Esse arquivo é chamado "SIM_SNAP" e tem como objetivo servir como entrada no simulador para configurar uma nova execução. Exemplo dos arquivos "SIM_LOG" e "SIM_SNAP" estão neste anexo.

O comando para execução do simulador e seus parâmetros são apresentados no exemplo a seguir:

```
sim -x -s 963360534 -o arquivo_saída -int 500000 -w 8000000 -jitter 2  
conf_temp 4800000
```

Onde,

sim é o nome do programa executável do NIST ATM/HFC;

-x é o parâmetro utilizado para especificar o modo *background* de execução;

-s é o parâmetro para informar a semente para o gerador de números aleatórios;

-o é o parâmetro que informa o nome do arquivo de saída dos resultados (se a opção não for informada, o nome default é SIM_LOG);

-int é o parâmetro para informar o tamanho do subintervalo em microssegundos. Esse parâmetro não é original da versão do NIST e foi implementado em [GON 2001];

-w é o parâmetro utilizado para informar o tamanho em microssegundos do tempo de instabilidade (*warmuptime*) que deverá ser aguardado após o início da simulação para começar a contabilização das variáveis observadas;

-jitter é o parâmetro utilizado para informar qual será a forma do cálculo do CDV na simulação. Os valores possíveis são: 1 ou 2. Esse parâmetro não é original da versão do NIST e foi implementado nesse trabalho;

conf_temp é o nome do arquivo de configuração do cenário de rede a ser simulado. Contém todas as informações das características dos elementos que compõem a rede. Por exemplo, para o comutador temos: quantidade de portas, quantidade de *buffer* por porta, velocidade dos enlaces, *delay* interno de comutação, e outros;

4800000 é o valor em microssegundos para a duração do tempo simulado. Quando o simulador atinge esse valor de tempo simulado, ele encerra a execução gravando os arquivos de saída SIM_LOG e SIM_SNAP.

A versão original do NIST encerra a execução do simulador somente quando o tempo simulado informado na linha de comando é alcançado. Em [GON 2001] foi adicionado no simulador a condição de também encerrar a simulação após a execução de no mínimo 30 subintervalos quando o intervalo de confiança do CTD tiver uma largura de borda¹ menor que 10%, condição esta que permanece válida para esta dissertação.

No final da execução da simulação, dois tipos de detalhes de informações (linha do arquivo de saída) foram implementados no arquivo SIM_LOG. Um tipo para mostrar os valores obtidos em cada subintervalo e outro, no final do arquivo, com o consolidado de todos os subintervalos. O formato das linhas de saída é apresentado a seguir. Esta implementação foi inicialmente feita em [GON 2001] e continua valendo para esta dissertação.

Após esse arquivo é tratado dentro de um script feito em “gawk”², onde é feito o armazenamento de alguns campos de interesse, sendo depois replicado 10 vezes o processo para cada cenário e, ao final é executado uma rotina, dentro do mesmo script, que faz os cálculos estatísticos sobre os valores resultantes das 10 replicações e armazena os dados finais em um arquivo chamado RESULTADO_SIMULACAO_TAT_SS para o fluxo auto-similar e RESULTADO_SIMULACAO_TAT_VBR para os fluxos VBR. Abaixo temos uma impressão exemplo destes arquivos de saída que são entrada para os scripts geradores dos gráficos apresentados nessa dissertação.

Linha detalhe de final de subintervalo (SIM_LOG):

```
# 2 'REC_ss'
# '2 Tick Num_Int CLR_Int CTD_Int CDV_Int Cel_Rec CLR_Med CTD_Med
CDV_Med CTD_IConf(99%) CDV_IConf(99%)'
1 850000000 1 0.0000 122.54 0.00 182 0.0000 122.54 0.00 0.00 0.00
A B C D E F G H I J K L
```

Onde,

A = Identificador da linha detalhe. É igual para todas as linhas detalhes de final de subintervalo.

B = Instante em ticks (ns) da impressão da linha de detalhe do intervalo

C = Número do subintervalo

D = Razão de perda de células do subintervalo corrente (porcentagem)

E = Média do delay (CTD) do intervalo corrente

F = Média do jitter (CDV) do intervalo corrente

G = Número de células do intervalo

H = Média da razão de perda de células entre todos os subintervalos

I = Média do delay entre todos os intervalos

J = Média do jitter entre todos os intervalos

¹ Largura de borda é a distância do centro até as bordas do intervalo de confiança. 10 % de largura corresponde a um intervalo de (0,9 x CTDmédio) a (1,1 x CTDmédio).

² Apresentado no Anexo 2.

K = Valor para cálculo do intervalo de confiança do delay. [I-K,I+K] 99%

L = Valor para cálculo do intervalo de confiança do jitter. [J-L,J+L] 99%

Linhas detalhes de final de simulação (SIM_LOG):

```
# 888 'RESULTADO FINAL CONEXOES SCONNECTION'
#      '888 Num_Cnx      | Cel_Env      | Cel_Rec      | Num_Int      | Tam_Int(us) |
Warmup(us) |
#      '      CLR_Med(%) | CTD_Med(us) | CDV_Med(us) | CTD_IC99(us) | CDV_IC99(us) |
T_Sim(s)   | T_Exec(s)   '
888 1 366047 35796 31 500000 8000000 0.9019 43370.14 6858.58 1094.48 964.42 24 23
# 999 'RESULTADO FINAL CONEXOES VBRCONNECTION'
#      '999 Num_Cnx      | Cel_Env      | Cel_Rec      | Num_Int      | Tam_Int(us) |
Warmup(us) |
#      '      CLR_Med(%) | CTD_Med(us) | CDV_Med(us) | CTD_IC99(us) | CDV_IC99(us) |
T_Sim(s)   | T_Exec(s)   '
999 1 6645 6645 31 500000 8000000 0.0000 180.89 95.67 15.15 11.92 24 23
A  B  C  D  E  F  G  H  I  J  K  L  M  N
```

Onde,

A = Identificação do parâmetro a ser gravado. É 888 para a linha detalhe de final de simulação para os fluxos auto-similares e é 999 para a linha detalhe de final de simulação para os fluxos vbr;

B = Número de conexões;

C = O total de células enviadas por todas as conexões, as células do período de *warmuptime* não são contabilizadas;

D = O total de células recebidas por todas as conexões, as células do período de *warmuptime* não são contabilizadas;

E = Quantidade de subintervalos que a simulação foi subdividida;

F = Tamanho do subintervalo em microssegundos;

G = Tamanho do período inicial (*warmuptime*) da simulação aguardado para começar a contabilizar os subintervalos estatísticos;

H = Média da razão de perda de células da simulação considerando todas as conexões;

I = Média do CTD da simulação considerando todas as conexões;

J = Média do CDV da simulação considerando todas as conexões;

K = Intervalo de confiança de 99% para a média do CTD [I-K , I+K]

L = Intervalo de confiança de 99% para a média do CDV [J-L , J+L]

M = Tempo simulado em segundos;

N = Tempo de execução da simulação em segundos;

Exemplo de arquivo de saída SIM_LOG.xxxx:

```

# 1 'REC1'
# '1 Tick Num_Int CLR_Int CTD_Int CDV_Int Cel_Rec CLR_Med CTD_Med CDV_Med CTD_IC99(us) CDV_IC99(us)'
# 2 'REC_ss'
# '2 Tick Num_Int CLR_Int CTD_Int CDV_Int Cel_Rec CLR_Med CTD_Med CDV_Med CTD_IC99(us) CDV_IC99(us)'
1 850000000 1 0.0000 122.54 0.00 182 0.0000 122.54 0.00 0.00 0.00
2 850000000 1 0.9052 42075.78 0.00 1187 0.9052 42075.78 0.00 0.00 0.00
1 900000000 2 0.0000 165.85 94.27 203 0.0000 144.20 94.27 55.44 120.67
2 900000000 2 0.8955 41943.54 4822.82 1166 0.9004 42009.66 4822.82 169.27 6173.22
1 950000000 3 0.0000 154.70 69.35 182 0.0000 147.70 81.81 33.24 72.20
2 950000000 3 0.9037 42950.71 5779.40 1187 0.9015 42323.34 5301.11 808.96 4578.52
1 1000000000 4 0.0000 198.97 89.46 145 0.0000 160.52 84.36 40.37 55.73
2 1000000000 4 0.8898 40744.66 5897.33 1224 0.8986 41928.67 5499.85 1161.05 3573.38
1 1050000000 5 0.0000 211.72 96.23 213 0.0000 170.76 87.33 40.80 46.35
2 1050000000 5 0.9079 43130.72 5632.90 1156 0.9004 42169.08 5533.11 1089.77 2873.60
1 1100000000 6 0.0000 193.08 119.84 264 0.0000 174.48 93.83 34.65 43.44
2 1100000000 6 0.9042 44732.97 6582.74 1105 0.9010 42596.40 5743.04 1410.11 2520.21
1 1150000000 7 0.0000 188.34 98.12 165 0.0000 176.46 94.55 29.72 37.43
2 1150000000 7 0.9012 42010.60 7013.39 1204 0.9011 42512.71 5954.77 1210.86 2280.43
1 1200000000 8 0.0000 203.33 96.10 176 0.0000 179.82 94.77 27.14 32.77
2 1200000000 8 0.8892 41755.06 7083.67 1193 0.8996 42418.01 6116.04 1076.30 2074.02
1 1250000000 9 0.0000 192.92 80.46 263 0.0000 181.27 92.98 24.22 28.91
2 1250000000 9 0.9106 45261.72 7360.77 1106 0.9008 42733.97 6271.63 1247.11 1916.32
1 1300000000 10 0.0000 171.29 109.39 340 0.0000 180.27 94.80 21.81 26.75
2 1300000000 10 0.9151 45153.52 8147.22 1029 0.9022 42975.93 6480.03 1275.89 1836.17
1 1350000000 11 0.0000 156.14 88.04 275 0.0000 178.08 94.13 20.52 24.20
2 1350000000 11 0.9092 48849.33 8761.53 1094 0.9029 43509.88 6708.18 1788.95 1795.36
1 1400000000 12 0.0000 167.64 56.35 311 0.0000 177.21 90.69 18.86 22.96
2 1400000000 12 0.9003 47183.02 10423.94 1058 0.9026 43815.97 7045.97 1811.35 1880.87
1 1450000000 13 0.0000 139.10 106.02 188 0.0000 174.28 91.97 18.90 21.59
2 1450000000 13 0.9043 42298.04 7493.06 1181 0.9028 43699.21 7083.23 1692.80 1742.09
1 1500000000 14 0.0000 161.10 104.50 322 0.0000 173.34 92.93 17.67 20.31
2 1500000000 14 0.9139 46993.68 7769.42 1048 0.9036 43934.53 7136.02 1679.02 1628.50
1 1550000000 15 0.0000 218.77 99.55 147 0.0000 176.37 93.41 18.18 19.04
2 1550000000 15 0.8989 41184.80 8221.37 1222 0.9033 43751.21 7213.54 1632.01 1540.29
1 1600000000 16 0.0000 164.44 65.65 172 0.0000 175.62 91.56 17.11 18.14
2 1600000000 16 0.8871 41856.68 3662.52 1197 0.9022 43632.80 6976.81 1556.41 1522.25
1 1650000000 17 0.0000 213.44 60.42 338 0.0000 177.85 89.61 17.05 17.47
2 1650000000 17 0.9151 46610.59 9449.71 1031 0.9030 43807.97 7131.36 1529.21 1495.50
1 1700000000 18 0.0000 204.51 92.98 249 0.0000 179.33 89.81 16.52 16.51
2 1700000000 18 0.9110 44674.63 7958.96 1120 0.9034 43856.12 7180.04 1447.02 1421.08
1 1750000000 19 0.0000 138.53 96.60 173 0.0000 177.18 90.19 16.57 15.70
2 1750000000 19 0.8970 43314.06 7801.99 1196 0.9031 43827.59 7214.60 1370.69 1351.23
1 1800000000 20 0.0000 184.77 107.54 131 0.0000 177.56 91.10 15.75 15.16
2 1800000000 20 0.8879 40419.48 4704.89 1238 0.9023 43657.18 7082.51 1371.57 1310.56
1 1850000000 21 0.0000 267.98 101.94 171 0.0000 181.87 91.64 18.60 14.54
2 1850000000 21 0.8995 41679.72 5389.07 1198 0.9022 43563.02 6997.84 1326.71 1257.24
1 1900000000 22 0.0000 181.96 105.54 149 0.0000 181.87 92.30 17.73 14.03
2 1900000000 22 0.9048 40921.60 5303.23 1220 0.9023 43442.95 6917.14 1301.77 1209.15
1 1950000000 23 0.0000 132.10 120.43 261 0.0000 179.71 93.58 17.82 13.88
2 1950000000 23 0.9016 44975.95 6379.84 1108 0.9023 43509.60 6892.72 1255.54 1155.65
1 2000000000 24 0.0000 195.73 72.22 171 0.0000 180.37 92.65 17.15 13.41
2 2000000000 24 0.8937 41754.86 6462.37 1198 0.9019 43436.49 6874.01 1216.57 1106.54
1 2050000000 25 0.0000 169.91 92.79 133 0.0000 179.95 92.66 16.49 12.87
2 2050000000 25 0.8946 40425.03 4445.44 1236 0.9016 43316.03 6772.82 1206.95 1083.78
1 2100000000 26 0.0000 234.03 129.69 162 0.0000 182.03 94.14 16.71 13.00
2 2100000000 26 0.9074 41391.96 4606.94 1207 0.9019 43242.03 6686.18 1174.98 1057.85
1 2150000000 27 0.0000 122.08 148.35 149 0.0000 179.81 96.22 17.05 13.66
2 2150000000 27 0.8904 40916.60 5131.03 1220 0.9014 43155.90 6626.37 1151.92 1025.33
1 2200000000 28 0.0000 203.17 95.24 262 0.0000 180.65 96.19 16.57 13.17
2 2200000000 28 0.9027 45111.57 8033.64 1108 0.9015 43225.75 6678.49 1124.33 999.52
1 2250000000 29 0.0000 207.34 98.44 353 0.0000 181.57 96.27 16.16 12.71
2 2250000000 29 0.9120 45819.95 9314.84 1016 0.9019 43315.20 6772.64 1108.77 997.27
1 2300000000 30 0.0000 174.14 81.09 242 0.0000 181.32 95.75 15.63 12.32
2 2300000000 30 0.9129 47022.57 9549.16 1127 0.9022 43438.78 6868.39 1116.91 997.11
1 2350000000 31 0.0000 168.11 93.46 153 0.0000 180.89 95.67 15.15 11.92
2 2350000000 31 0.8907 41311.02 6574.11 1216 0.9019 43370.14 6858.58 1094.48 964.42
# 888 'RESULTADO FINAL CONEXOES SCONNECTION'
# '888 Num_Cnx | Cel_Env | Cel_Rec | Num_Int | Tam_Int(us) | Warmup(us) |
# ' CLR_Med(%) | CTD_Med(us) | CDV_Med(us) | CTD_IC99(us) | CDV_IC99(us) | T_Sim(s) | T_Exec(s)'
888 1 366047 35796 31 500000 8000000 0.9019 43370.14 6858.58 1094.48 964.42 24 23
# 999 'RESULTADO FINAL CONEXOES VBRCONNECTION'
# '999 Num_Cnx | Cel_Env | Cel_Rec | Num_Int | Tam_Int(us) | Warmup(us) |
# ' CLR_Med(%) | CTD_Med(us) | CDV_Med(us) | CTD_IC99(us) | CDV_IC99(us) | T_Sim(s) | T_Exec(s)'
999 1 6645 6645 31 500000 8000000 0.0000 180.89 95.67 15.15 11.92 24 23

```

Exemplo de arquivo de saída RESULTADO_SIMULACAO_TAT_SS:

```
888 0.90041 42825.8 665.606 9.31626e-08 12184.8 3.33119e+06 1.16097 100 1
888 0.90041 428056 2945.78 9.31626e-08 1.21302e+06 5.34502e+07 1.16097 1000 1
888 0.90041 4.2637e+06 3575.25 9.31626e-08 1.1476e+08 8.54525e+07 1.16097 10000 1
888 0.80094 21361.9 284.625 1.96536e-07 1401.89 755502 2.32193 100 2
888 0.80094 214153 1061.5 1.96536e-07 138983 1.15707e+07 2.32193 1000 2
888 0.80094 2.13777e+06 1730 1.96536e-07 1.04869e+07 2.78978e+07 2.32193 10000 2
888 0.70043 14140.7 192.776 5.02328e-07 972.244 278076 3.4829 100 3
888 0.70043 142114 758.101 5.02328e-07 100877 4.18755e+06 3.4829 1000 3
888 0.70043 1.42066e+06 1308.44 5.02328e-07 8.57797e+06 1.81251e+07 3.4829 10000 3
888 0.60059 10542.7 133.396 4.36838e-07 234.031 144817 4.64387 100 4
888 0.60059 106536 538.959 4.36838e-07 21557.9 2.47938e+06 4.64387 1000 4
888 0.60059 1.06588e+06 1142.79 4.36838e-07 1.59225e+06 1.50579e+07 4.64387 10000 4
888 0.50036 8297.17 108.874 1.15053e-06 588.122 97768.1 5.80484 100 5
888 0.50028 85067.2 444.735 1.23339e-06 11822.8 1.53131e+06 5.80484 1000 5
888 0.50028 852842 1275.33 1.23339e-06 860350 1.5104e+07 5.80484 10000 5
888 0.40148 6701.25 94.469 3.34792e-07 554.639 83969.7 6.9658 100 6
888 0.40024 70574.1 359.799 6.09863e-07 4423.66 1.03274e+06 6.9658 1000 6
888 0.40024 709887 1222.55 6.09863e-07 377168 9.5591e+06 6.9658 10000 6
888 0.21263 4268.9 123.108 6.77033e-07 158.477 153410 9.28774 100 8
888 0.19872 50873.2 369.921 4.58692e-07 18911.4 1.36982e+06 9.28774 1000 8
888 0.19872 528687 887.525 4.58692e-07 251165 8.22036e+06 9.28774 10000 8
888 0.07688 2284.74 164.72 2.89015e-06 309.121 205334 11.6097 100 10
888 0.02315 21000.8 1606.41 7.2577e-06 56370.7 1.88092e+07 11.6097 1000 10
888 0 141347 3601.08 0 2.82222e+08 6.77146e+07 11.6097 10000 10
888 0.00035 143.892 19.666 1.0366e-08 30.9961 3541.43 17.4145 100 15
888 0 151.907 20.478 0 74.9308 4172.66 17.4145 1000 15
888 0 151.907 20.478 0 74.9308 4172.66 17.4145 10000 15
888 0 2.342 0.372 0 0.000265172 1.1385 29.0242 100 25
888 0 2.342 0.372 0 0.000265172 1.1385 29.0242 1000 25
888 0 2.342 0.372 0 0.000265172 1.1385 29.0242 10000 25
```


Exemplo de arquivo de saída RESULTADO_SIMULACAO_TAT_VBR:

```
999 0 185.083 9.658 0 4.64518 747.354 1.16097 100 1
999 0 185.083 9.658 0 4.64518 747.354 1.16097 1000 1
999 0 185.083 9.658 0 4.64518 747.354 1.16097 10000 1
999 0 97.396 5.158 0 0.468764 220.121 2.32193 100 2
999 0 97.396 5.158 0 0.468764 220.121 2.32193 1000 2
999 0 97.396 5.158 0 0.468764 220.121 2.32193 10000 2
999 0 66.815 3.57 0 0.13601 100.611 3.4829 100 3
999 0 66.815 3.57 0 0.13601 100.611 3.4829 1000 3
999 0 66.815 3.57 0 0.13601 100.611 3.4829 10000 3
999 0 50.686 2.849 0 0.0713412 60.3184 4.64387 100 4
999 0 50.686 2.849 0 0.0713412 60.3184 4.64387 1000 4
999 0 50.686 2.849 0 0.0713412 60.3184 4.64387 10000 4
999 0 40.854 2.186 0 0.0517089 38.4558 5.80484 100 5
999 0 40.949 2.186 0 0.0375755 38.9314 5.80484 1000 5
999 0 40.949 2.186 0 0.0375755 38.9314 5.80484 10000 5
999 0 34.435 1.839 0 0.0319653 27.1003 6.9658 100 6
999 0 34.248 1.824 0 0.0122234 27.3188 6.9658 1000 6
999 0 34.248 1.824 0 0.0122234 27.3188 6.9658 10000 6
999 0 25.529 1.411 0 0.0171714 16.079 9.28774 100 8
999 0 26.058 1.381 0 0.0155249 15.5724 9.28774 1000 8
999 0 26.058 1.381 0 0.0155249 15.5724 9.28774 10000 8
999 0 20.14 1.158 0 0.00428242 11.2942 11.6097 100 10
999 0 20.639 1.133 0 0.00288027 10.4714 11.6097 1000 10
999 0 20.8 1.105 0 0.00712589 10.1814 11.6097 10000 10
999 0 9.93 0.839 0 0.000422833 5.8115 17.4145 100 15
999 0 9.926 0.839 0 0.000355967 5.80039 17.4145 1000 15
999 0 9.926 0.839 0 0.000355967 5.80039 17.4145 10000 15
999 0 4.08 0.476 0 0.000127833 1.82725 29.0242 100 25
999 0 4.08 0.476 0 0.000127833 1.82725 29.0242 1000 25
999 0 4.08 0.476 0 0.000127833 1.82725 29.0242 10000 25
```

A3.2 – Simulador NS

A versão do simulador NS utilizada nesta dissertação foi a “b6” e a esta versão foi acrescido os módulos referentes ao protocolo RSVP descritos em [GRE 2001]. Foram necessários pequenos ajustes em alguns módulos para corrigir problemas de compilação antes do seu uso efetivo.

Abaixo está descrito um passo a passo das etapas executadas até a disponibilização do simulador para executar os scripts desse trabalho.

1. Alteração das seguintes linhas 339 e 786 do arquivo TclPosixStr.c:
de **#ifdef EOPNOTSUPP**
para **#if defined (EOPNOTSUPP) && (!defined(ENOTSUP)) || (EOPNOTSUPP != ENOTSUP)**
2. Alteração do arquivo `utils.cc` na linha 469 o argumento 3 do comando `'getsockname(int, sockaddr *, socklen_t *)'`
3. Estando no diretório `/ns-allinone-2.1b6`, então descompacte os módulos do RSVP usando: **`tar xvfz rsvp-patch-ns-2.1b6-0.5c.tar.gz`**
4. Para instalar a versão do NS digite **`./install`** (A compilação deverá demorar alguns minutos)

O comando para execução do simulador e seus parâmetros são apresentados no exemplo a seguir:

```
ns6 diss_1.tcl cenário numvbr buffers semente resv
```

Onde,

ns6 é o nome do programa executável do simulador NS-2 versão b6;

diss_1.tcl é o nome do script desenvolvido nessa dissertação;

cenário é o número do cenário que será simulado;

numvbr é o número de conexões VBR simultâneas;

buffers é o tamanho da fila do roteador;

semente é o número utilizado para gerar as sementes no simulador;

resv é a banda reservada por conexão VBR.

Ao final de cada simulação é gerado um trace que contém todos os eventos ocorridos durante a simulação. Para a análise desse arquivo, e cálculo efetivo das métricas de interesse desse trabalho, foi utilizado um programa feito em JAVA.

O comando para execução do analisador e seus parâmetros são apresentados no exemplo a seguir:

```
java AnalisaTrace -src 0 -dst 1 -int 500000 -o sim_log trace
```

Onde,

java é o comando que chama a máquina virtual JAVA;

AnalisaTrace é o nome do executável feito em JAVA;

-src é o parâmetro que indica o nodo origem;

-dst é o parâmetro que indica o nodo destino. As métricas são calculadas para este par de nodos;

-int é o parâmetro para informar o tamanho do subintervalo em microssegundos;

-o informa o nome do arquivo de saída do analisador;

trace é o nome do trace de entrada (saída do simulador).

No final da execução do analisador, no final do arquivo de saída é apresentado as linhas com o resultado consolidado de todos os subintervalos computados. O formato das linhas de saída é apresentado a seguir. Esta implementação segue a idéia implementada no outro simulador para facilitar a padronização da parte gráfica dos resultados finais.

Após esse arquivo é tratado dentro de um script feito em “gawk”¹, onde é feito o armazenamento de alguns campos de interesse. Todo o processo é replicado 10 vezes para cada cenário e, ao final é executado uma rotina, dentro do mesmo script, que faz os cálculos estatísticos sobre os valores resultantes das 10 replicações e armazena os dados finais em um arquivo chamado RESULT_SIM_NS_TAT_SS para o fluxo auto-similar e RESULT_SIM_NS_TAT_VBR para os fluxos VBR. Abaixo temos uma impressão exemplo destes arquivos de saída que são entrada para os scripts geradores dos gráficos apresentados nessa dissertação.

¹ Apresentado no Anexo 2.

Exemplo de arquivo de saída RESULT_SIM_NS_TAT_SS:

```

888 0.0224809 0.0837661 0.0001582 0.0373845 0.117323 0.0209609 100 1
888 0.0224809 0.0837661 0.0001582 0.0373845 0.117323 0.0209609 1000 1
888 0.0224809 0.0837661 0.0001582 0.0373845 0.117323 0.0209609 10000 1
888 0.0473368 0.0732392 0.0009661 0.0628917 0.0819581 0.00696673 100 2
888 0.0473368 0.0732392 0.0009661 0.0628917 0.0819581 0.00696673 1000 2
888 0.0473368 0.0732392 0.0009661 0.0628917 0.0819581 0.00696673 10000 2
888 0.0389979 0.0458369 4.23e-05 0.0580308 0.0519069 0.00323113 100 3
888 0.0389979 0.0458369 4.23e-05 0.0580308 0.0519069 0.00323113 1000 3
888 0.0389979 0.0458369 4.23e-05 0.0580308 0.0519069 0.00323113 10000 3
888 0.0480179 0.0432086 0.0003247 0.0637247 0.0431111 0.00202372 100 4
888 0.0480179 0.0432086 0.0003247 0.0637247 0.0431111 0.00202372 1000 4
888 0.0480179 0.0432086 0.0003247 0.0637247 0.0431111 0.00202372 10000 4
888 0.0453291 0.0342927 2.3e-05 0.0616731 0.0321188 0.00153912 100 5
888 0.0453291 0.0342927 2.3e-05 0.0616731 0.0321188 0.00153912 1000 5
888 0.0453291 0.0342927 2.3e-05 0.0616731 0.0321188 0.00153912 10000 5
888 0.0444634 0.0295493 1.8e-05 0.0615496 0.0263571 0.00103475 100 6
888 0.0444634 0.0295493 1.8e-05 0.0615496 0.0263571 0.00103475 1000 6
888 0.0444634 0.0295493 1.8e-05 0.0615496 0.0263571 0.00103475 10000 6
888 0.0229699 0.0179769 1.09e-05 0.049969 0.0165 0.000525116 100 8
888 0.0229699 0.0179769 1.09e-05 0.049969 0.0165 0.000525116 1000 8
888 0.0229699 0.0179769 1.09e-05 0.049969 0.0165 0.000525116 10000 8
888 0.0284619 0.0185309 1.09e-05 0.053414 0.0176434 0.000451669 10000 8
888 0.0284107 0.0172202 8.6e-06 0.0563622 0.0135652 0.000421212 100 10
888 0.0284107 0.0172202 8.6e-06 0.0563622 0.0135652 0.000421212 1000 10
888 0.0284107 0.0172202 8.6e-06 0.0563622 0.0135652 0.000421212 10000 10
888 0.0284107 0.0161902 8.6e-06 0.0563622 0.0145585 0.000420228 10000 10
888 0.0223461 0.0135396 4.1e-06 0.041741 0.00622078 0.000271331 100 15
888 0.0223461 0.0135396 4.1e-06 0.041741 0.00622078 0.000271331 1000 15
888 0.0223461 0.0135396 4.1e-06 0.041741 0.00622078 0.000271331 10000 15
888 0.0169211 0.011572 2.9e-06 0.029464 0.00252021 0.000142136 100 25
888 0.0169211 0.011572 2.9e-06 0.029464 0.00252021 0.000142136 1000 25
888 0.0169211 0.011572 2.9e-06 0.029464 0.00252021 0.000142136 10000 25

```

Exemplo de arquivo de saída RESULT_SIM_NS_TAT_VBR:

```

999 0.0535966 0.0663027 0.0001163 0.0892663 0.0891791 0.0441304 100 1
999 0.0535966 0.0663027 0.0001163 0.0892663 0.0891791 0.0441304 1000 1
999 0.0535966 0.0663027 0.0001163 0.0892663 0.0891791 0.0441304 10000 1
999 0 0.0113237 2.17e-05 0 0.000454248 0.000189656 100 2
999 0 0.0113237 2.17e-05 0 0.000454248 0.000189656 1000 2
999 0 0.0113237 2.17e-05 0 0.000454248 0.000189656 10000 2
999 5.42e-05 0.0109172 1e-05 0.00017615 0.000403381 0.000187726 100 3
999 5.42e-05 0.0109172 1e-05 0.00017615 0.000403381 0.000187726 1000 3
999 5.42e-05 0.0109172 1e-05 0.00017615 0.000403381 0.000187726 10000 3
999 0.0005241 0.0109146 9.4e-06 0.00170332 0.000891148 0.00049557 100 4
999 0.0005241 0.0109146 9.4e-06 0.00170332 0.000891148 0.00049557 1000 4
999 0.0005241 0.0109146 9.4e-06 0.00170332 0.000891148 0.00049557 10000 4
999 0 0.0105377 7.4e-06 0 0.000152477 6.70402e-05 100 5
999 0 0.0105377 7.4e-06 0 0.000152477 6.70402e-05 1000 5
999 0 0.0105377 7.4e-06 0 0.000152477 6.70402e-05 10000 5
999 0.0019462 0.0108691 5.2e-06 0.00349452 0.000906941 0.000347069 100 6
999 0.0019462 0.0108691 5.2e-06 0.00349452 0.000906941 0.000347069 1000 6
999 0.0019462 0.0108691 5.2e-06 0.00349452 0.000906941 0.000347069 10000 6
999 0.0007624 0.0104227 2.1e-06 0.0024778 0.000438153 0.000179304 100 8
999 0.0007624 0.0104227 2.1e-06 0.0024778 0.000438153 0.000179304 1000 8
999 0.0009265 0.0104508 2.1e-06 0.00301113 0.00052854 0.000214173 10000 8
999 0.0030588 0.0107051 2.2e-06 0.00613892 0.000956156 0.000238128 100 10
999 0.0030588 0.0107051 2.2e-06 0.00613892 0.000956156 0.000238128 1000 10
999 0.0030588 0.0107051 2.2e-06 0.00613892 0.000956156 0.000238128 10000 10
999 0.002836 0.0103701 1.1e-06 0.00513034 0.000413963 0.00015092 100 15
999 0.002836 0.0103701 1.1e-06 0.00513034 0.000413963 0.00015092 1000 15
999 0.002836 0.0103701 1.1e-06 0.00513034 0.000413963 0.00015092 10000 15
999 0.0022096 0.0102028 7e-07 0.00366198 0.000195093 6.59127e-05 100 25
999 0.0022096 0.0102028 7e-07 0.00366198 0.000195093 6.59127e-05 1000 25
999 0.0022096 0.0102028 7e-07 0.00366198 0.000195093 6.59127e-05 10000 25

```

Bibliografia

- [ARA 94] ARAS, Çağlan et al. Real-Time Communication in Packet-Switched Networks. **Proceedings of the IEEE**, New York, v. 82, n. 1, p. 122-139, Jan. 1994.
- [ARO 99] ARORA, Rakesh. **Voice over IP: Protocols and Standards**. 1999. Disponível em: <http://www.cis.ohio-state.edu/~jain/cis788-99/voip_protocols/index.html>. Acesso em: 1999.
- [BAN 96] BANKS, Jerry; CARSON, John S. ; NELSON, Barry. **Discrete-event system simulation**. 2nd ed. Upper Saddle River: Prentice-Hall, 1996. 548p.
- [BAS 98] BASHFORTH, Byron; WILLIAMSON, Carey. **Statistical Multiplexing of Self-Similar Video Streams: Simulation Study and Performance Results**. Canada: Department of Computer Science, University of Saskatchewan, 1998. Disponível em: <<http://citeseer.nj.nec.com/200969.html>>. Acesso em: 2001.
- [BER 98] BERGER, A. W.; WHITT W. Extending the effective bandwidth concept to networks with priority classes. **IEEE Communications Magazine**, New York, v. 36, n. 8, p. 78-83, Aug. 1998.
- [BER 98a] BERNET, Yoram et al. **A Framework for Differentiated Services**. 1998. Disponível em: <<http://www.ietf.org/proceedings/98aug/I-D/draft-ietf-diffserv-framework-00.txt>>. Acesso em: 2001.
- [BER 99] BERSON, Steven. **RSVP - ReSerVation Protocol**. 1999. Disponível em: <<http://www.isi.edu/div7/rsvp/rsvp.html>>. Acesso em: 1999.
- [BLA 95] BLACK, Uyless D. **ATM: foundation for broadband networks**. New Jersey: Prentice Hall, 1995. 426p.
- [BOR 95] BORDEN, M. et al. **Integration of Real-time Services in an IP-ATM Network Architecture: RFC-1821**. 1995. Disponível em: <<ftp://ftp.isi.edu/in-notes/rfc1821.txt>>. Acesso em: 2001.
- [BRA 94] BRADEN, Robert; CLARK, David; SHENKER, Scott. **Integrated Services in the Internet Architecture: an overview: RFC-1633**. 1994. Disponível em: <<ftp://ftp.isi.edu/in-notes/rfc1633.txt>>. Acesso em: maio 1998.
- [BRA 97] BRADEN, Robert et al. **Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification: RFC-2205**. 1997. Disponível em: <<ftp://ftp.isi.edu/in-notes/rfc2205.txt>>. Acesso em: maio 1998.
- [BUS 96] BUSSE, Ingo; DEFFNER, Bernd; SCHULZRINNE, Henning. Dynamic Qos Control of Multimedia Applications based on RTP. **Computer Communications**, Surrey, v. 19, n. 1, p.49-58, Jan. 1996.
- [CER 97] CEREDA, Ronaldo Luiz Dias et al. **ATM: o futuro das redes**. São Paulo: Makron Books: Brisa, 1997. 178p.

- [CLA 98] CLAYPOOL, Mark; RIEDL, John. **End-to-End Quality in Multimedia Applications**. Massachusetts: Worcester Polytechnic Institute, 1998. (Computer Science Technical Report Series, WPI-CS-TR-98-18). Disponível em: <<http://www.cs.wpi.edu/Resources/techreports>>. Acesso em: julho 2001.
- [CLA 2001] CLAYPOOL, Mark; CHUNG, Jae. **NS by Example**. Disponível em: <<http://www.cs.wpi.edu/Resources/techreports>>. Acesso em: julho 2001.
- [COM 98] COMER, Douglas E. **Interligação em rede com TCP/IP**. Rio de Janeiro: Campus, 1998. 672p.
- [DEL 99] DELFINO, Gardel Moreira. **Voz sobre IP – Seminário de Redes de Computadores**. Disponível em: <<http://www.gta.ufrj.br/~gardel/redes.rede.htm>>. Acesso em: dez.1999.
- [DOR 96] DORLING, Brian et al. **Internetworking over ATM: an introduction**. New Jersey: Prentice Hall, 1996. 260p.
- [DUT 95] DUTTON, Harry J. R.; LENHARD, Peter. **Asynchronous Transfer Mode (ATM): Technical Overview**. 2nd ed. New Jersey: Prentice Hall, 1995.
- [EXP 90] EXPERIMENTAL Internet Stream Protocol, Version 2 (ST-II): RFC 1190. 1990. Disponível em: <<http://www.scit.wlv.ac.uk/~c9451595/rfc/rfc11xx/>>. Acesso em: 22 ago. 1999.
- [FAN 99] FAN, Z. New Trends in ATM networks: a research view. **Computer Communications**, Surrey, v. 22, p.499-515, 1999.
- [FUR 99] FURINI, Marco; TOWSLEY, Don. **Real-Time traffic transmission over the internet**. [S.l.:s.n.], 1999. (Umass CMPSCI Technical Report 99-73). Disponível em: <http://www.rfai.li.univ-tours.fr/~crucianu/rm_etudes/mathe/etude.html>. Acesso em: 1999.
- [GAR 98] GARRETT, M.; BORDEN, M. **Interoperation of Controlled-Load Service and Guaranteed Service with ATM**: RFC-2381. 1998. Disponível em: < <http://www.doclib.org/rfc/rfc2381.html> >. Acesso em: 2001.
- [GEI 2000] GEIB, Rüdiger. **Differential Services for the Internet and ATM**. Disponível em: <<http://www.internet2.edu/qos/qbone/qsg/i2qos-geib-difs-atm-02.html>>. Acesso em: set. 2000.
- [GIR 99] GIROUX, Natalie; GANTI, Sudhakar. **Quality of Service in ATM Networks: state of the art traffic management**. New Jersey: Prentice Hall, 1999. 250p.
- [GOL98] GOLMIE, N. et al. **The NIST ATM/HFC Network Simulator Operation and Programming Guide Version 4.0**. 1998. Disponível em: <<http://isdn.ncsl.nist.gov>>. Acesso em: abril 2001.
- [GON 2001] GONÇALVES, Adriano Ramos. **Método para Planejamento de Capacidade de Redes ATM baseado em Simulação**. 2001. 125p. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.

- [GRE 2001] GREIS, Marc. **RSVP/ns**: an implementation of RSVP for the network simulator ns-2. [S.l.]: University of Bonn. Disponível em: <<http://www.lab.dit.upm.es/~labrst/00-01/p.sim/rsvpns.pdf>>. Acesso em: 2001.
- [HER 99] HERMANN, Ana Carolina. **Especificação de Qualidade de Serviço e Adaptação de Aplicações Multimídia Distribuídas**. 1999. 82p. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- [HSU 96] HSU, I.; WALRAND, J. Admission Control for Multi-class ATM traffic with overflow constraints. **Computer Networks and ISDN Systems Journal**, Amsterdam, v. 28, n. 13, p. 1739-1752, Oct. 1996.
- [IET 98] INTERNET ENGINEERING TASK FORCE. **Differentiated Services (diffserv)**. Disponível em: <<http://www.ietf.org/html-charters/diffserv.html>>. Acesso em: jun. 1998.
- [JAI 91] JAIN, Raj. **The art of computer systems performance analysis**: techniques for experimental design, measurement, simulation, and modeling. New York: Wiley, 1991. 685p.
- [JAI 96] JAIN, Raj. **Congestion Control and Traffic Management in ATM Networks**: Recent Advances and a Survey. [S.l.]: Ohio State University, Aug. 1996. Disponível em: <<http://www.cis.ohio-state.edu/~jain/papers/cnis.html>>. Acesso em: ago. 2001.
- [KAM 2000] KAMIENSKI, Carlos Alberto; SADOK, Djamel. Qualidade de Serviço na Internet. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, SBRC, 18., 2000, Belo Horizonte. **Minicurso 2**. Belo Horizonte: UFMG, 2000. 50p.
- [KAM 2000a] KAMIENSKI, Carlos Alberto; SADOK, Djamel. Engenharia de tráfego em uma rede de Serviços Diferenciados. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, SBRC, 18., 2000, Belo Horizonte, **Anais...** Belo Horizonte: UFMG, 2000. p.151-166.
- [KES 93] KESIDIS, G.; WALRAND, J.; CHANG, C. Effective bandwidths for multiclass Markov fluids and other ATM sources. **IEEE/ACM Transactions on Networking**, New York, v.1, n. 4, p. 424-428, Aug. 1993.
- [KUL 94] KULKARNI, V.; GUN, L.; CHIMENTO, P. Effective bandwidth vector for two-priority ATM traffic. In: IEEE INFOCOM, 1994. **Proceedings...** Toronto: [s.n.], 1994. p.1056-1064.
- [KUR 98] KUROSE, Jim et al. **Correlation of Packet Delay and Loss in the Internet**. Massachusetts: University of Massachusetts, Jan.1998. (Technical Report 98-11). Disponível em: <<http://www-net.cs.umass.edu/papers/papers.html>>. Acesso em: jul 2001.
- [LAK 97] LAKSHMAN, T. V.; MADHOW, Upamanyu. The Performance of TCP/IP for Networks with High Bandwidth-Delay Products and Random Loss. **IEEE/ACM Transactions on Networking**, New York, v. 5, n. 3, p.336-350, June 1997.

- [LAN 2000] LANGONA, Karen; CARVALHO, T. C. M. de B. Controle de qualidade de serviço para conexões rt-VBR em redes ATM. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, SBRC, 18., 2000, Belo Horizonte. **Anais...** Belo Horizonte: UFMG, 2000. p.337-351.
- [LAU 95] LAUER, Hugh C. On the Duality of Rate-based and Credit-based Flow Control. In: ACM SIGCOMM CONFERENCE, 1995. **Proceedings...** [S.l.:s.n.], 1995.
- [LEL 91] LELAND, W. E. ; WILSON, D. V. High Time-Resolution Measurement and Analyses of LAN Traffic: Implications for LAN Interconnection. In: IEEE INFOCOM, 1991. **Proceedings...** [S.l.:s.n.], 1991. p. 1360-1366.
- [LEL 94] LELAND, Will E. et al. On the Self-Similar Nature of Ethernet (Extended version). **IEEE/ACM Transactions on Networking**, New York, v.2, n.1, p.1-15, Feb. 1994.
- [LEO 2000] LEOCÁDIO, Márcio Augusto P.; RODRIGUES, P. H. de Aguiar. Uma Ferramenta para geração de tráfego e medição em ambiente de alto desempenho. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, SBRC, 18., 2000, Belo Horizonte. **Anais...** Belo Horizonte: UFMG, 2000. p. 321-336.
- [LIU 97] LIU, Chunlei. **Multimedia Over IP: RSVP, RTP, RTCP, RTSP**. Disponível em: <http://www.cis.ohio-state.edu/~jain/cis788-97/ip_multimedia.html>. Acesso em: jan. 2000.
- [MCD 2000] McDYSAN, D. **QoS & Traffic Management in IP & ATM Networks**. New York: McGraw-Hill, 2000. 457p.
- [MIL 2000] MILANI, Rogério. **QoS em Redes IP e ATM**. 2000. 134p. Monografia (Graduação em Análise de Sistemas) - Unisinos, São Leopoldo.
- [OLI 97] OLIVEIRA, Adriano L. I. de. **Modelos de Tráfego para a Multiplexação Estatística do Tráfego de Dados em Redes ATM**. Recife: UFPE, Departamento de Informática, 1997.
- [ONV 95] ONVURAL, Raif O. **Asynchronous Transfer Mode Networks. Performance Issues**. 2nd ed. Norwood, MA: Artech House, 1995.
- [OUS 94] OUSTERHOUT, John K. **Tcl and the Tk toolkit**. Reading: Addison-Wesley Publishing Company, 1994. 458p.
- [PAX 95] PAXSON, Vern. **Fast Approximation of Self-Similar Network Traffic**. Berkeley: Laurence Berkeley Laboratory, University of California, 1995.
- [PAX 98] PAXSON, Vern et al. **Framework for IP Performance Metrics**: RFC-2330. Disponível em: <<ftp://ftp.isi.edu/in-notes/rfc2330.txt>>. Acesso em: jun. 1998.
- [PEA 96] PEARSON, Murray et al. **Current Techniques for Measuring and Modeling ATM Traffic**. New Zealand: University of Waikato, 1996. 36p. Disponível em: <<http://atm.cs.waikato.ac.nz/wand/publications/CTMMATMT-abs.html>>. Acesso em: 19 abr. 2001.

- [PRY 95] PRYCKER, M. de. **Asynchronous Transfer Mode - Solution for Broadband ISDN**. 3rd ed. London: Prentice Hall, 1995. 380p.
- [ROB 96] ROBERTS, J. et al.(Ed.). **Broadband Networks Teletraffic: Performance Evaluation and Design of Broadband Multiservices networks**. Berlin: Springer Verlag, 1996. p.6-46.
- [ROS 93] ROSE, O. ; FRATER M. R. **A Comparison of Models for VBR Video Traffic Sources in B-ISDN**. [S.l.:s.n.], 1993. (Report, n. 72).
- [SCH 99] SCHEIBLER, M. D'. **Gerência de Performance de Serviços Multimídia Interativos**. 1999. 60p. Dissertação (Mestrado em Ciência da Computação) - Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- [SCH 96] SCHULZRINNE, Henning et al. **RTP: A Transport Protocol for Real-Time Applications: RFC-1889**. 1996. Disponível em: <ftp://ftp.isi.edu/in-notes/rfc1889.txt>. Acesso em: 1999.
- [SCH 97] SCHULZRINNE, Henning. The Impact of Resource Reservation for Real-Time Internet Services In: NATIONAL RESEARCH COUNCIL WORKSHOP ON INFORMATION SYSTEM TRUSTWORTHINESS, 1997, Washington. **Proceedings...** Disponível em: <http://www.cs.columbia.edu/~hgs/papers/Schu9610_Impact.ps.gz>. Acesso em: maio 1998.
- [SCH 97a] SCHULZRINNE, Henning. **Some Frequently Asked Questions about RTP**. 1997. Disponível em: <http://www.cs.columbia.edu/~hgs/rtp>. Acesso em: 1999.
- [SCH 98] SCHULZRINNE, Henning; RAO, Anup; LANPHIER, R. **Real Time Streaming Protocol (RTSP): RFC-2326**. 1998. Disponível em: <ftp://ftp.isi.edu/in-notes/rfc2326.txt>. Acesso em: maio 1998.
- [SHE 97] SHENKER, S.; PARTRIDGE, C.; GUERIN, R. **Specification of Guaranteed Quality of Service: RFC-2212**. 1997. Disponível em: <http://www.doclib.org/rfc/rfc2212.html>.
- [SHE 97a] SHENKER, S.; WROCLAWSKI, J. **General Characterization Parameters for Integrated Service Network Elements: RFC-2215**. 1997. Disponível em: <http://www.ietf.org/rfc/rfc2215.txt>.
- [SIL 2000] SILVA, Fernanda Mendes Ascensão. **Multiplexação Estatística de Fontes Heterogêneas em ATM usando o Modelo de Tráfego On/Off Pareto**. 2000. Dissertação (Mestrado em Ciência da Computação) - Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- [SIT 2001] SITOLINO, Claudio L. **VoIP: um estudo experimental**. 2001. Dissertação (Mestrado em Ciência da Computação) - Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- [SOA 90] SOARES, Luiz Fernando Gomes. **Modelagem e Simulação Discreta de Sistemas**. São Paulo: IME-USP, 1990. 250p.
- [SOA 95] SOARES, Luiz Fernando Gomes; SOUZA FILHO, Guido Lemos de; COLCHER, Sérgio. **Redes de Computadores: das lans, mans e wans as redes atm**. 2. ed. Rio de Janeiro: Campus, 1995. 705p.

- [SPO 91] SPOHN, Marcelo. Análise de Performance em redes de computadores. In: TELEMÁTICA - SIMPÓSIO DE REDES DE COMPUTADORES E SUAS APLICAÇÕES, 1991, Porto Alegre. **Anais...** [S.l.:s.n.], 1991. p.8.1-8.17.
- [STA 94] STAMOULIS, G. D. ; ANAGNOSTOU, M. E. ; GEORGANTAS A. D. Traffic Source Models for ATM Networks: a survey. **Computer Communications**, [S.l.], v.17, n.6, June 1994.
- [STA 98] STALLINGS, William. **High-speed networks: TCP/IP and ATM** design and principles. Upper Saddle River: Prentice-Hall, 1998. p.125-208.
- [STA 2000] STALLINGS, William. **Data and computer communications**. 6. ed. Upper Saddle River: Prentice-Hall, 2000. 810p.
- [SUN 98] SUN, Hairong; HUANG, Ke; LI, Lemin. Supporting IP on the ATM networks: an overview. **Computer Communications**, [S.l.], v. 21, n. 11, p.1020-1029, Aug. 1998.
- [TAN 97] TANENBAUM, Andrew. **Redes de Computadores**. 3.ed. Rio de Janeiro: Campus, 1997. 923p.
- [VIA 2000] VIANA, Aline C. et al. Perspectivas sobre Qualidade de Serviço nos Protocolos da Internet - Estudo de Caso: Aplicações de Vídeo sob Demanda. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, SBRC, 18.; WORKSHOP RNP2, 2., 2000, Belo Horizonte. **Anais...** Belo Horizonte: UFMG, 2000. p.1-12. Disponível em: <<http://www.rnp.br/newsgen/0007/art1.shtml>>. Acesso em: set. 2000.
- [WIL 95] WILLINGER, Walter et al. Self-Similarity Through High Variability: Statistical Analysis of Ethernet LAN Traffic at the source Level. In: SIGCOMM, 1995. **Proceedings...** Cambridge: [s.n.], 1995. p. 100-113.
- [WIL 97] WILLINGER, Walter et al. Self-Similarity Through High Variability: Statistical Analysis of Ethernet LAN Traffic at the source Level. **IEEE/ACM Transactions on Networking**, New York, v. 5, n. 1, p.71-86, Feb. 1997.
- [WRO 97] WROCLAWSKI, J. **The Use of RSVP with IETF Integrated Services: RFC-2210**. 1997. Disponível em: <<http://www.doclib.org/rfc/rfc2210.html>>. Acesso em: 2001.
- [WRO 97a] WROCLAWSKI, J. **Specification of the Controlled-Load Network Element Services: RFC-2211**. 1997. Disponível em: <<http://www.doclib.org/rfc/rfc2211.html>>. Acesso em: 2001.
- [ZAN 99] ZANIN, Fábio A. **O Ambiente de videoconferência sobre redes ATM**. 1999. 83p. Trabalho Individual (Mestrado em Ciência da Computação) - Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.