

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA  
COMPUTAÇÃO**

**Eduardo Bueno**

**Aperfeiçoamento do Desempenho do TCP em Links  
Assimétricos via Políticas de Escalonamento de Filas**

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

**Prof. Dr. Carlos Becker Westphall**  
Orientador

Florianópolis, outubro de 2004.

# **Aperfeiçoamento do Desempenho do TCP em Links Assimétricos via Políticas de Escalonamento de Filas**

Eduardo Bueno

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

---

Prof. Raul Sidnei Wazlawick, Dr.  
Coordenador

Banca Examinadora

---

Prof. Carlos Becker Westphall, Dr.  
Orientador

---

Prof. Rômulo da Silva Oliveira, Dr.

---

Prof. Arthur Ronald de V. Buchsbaum, Dr.

---

Prof. Roberto Willrich, Dr.

---

Prof. Mário Antonio Ribeiro Dantas, Dr.

## Sumário

<b>Sumário</b> .....	iii
<b>Agradecimentos</b> .....	vii
<b>Dedicatória</b> .....	viii
<b>Glossário</b> .....	xix
<b>Resumo</b> .....	xi
<b>Abstract</b> .....	xii
<b>Lista de Figuras</b> .....	xiv
<b>Capítulo 1 – Introdução</b> .....	14
<i>Trabalhos Correlatos</i> .....	15
<b>Capítulo 2 - O Acesso de Banda Larga</b> .....	17
2.1 <i>Sistemas de Acesso de Banda Larga</i> .....	17
2.1.1 <i>HDSL</i> .....	17
2.1.2 <i>SDSL</i> .....	17
2.1.3 <i>ADSL</i> .....	18
2.1.4 <i>RADSL</i> .....	19
2.1.5 <i>VDSL</i> .....	19
2.2 <i>O Serviço ADSL</i> .....	19
2.3 <i>Perspectivas para utilização do ADSL</i> .....	20
2.3.1 <i>Perspectiva para o provedor de serviço</i> .....	20
2.3.2 <i>Perspectiva para o usuário final</i> .....	21
2.4 <i>Oportunidades de Negócios</i> .....	22
2.4.1 <i>Acesso a Internet em alta Velocidade</i> .....	22
2.4.2 <i>Serviços de Interconexão</i> .....	22
2.4.3 <i>Negócios entre empresas</i> .....	22
2.4.4 <i>Serviços de entrega de Conteúdo</i> .....	23
2.5 <i>Arquitetura do Sistema ADSL</i> .....	23
2.5.1 <i>Rede de Acesso</i> .....	23
2.5.1.1 <i>Modem do Assinante</i> .....	23
2.5.1.2 <i>Rede Metálica</i> .....	24

2.5.1.3	<i>Concentradores de Acesso - DSLAM</i> .....	25
2.5.2	<i>Rede de Transporte</i> .....	25
2.5.2.1	<i>Terminação ATM - BRAS</i> .....	25
2.6	<i>Tecnologia ADSL</i> .....	26
2.3.1	<i>Modulação DMT</i> .....	27
2.3.2	<i>Modulação QAM</i> .....	28
<b>Capítulo 3 – Pilha de Protocolos</b> .....		<b>29</b>
3.1	<i>O ATM</i> .....	29
3.1.1	<i>Camada Física</i> .....	29
3.1.2	<i>Camada ATM</i> .....	30
3.1.3	<i>Camada de Adaptação 1 - AAL1</i> .....	31
3.1.4	<i>Camada de Adaptação 5 - AAL5</i> .....	31
3.1.5	<i>Conexões Lógicas</i> .....	31
3.2	<i>Point-to-Point Protocol</i> .....	32
3.2.1	<i>O Protocolo de Controle do Link</i> .....	33
3.2.2	<i>O Protocolo de Autenticação</i> .....	33
3.2.3	<i>O Protocolo de Controle de Camada de Rede</i> .....	34
3.3	<i>TCP/IP</i> .....	34
3.3.1	<i>Internet Protocol - IP</i> .....	34
3.3.2	<i>Protocolo TCP</i> .....	35
3.3.3	<i>Pacote TCP</i> .....	36
3.3.4	<i>O campo Options</i> .....	37
3.3.5	<i>Maximum Transmission Unit - MTU</i> .....	37
3.3.6	<i>Round Rip Time</i> .....	38
<b>Capítulo 4 - O Desempenho do TCP</b> .....		<b>39</b>
4.1	<i>Disciplina de Filas</i> .....	39
4.1.1	<i>FIFO</i> .....	40
4.1.2	<i>Priority Round Robin</i> .....	40
4.1.3	<i>Weighted Round Robin</i> .....	41
4.2	<i>O problema da Assimetria</i> .....	42
4.2.1	<i>Assimetria de Faixa</i> .....	42
4.2.2	<i>Assimetria de Tráfego</i> .....	42
4.3.3	<i>Tráfego Unidirecional</i> .....	42

4.3.4	<i>Tráfego Bidirecional</i> .....	43
4.4	<i>Soluções para Problemas de Assimetria</i> .....	43
4.4.1	<i>Ack Filtering</i> .....	44
4.4.2	<i>Acks -First Scheduling</i> .....	44
4.4.3	<i>Ack Filtering com Acks-First Scheduling</i> .....	45
4.4.4	<i>ACK Reconstruction</i> .....	46
4.4.5	<i>Técnica Proposta</i> .....	46
<b>Capítulo 5 - Análise de Modelos</b> .....		<b>48</b>
5.1	<i>Modelo Experimental</i> .....	48
5.1.1	<i>Topologia</i> .....	48
5.1.2	<i>Desempenho</i> .....	48
5.2	<i>Network Simulator</i> .....	49
5.2.1	<i>Topologia</i> .....	50
5.2.2	<i>Configuração NS2</i> .....	51
<b>Capítulo 6 - Implementação</b> .....		<b>52</b>
6.1	<i>FIFO</i> .....	52
6.1.1	<i>Tráfego Unidirecional</i> .....	52
6.1.1.1	<i>Desempenho no Sentido Reverso</i> .....	52
6.1.1.2	<i>Desempenho no Sentido direto</i> .....	53
6.1.2	<i>Tráfego Bidirecional</i> .....	53
6.1.2.1	<i>Análise da Fila</i> .....	53
6.1.2.2	<i>Desempenho no Sentido Reverso</i> .....	54
6.1.2.3	<i>Desempenho no Sentido Direto</i> .....	55
6.2	<i>Ack Filtering</i> .....	55
6.2.1	<i>Tráfego Unidirecional</i> .....	56
6.2.1.1	<i>Desempenho no Sentido Reverso</i> .....	56
6.2.1.2	<i>Desempenho no Sentido direto</i> .....	56
6.2.2	<i>Tráfego Bidirecional</i> .....	57
6.2.2.1	<i>Análise da Fila</i> .....	57
6.2.2.2	<i>Desempenho no Sentido Reverso</i> .....	58
6.2.2.3	<i>Desempenho no Sentido Direto</i> .....	58
6.3	<i>Acks -First Scheduling</i> .....	59
6.3.1	<i>Tráfego Unidirecional</i> .....	59

6.3.1.1	<i>Desempenho no Sentido Reverso</i> .....	59
6.3.1.2	<i>Sentido direto</i> .....	59
6.3.2	<i>Tráfego Bidirecional</i> .....	60
6.3.2.1	<i>Análise da Fila</i> .....	60
6.3.2.2	<i>Desempenho no Sentido Reverso</i> .....	61
6.3.2.3	<i>Desempenho no Sentido Direto</i> .....	61
6.4	<i>Solução Proposta</i> .....	62
6.4.1	<i>Tráfego Unidirecional</i> .....	62
6.4.1.1	<i>Desempenho no Sentido Reverso</i> .....	62
6.4.1.2	<i>Desempenho no Sentido direto</i> .....	62
6.4.2	<i>Tráfego Bidirecional</i> .....	63
6.4.2.1	<i>Análise da Fila</i> .....	63
6.4.2.2	<i>Desempenho no Sentido Reverso</i> .....	64
6.4.2.3	<i>Desempenho no Sentido Direto</i> .....	65
6.4.3	<i>Análise para pacotes de 1 Kbytes</i> .....	65
6.4.3.1	<i>Análise da Fila</i> .....	66
6.4.3.2	<i>Desempenho no Sentido Reverso</i> .....	66
6.4.3.3	<i>Desempenho no Sentido Direto</i> .....	67
6.4.4	<i>Análise para pacotes de 500 bytes</i> .....	67
6.4.4.1	<i>Análise da Fila</i> .....	67
6.4.4.2	<i>Desempenho no Sentido Reverso</i> .....	68
6.4.4.3	<i>Desempenho no Sentido Direto</i> .....	69
6.5	<i>Comparativo de Desempenho</i> .....	69
<b>Capítulo 7 - Conclusão</b> .....		<b>71</b>
7.1	<i>Conclusão</i> .....	71
7.2	<i>Sugestão para futuros trabalhos</i> .....	72
<b>Referências Bibliográficas</b> .....		<b>73</b>
<i>ANEXO 1</i> .....		82

# Agradecimentos

Agradeço ao professor e orientador Dr. Carlos Becker Westphall pela oportunidade, crédito e incentivos constantes, aos colegas do LRG pelas trocas de informações e a todos os demais que ajudaram a concretizar este trabalho. Agradeço também a esta notável academia chamada UFSC, e ao INE onde realizei sonhos.

# Dedicatória

*Para minha esposa Liane,  
e meus pais  
Raul (in memoriam) e Ana de Lourdes,*



# Glossario

AAL - *ATM Adaption Layer*

ABR - *Available Bit Rate*

ADSL - *Asymmetric Digital Subscriber Line*

ANSI - *American National Standards Institute*

ATM - *Asynchronous Transfer Mode*

CBR - *Constant Bit Rate*

CCITT - *Consultative Committee for International Telegraph and Telephone*

FTP - *File Transfer Protocol*

HDLC - *High-level Data Link Control*

HDSL - *High-bit-rate Digital Subscriber Line*

ICMP - *Internet Control Message Protocol*

IEEE - *Institute of Electrical & Electronic Engineers*

IETF - *Internet Engineering Task Force*

IP - *Internet Protocol*

ISDN - *Integrated Services Digital Network*

MAU - *Media Adaptor Unit*

MAC - *Media Access Control*

OSI - *Open System Interconnect*

POP - *Post Office Protocol*

PPP - *Point to Point Protocol*

RFC - *Request For Comments*

SDH - *Synchronous Digital Hierarchy*

*SDLC - Serial Data Link Control*

*SMTP - Simple Mail Transfer Protocol*

*SNMP - Simple Network Management Protocol*

*SONET - Synchronous Optical Network*

*TCP/IP - Transmission Control Protocol/Internet Protocol*

*WAN - Wide Area Network*

# Resumo

O objetivo deste trabalho é apresentar uma proposta de melhoria a problemas de desempenho do TCP, que ocorrem devido aos efeitos da assimetria. Estes problemas acontecem em redes de acesso assimétricas, causando degradação no desempenho final do TCP, os quais ocorrem devido as imperfeições e variações no retorno dos pacotes de confirmação, *Acknowledgement*, enviados pela estação cliente em resposta aos pacotes de dados recebidos da estação servidora. Este trabalho detalha algumas soluções a este problema, os quais usam técnicas locais de camada de rede.

Como resultado de nossa pesquisa apresentamos uma proposta baseada no aperfeiçoamento e combinação de algumas destas técnicas, e cujos resultados obtidos analisamos através de simulação de modelos.

**Palavras-chave** *Acknowledgement* , ADSL, Assimetria, TCP.

# Abstract

The intention of this paper is to propose a solution to the TCP performance problems caused by the asymmetry. These problems happen in asymmetrical network access, causing TCP final performance degradation, which occurs because of the imperfections and variations of the *Ack* return as reply of the receiver to the transmitter. This work details diverse solutions to this problem, that use combinations of local link-layer. It considers a solution based on the combination of some techniques, analyzing the results gotten through modeling in simulator.

Key words: *Ack* , ADSL, Asymmetry, TCP.

## Lista de Figuras

5.1.1	<i>Topologia</i> .....	48
5.1.2	<i>Desempenho</i> .....	48
5.2.1	<i>Topologia</i> .....	50
6.1.1.1	<i>Desempenho no Sentido Reverso</i> .....	52
6.1.1.2	<i>Desempenho no Sentido direto</i> .....	53
6.1.2.1	<i>Análise da Fila</i> .....	53
6.1.2.2	<i>Desempenho no Sentido Reverso</i> .....	54
6.1.2.3	<i>Desempenho no Sentido Direto</i> .....	55
6.2.1.1	<i>Desempenho no Sentido Reverso</i> .....	56
6.2.1.2	<i>Desempenho no Sentido direto</i> .....	56
6.2.2.1	<i>Análise da Fila</i> .....	57
6.2.2.2	<i>Desempenho no Sentido Reverso</i> .....	58
6.2.2.3	<i>Desempenho no Sentido Direto</i> .....	58
6.3.1.1	<i>Desempenho no Sentido Reverso</i> .....	59
6.3.2.1	<i>Análise da Fila</i> .....	60
6.3.2.2	<i>Desempenho no Sentido Reverso</i> .....	61
6.3.2.3	<i>Desempenho no Sentido Direto</i> .....	61
6.4.1.1	<i>Desempenho no Sentido Reverso</i> .....	62
6.4.1.2	<i>Desempenho no Sentido direto</i> .....	62
6.4.2.1	<i>Análise da Fila</i> .....	63
6.4.2.2	<i>Desempenho no Sentido Reverso</i> .....	64
6.4.2.3	<i>Desempenho no Sentido Direto</i> .....	65
6.4.3.1	<i>Análise da Fila</i> .....	66
6.4.3.2	<i>Desempenho no Sentido Reverso</i> .....	66
6.4.3.3	<i>Desempenho no Sentido Direto</i> .....	67
6.4.4.1	<i>Análise da Fila</i> .....	67
6.4.4.2	<i>Desempenho no Sentido Reverso</i> .....	68
6.4.4.3	<i>Desempenho no Sentido Direto</i> .....	69
6.5	<i>Comparativo de Desempenho</i> .....	69

# Capítulo 1 – Introdução

A necessidade crescente dos usuários por conexão em alta velocidade levou ao desenvolvimento de novas tecnologias de acesso à rede. Algumas destas, tais como modem a cabo e ADSL<sup>1</sup> [ADSL] [ITUT99], com o objetivo de solucionar o problema de acesso à última milha.

Estas tecnologias de rede possuem como característica a assimetria no tráfego de dados, onde as características de largura de faixa da rede em um sentido podem ser completamente diferentes das apresentadas no sentido oposto [BP01].

Quando a largura de faixa é assimétrica tal que o trajeto reverso usado pelos *Acks* em confirmação aos pacotes de dados do sentido direto da conexão encontra-se congestionado, o retorno lento ou a perda de confirmação, ocasiona a degradação do desempenho no sentido direto. Esta queda de desempenho deve-se a necessidade do TCP da estação servidora esperar pela confirmação de recebimento dos pacotes enviados, para a transmissão de novos pacotes .

Considera-se que uma rede é assimétrica com respeito ao desempenho do TCP<sup>2</sup> [GH99], se a vazão de pacotes de dados conseguida não for unicamente em função das características da conexão no sentido direto, mas depende também das características da conexão no sentido reverso [BP01] [BPK99]. Mesmo quando a largura de faixa é simétrica, em determinadas situações devido a condições de tráfego, o retorno da confirmação (*Acks*) pode ser variável.

A classificação de assimetria pode fazer parte de diversos modelos, sendo que neste caso, a característica assimétrica da conexão é composta pela assimetria de largura de faixa acentuada pela assimetria de tráfego.

Como objeto deste estudo abordaremos tanto os casos de tráfego unidirecional como bidirecional, onde as considerações são adotadas como elementos de comparação para casos de alto grau de assimetria de tráfego. Esta situação ocorre em determinadas situações de tráfego sobre ADSL, onde se manifestam explicitamente as limitações do TCP para este tipo de situação.

---

<sup>1</sup>ADSL - *Assimetric Digital Subscriber Line*

<sup>2</sup> TCP – *Transmission Control Protocol*

## Trabalhos Correlatos

Diversos estudos tem sido desenvolvidos relativos aos problemas do TCP quando utilizado em enlaces com alto grau de assimetria.

Lakshman, [LMS97], analisou os problemas relativos a assimetria de largura de faixa, e mostrou que a performance degrada quando relação de largura de faixa  $k > 1$ , onde  $k$  é a relação de largura de faixa dividida pela relação de tamanho do pacote usado nos dois sentidos da conexão.

Phanse [PSK00] analisa o desempenho do TCP sobre PPP sobre ATM e sobre links assimétricos, para casos de tráfego unidirecional e bidirecional, quantificando a parcela de perdas devido as características de cada protocolo .

De acordo com Kalampoukas em [KVR98], os problemas de desempenho se acentuam quando o tráfego bidirecional esta presente em redes com assimetria de largura de faixa. Propõe um algoritmo denominado *Acks First Scheduling*, que implementa um esquema de gerência de fila que aplica prioridade mais elevada aos pacotes de confirmação (*Acks*) sobre os pacotes dos dados. Com isto pretende diminuir o tempo de inatividade da conexão direta diminuindo o tempo em que os *Acks* gastam enfileirados atrás de pacotes de dados na conexão reversa..

Chadi Barakat em [BA00] desenvolve um algoritmo denominado *Ack Filtering*, cuja idéia principal é reduzir o número de *Acks* emitidos na conexão reversa, o que devido a característica cumulativa do *Ack* proporciona um acréscimo de desempenho na conexão direta para casos de alto grau de assimetria.

Segundo Balakrishnan [BPF02], o desempenho do TCP pode ser substancialmente incrementada utilizando técnicas que implementem a supressão de *Acks* na conexão reversa. Para isto, propõe a utilização da técnica *Ack Filtering* em conjunto com a posterior reconstrução dos *Acks* após o trajeto do link reverso, cuja técnica denomina de *Ackreconstruction*.

## 1.2 Organização do Trabalho

Este trabalho está organizado em 7 (sete) capítulos, como detalhado a seguir:

- Capítulo 1: Contém a introdução e os trabalhos correlatos.
- Capítulo 2: Apresenta o ADSL, onde são introduzidos os conceitos básicos dos sistemas de acesso de banda larga existentes e a arquitetura do sistema ADSL sobre a qual é desenvolvida as investigações deste trabalho.
- Capítulo 3: Devido a tecnologia de acesso ADSL se utilizar de vários protocolos, neste capítulo são apresentados os princípios básicos da pilha de protocolos de utilizados em tecnologia de Banda Larga.
- Capítulo 4: As características de Redes de Acesso Assimétricas bem como o problema da assimetria são introduzidos neste capítulo.
- Capítulo 5: Introduz uma abordagem sobre Simulação de Modelos, onde é apresentado o simulador de rede utilizado neste trabalho para realizar as experiências de TCP.
- Capítulo 6: Apresenta os resultados do modelo proposto e de trabalhos correlatos que serviram como suporte a esta investigação.
- Capítulo 7: Finalmente a Conclusão e sugestões para trabalhos futuros são apresentadas neste capítulo.



## Capítulo 2 - O Acesso de Banda Larga

O ADSL [BP01] [BPK99] [Jac88], é apenas um entre os membros de uma família de vários sistemas de acesso de banda larga, DSL<sup>1</sup>, construídos sobre uma infraestrutura de pares de cobre existente. Estes diferentes tipos de tecnologia são popularmente conhecidos como *xDSL*, indicando que pertencem a um padrão, todos habilitados a prover alta velocidade de comunicação sobre linhas telefônicas de pares de cobre .

### 2.1 Sistemas de Acesso de Banda Larga

#### 2.1.1 HDSL

O HDSL<sup>2</sup> [FG99], é a introdução da primeira versão de um sistema DSL. Provê o DSL simétrico sobre dois pares de cobre de até 3,5 Km de distância. É importante pelo seu caráter pioneiro em serviços de alta velocidade, possuindo banda de transmissão de 800 kbps sobre cada par telefônico do assinante. Porém ele não é adequado para oferecer serviços de banda larga a usuários residenciais, visto que necessita de dois pares de cobres para prover banda de 1.5 Mbps e não pode compartilhar o serviço de dados e voz em um mesmo pares de cobre.

#### 2.1.2 SDSL

O SDSL<sup>3</sup> [BAFW03], é um serviço simétrico, que provê a mesma taxa de transmissão em ambos os sentidos da conexão, padrão distinto da família xDSL, que opera sobre um único pares de cobre e suporta serviços de dados e de voz em uma mesma linha. A taxa de dados normalmente varia de a uma faixa de 160 kbps a 2.048 Mbps.

O maior problema com o SDSL é com o limite de *diafonia*<sup>4</sup>, que limita sua operação até no máximo 3 Km para altas taxas de transmissão. O SDSL pode ser utilizado em serviços onde o usuário está próximo da central, e a alta capacidade de largura de banda do canal no sentido do assinante para a central é tão importante como

---

<sup>1</sup> DSL - *Digital Subscriber Line*

<sup>2</sup> HDSL - *High-speed digital subscriber line*

<sup>3</sup> SDSL - *Single-line DSL*

<sup>4</sup> Diafonia - misturas de sinais entre canais diferentes

a do canal no sentido da central para o assinante. O SDSL é projetado para dar suporte a aplicações de servidores de conteúdo, computação interativa e acesso à *Internet*.

### 2.1.3 ADSL

O ADSL, é a principal tecnologia da família *xDSL*, usada em comunicação de banda larga. O termo assimétrico significa que é reservado maior largura de faixa no sentido direto do que para o sentido reverso do canal.

Utiliza a estrutura dos cabos telefônicos atuais com investimentos mínimos se comparados com outros sistemas como cabo, fibra ótica ou o satélite. Permite a simultaneidade e compatibilidade com serviços telefônicos. Isto significa que com a mesma linha telefônica é possível fazer, receber e manter uma chamada telefônica simultaneamente à transferência de dados, como o acesso à *Internet*, sem que nenhum dos serviços seja afetado pelo outro. O ADSL divide a linha telefônica em duas partes, sendo uma para o tráfego de voz e outra para a transmissão de dados. Esta separação permite aplicar aos serviços baseados no ADSL políticas de preços e custos totalmente independentes do serviço telefônico. A tecnologia ADSL converte as linhas de cobre de telefones existentes em caminhos de acesso para aplicações de multimídia e conexões de dados de alta velocidade.

Na tecnologia ADSL, como consequência da assimetria das taxas de transferência de dados, o par metálico pelo qual trafega tem uma maior imunidade a interferência na transmissão devido a diafonia que na transmissão simétrica.

O sucesso comercial da implementação de ADSL implica em requisitos mínimos na capacidade de transmissão. A maioria dos consumidores encontram-se a menos de 3,5 Km da central, sendo que apenas uma pequena percentagem pode necessitar transmissões em distâncias maiores. Normalmente, taxas de transmissão de até 1,5 Mbps são tratadas para estas distâncias.

Avanços na digitalização e compressão na informação tornam possível vídeo de qualidade em taxas abaixo de 1,5 Mbps possibilitando a assinante com linhas digitais operando nesta faixa a receberem serviços multimídia. Taxas de transmissão de 6 Mbps sobre curtas distâncias permitem que diversos canais possam ser acessados simultaneamente em um simples assinante local.

#### 2.1.4 RADSL

O RADSL<sup>5</sup> pode ser entendido como uma versão mais avançada de ADSL, o qual possibilita de forma automática de acordo com as condições momentâneas do pares de cobre otimizar a taxa de transmissão para a melhor desempenho. Isto é importante porque a qualidade de transmissão sobre o pares de cobre depende de varias condições, como temperatura, umidade, interferências elétricas e uma variedade de outros fatores.

Os modems RADSL compensam automaticamente estas condições, utilizando a banda total quando se apresentam ótimas condições, mas diminuem a banda quando a qualidade da linha é degradada.

#### 2.1.5 VDSL

O VDSL<sup>6</sup> [RA04] é uma tecnologia recente que promete fornecer altas taxas de dados para o assinante sobre um pares de cobre a curtas distâncias. Taxas no sentido reverso de até 2.0 Mbps e no sentido direto até 52 Mbps podem ser suportadas a uma distância de 300 metros, enquanto que para 15 Mbps pode ser utilizada até 1 Km. As altas taxas suportadas pelo VDSL para curtas distâncias poderão ser interessantes para aplicações que necessitem de altas taxa de transmissão como por exemplo a distribuição de programas de TV digital para condomínios.

### 2.2 O Serviço ADSL

O ADSL permite a transmissão de uma grande quantidade de dados digitais sobre uma linha telefônica até residências particulares. Esta tecnologia permite que serviços de multimídia como vídeo sob demanda e compras em casa sejam acessíveis em larga escala a um custo relativamente acessível.

Até poucos anos atrás, as previsões indicavam que a maioria das residências num futuro próximo seriam conectadas as linhas de transmissão de alta velocidade, tornando serviços interativos de multimídia disponíveis para o público em geral.

Linhas de fibra óptica eram o meio com o qual se imaginava que estes serviços seriam fornecidas em cada residência.

---

<sup>5</sup> RADSL - *Rate adaptive DSL*

<sup>6</sup> VDSL - *Very-high-rate DSL*

Como a fibra óptica, devido aos custos, se tornou o meio adequado para transmissão entre centrais locais, restou para os quilômetros finais entre centrais e residências as antigas linhas de cobre de par trançado, as quais possuem largura de banda limitada. O custo da substituição destas linhas em larga escala é alto, tornando inviável comercialmente. Por esta razão prevê-se que a transição para fibra deverá acontecer de forma gradual e de acordo com a demanda para estes serviços.

Utilizando as vantagens da transmissão digital e das técnicas de compressão é possível utilizar a infra-estrutura baseada em fios de cobre para transmitir informação digitalizada à altas taxas. A idéia básica é que os serviços previstos requerem alta taxa de transmissão somente na direção da central para o assinante, sendo que a transmissão no sentido do assinante para central é requerida somente para sinais de controle pelo usuário, o qual pode ser transmitido em taxas muito baixas. Esta consideração, tem colocado a transmissão em linhas de cobre possível, propiciando o surgimento da tecnologia ADSL.

### 2.3 Perspectivas para utilização do ADSL

O ADSL permite que as companhias de telefone usem infraestrutura de linhas telefônicas de fios de cobre existente disponibilizando alta velocidade para acesso remoto à *Internet*, redes corporativas e serviços em tempo real em cima de linhas de telefone comuns. Essencialmente, o ADSL provê os meios para entregar a próxima geração de serviços de banda larga em cima de redes de telecomunicações existentes habilitando atualizações e vantagens de mercado.

#### 2.3.1 Perspectiva para o provedor de serviço

O ADSL possibilita novas aplicações em tempo real, multimídia interativa com qualidade na transmissão de vídeo. Tais aplicações incluem computação interativa, vídeo conferência, ensino a distância que requerem grande quantidade de banda disponível.

O ADSL permite aos provedores de serviço a oferecer uma taxa contínua garantida ou alternativamente uma taxa de serviço semelhante e adaptável a modems analógicos. Com ADSL, os usuários podem obter altas velocidades quando comparadas a outros métodos existentes.

O uso residencial e comercial já ocorre para o tráfego ADSL sem a necessidade do uso de linhas sobressalentes utilizando cabos de telefone já instalados. O ADSL proporciona para os provedores de serviço a capacidade de usar uma linha para trafegar dados, mantendo o serviço de telefonia, alavancando assim a infra-estrutura existente.

O ADSL proporciona para as companhias de telefone a capacidade de oferecer um canal privado e segurança de comunicações entre o consumidor e o provedor de serviço sendo que entre outras as vantagens, oferece uma linha dedicada ao cliente, onde as velocidades de transmissão não são afetadas por outros usuários que estão conectados.

Em função da aceitação do mercado, grande número de fornecedores de equipamentos estão oferecendo produtos de última geração, que permitem melhor desempenho com baixo custo.

### 2.3.2 Perspectiva para o usuário final

Do ponto de vista do usuário final, uma das grandes vantagens do ADSL está no fato que este permite usar o telefone ao mesmo tempo para conversações normais enquanto efetua a transmissão de dados. Isso proporciona uma enorme facilidade tanto em casa como no trabalho.

O ADSL provê um acesso remoto de alta velocidade à *Internet*, redes corporativas, e serviços *on-line* em cima de linhas de telefone comuns.

Há uma ampla variedade de velocidades de serviço e opções de centenas de provedores a nível mundial. O ADSL habilita o uso em tempo real de multimídia interativa e transmissão de vídeo com qualidade superior ao utilizado hoje para novos serviços como transmissão de canais de TV pela *Internet*, vídeo-conferência, e até ensino a distância através de texto, áudio e vídeo.

Tanto estabelecimentos residenciais como comerciais, já estão sofrendo com a escassez de linhas livres em cabos de telefone instalados, duplicando deste modo sua capacidade em um benefício real.

O ADSL funciona permanentemente em conjunto com o telefone, o que significa que não há nenhum tempo desperdiçado discando para o provedor, tentando acessar o serviço várias vezes ao dia esperando para ser conectado, uma vez que o ADSL está sempre pronto para uso.

## 2.4 Oportunidades de Negócios

A tecnologia ADSL prove uma nova plataforma para conectividade em alta velocidade e oferta de serviços de banda larga para pequenos negócios. É esta alta velocidade de conectividade para residências, empresas de pequeno porte e escritórios remotos que criarão muitas novas oportunidades de negócio para fornecedores de serviço.

### 2.4.1 Acesso a Internet em alta Velocidade

Com o crescimento explosivo da *Internet* em anos recentes, este é um dos principais mercados para o ADSL. O ADSL pode entregar não somente o acesso de alta velocidade da *Internet*, mas também um serviço dedicado que permita que o usuário navegue na *Internet* e utilize o telefone. Além disso, o fornecedor de serviço pode também oferecer conteúdo, serviços de hospedagem de páginas pessoais e ferramentas de suporte.

### 2.4.2 Serviços de Interconexão

O ADSL se presta para a interconexão de filiais eficazmente substituindo linhas dedicadas na formação de extensões de suas redes locais. A maioria de aplicações de negócio por computador executam uma comunicação assimétrica, tornando o ADSL uma tecnologia apropriada para conectar escritórios remotos à empresa.

### 2.4.3 Negócios entre empresas

Empresas com negócios em comum podem querer compartilhar de uma VPN<sup>7</sup> [FG99] que seja confiável e segura. O provedor de rede pode criar uma rede privada virtual usando o ADSL conjuntamente com as redes dos *Backbones*<sup>8</sup> existentes interconectando negócios. Alguns negócios adicionais fora da área de serviço do operador de serviço podem ser conectados através do *Internet* usando determinados protocolos seguros de tunelamento.

---

<sup>7</sup> VPN – *Virtual Private Network*

<sup>8</sup> Backbone – Infraestrutura de rede principal da Internet

#### 2.4.4 Serviços de entrega de Conteúdo

Embora uma conexão de rede de elevada largura de faixa por si só pareça atrativa, muito mais pode ser feito, realçando a qualidade e a quantidade de conteúdo entregue sobre ela, tais como escolha de produtos, música, vídeo e jogos interativos. Tal combinação de rede de alta velocidade e de conteúdo oferecido apresentará um forte atrativo para negócios e clientes residenciais

### 2.5 Arquitetura do Sistema ADSL

De uma característica bastante heterogênea, a rede de suporte ao serviço ADSL se divide em algumas entidades específicas que visam principalmente estabelecer os diversos caminhos e plataformas de serviço para as diversas etapas da conexão do usuário final, desde o seu equipamento inicial, até o ambiente *Internet* de destino final. Na residência do assinante, o modem ADSL é ligado a um computador pessoal ou rede de computadores, bem como a um telefone. O par trançado da linha telefônica é conectado a um divisor à central telefônica local, onde outro divisor separa a voz dos dados. A chamada de voz é enviada à rede telefônica pública e os dados transmitidos até um concentrador, e deste encaminhados a um roteador de alta velocidade de um *backbone Internet* para acessar os serviços requisitados.

A rede de suporte aos serviços ADSL pode ser subdividida em Rede de Acesso e Rede de Transporte.

#### 2.5.1 Rede de Acesso

Trata-se da porção da rede de serviço ADSL que é constituída de modems dos assinantes, rede de pares metálicos e concentradores de acesso *DSLAMS*.

##### 2.5.1.1 Modem do Assinante

Por suas funções, descreve-se o modem como elemento de conexão, entre o computador do cliente e a rede de pares metálicos, normalmente já existente na operadora de telefonia. O modem provê a adaptação do sinal digital, oriundo da interface de rede do computador às características de transmissão exigidas pelo par de fios da rede telefônica. Por técnicas especiais de modulação e transmissão, consegue-se

transmitir um sinal de alta velocidade em meio de transmissão com banda passante muito limitada.

### 2.5.1.2 Rede Metálica

Quanto a rede de pares metálicos, trata-se da rede externa já existente nas plantas das operadoras públicas de telefonia fixa comutada. Com a necessidade de adaptação do sistema de telefonia à transmissão de sinais de dados, foram desenvolvidos os modems, equipamentos que modulam o sinal de dados adaptando-os para transmissão em canais de voz.

A restrição de banda imposta pelo sistema de telefonia impede o tráfego de dados modulados em sinal analógico a altas velocidades, restringindo a velocidade de acesso máxima a 56 Kbps.

As limitações da banda de frequência não são devidas à linha do assinante, mas à limitação imposta pela rede telefônica que limita a banda de transmissão em 3,3 KHz.

Linhas de cobre puras podem transmitir sinais na região de MHz, mas com grande atenuação, que é incrementada com o aumento da frequência e com o aumento do comprimento da linha. Além disso o sinal pode ser degradado em função da baixa relação sinal ruído. Isto define as dificuldades de transmitir dados em alta velocidade sobre um par de fios de cobre. O diâmetro do condutor é outro fator decisivo na atenuação de sinal. Quanto menor o diâmetro, maior a atenuação. O padrão utilizado para os fios de cobre é 24 AWG<sup>9</sup>.

Com o advento das redes digitais de telecomunicações foi possível criar tecnologias sobre pares de fios de cobre que permitam a transmissão de dados em alta velocidade, uma vez que a barreira dos sistemas analógicos de tratamento de voz foi quebrada. Assim podemos transmitir dados sobre pares de fios de cobre com uma frequência aproximada de no máximo 1 MHz.

---

<sup>9</sup> AWG - *American Wire Gauge*



### 2.5.1.3 Concentradores de Acesso - DSLAM

Os DSLAMs são o núcleo da rede ADSL, pois recebem a cabeação de todos os assinantes, promovem o tratamento e recepção dos sinais de linha e os concentra em um feixe ATM<sup>10</sup> [FG99], através de uma interface óptica padrão de mercado.

O DSLAM<sup>11</sup> é um concentrador de linhas de acesso, situado geralmente na central da companhia telefônica que recebe sinais de múltiplas conexões ADSL e coloca os sinais sobre uma linha de alta velocidade usando técnicas de multiplexação.

Dependendo do produto, os multiplexadores DSLAM conectam linhas ADSL com combinações de ATM [STTN02][GH99], ou redes TCP/IP<sup>12</sup> [RFC791][RFC793].

## 2.5.2 Rede de Transporte

Trata-se da porção da rede de serviço ADSL que é constituída de Terminadores ATM e Servidores de Acesso Remoto.

### 2.5.2.1 Terminação ATM - BRAS

Como os DSLAM's concentram os assinantes em determinadas coordenadas geográficas dentro da planta da operadora de serviços telefônicos, é necessário transportar os feixes ATM resultantes até os pontos de conexão da rede da operadora com os elementos finais da rede, como os servidores de autenticação e terminadores de sessão.

Usualmente utilizam-se os transportes via rede SDH [RFC 2823] existente e via rede ATM até o ponto de conexão com o BRAS<sup>13</sup>.

Não importando a opção disponível, a confiabilidade da solução ADSL como um todo depende fortemente da qualidade da rede de transporte para a correta operação.

O BRAS consiste de um equipamento que provê a conexão do usuário com a rede IP ao qual se destina. Entre suas funções, a cessão temporária de um endereço IP ao usuário, durante o período da conexão é uma das principais. Quando se usa o PPP<sup>14</sup> [Per89], uma conexão ponto a ponto é estabelecida. A fim de acomodar eficientemente

---

<sup>10</sup> ATM - *Asynchronous Transfer Mode*

<sup>11</sup> DSLAM – Multiplexador de Acesso Digital de Linha de Assinante

<sup>12</sup> TCP / IP - *Transport Control Protocol / Internet Protocol*

<sup>13</sup> BRAS – *Broadband Remote Access Server*

<sup>14</sup> PPP - *Point to Point Protocol*

múltiplas conexões a uma variedade de serviços, um mecanismo de tunelamento baseado em L2TP<sup>15</sup> [STTN02] pode ser usado. O Servidor de Acesso Remoto de Banda Larga estabelece um túnel a cada um dos serviços suportados. Este túnel acomodará as conexões múltiplas do PPP que vêm de todos os usuários conectados.

As principais tarefas do BRAS são as seguintes: gerência de QoS<sup>16</sup>, autenticação, criação de túneis para diferentes serviços, e agregar inúmeros assinantes em túneis quando usando PPP.

## 2.6 Tecnologia ADSL

O ADSL [ADSL] [ANSI98] [ITU99] [Kwo99] é uma técnica de modulação para a transmissão de dados em alta velocidade sobre o par de cobre. Diferentemente da técnica de modulação usadas pelos modems de banda de voz [ITU-T V.32 - V.90] que só transmitem na banda de freqüências usada em telefonia de 300 Hz a 3400 Hz, os modems ADSL operam em uma faixa de freqüências mais ampla que vai desde 24 kHz até 1104 kHz, aproximadamente, e permitem a coexistência de serviço de dados com o serviço telefônico na mesma linha de assinante, o que não é possível com um modem convencional que opera na banda de voz.

Por ser uma modulação em que se transmitem diferentes portadoras, no sentido usuário para a rede e rede para o usuário, o modem ADSL situado no lado do usuário é diferente daquele localizado do outro lado, na central telefônica.

Além dos modems situados na casa do usuário, ATU-R<sup>17</sup> e na central, ATU-C<sup>18</sup> à frente de cada um deles é necessária a instalação de um dispositivo denominado *splitter*. Este dispositivo é um conjunto de filtros passa altas e outro passa baixas cuja finalidade é separar os sinais transmitidos pela linha de assinante. Os sinais de baixa freqüência de telefonia serão separados dos de alta freqüência do ADSL.

---

<sup>15</sup> L2TP – *Layer Two Tunnel Protocol*

<sup>16</sup> QOS – *Quality of Service*

<sup>17</sup> ATU-R - *ADSL Terminal Unit– Remote*

<sup>18</sup> ATU-C - *ADSL Terminal Unit – Central*

### 2.3.1 Modulação DMT

A modulação DMT<sup>19</sup> [TSC02], consiste na utilização de múltiplas portadoras, sendo que cada uma destas portadoras denominadas de subportadora é modulada em quadratura 16 QAM<sup>20</sup> [CCITT V.22 bis] por uma parte do fluxo total de dados que vão ser transmitidos.

As mais baixas frequências são reservadas para os serviços analógicos das companhias telefônicas, ocupando os primeiros 4 KHz do espectro de 300 a 3400 khz. A parte principal da banda para o ADSL é a faixa de 20 KHz até cerca de 1.1 MHz, sendo que é reservada de 20 a 138 khz para transmissão digital de dados na direção do assinante até a central e o canal de transmissão do central em direção ao assinante ocupa de 138 a 1.1 Mhz de banda. Estas subportadoras estão separadas entre si de 4,3125 kHz e a largura de banda que ocupa cada subportadora modulada é de 4 kHz.

A distribuição do fluxo de dados entre subportadoras se faz através da estimativa da relação Sinal/Ruído na banda designada a cada uma delas. Quanto maior é esta relação, maior é o banda que se pode transmitir por uma subportadora sendo um máximo de 15 bits por portadora. A estimativa da relação Sinal/Ruído se faz no começo, quando se estabelece o enlace entre o ATU-R e o ATU-C, por meio de uma seqüência de treinamento pré definida. A técnica de modulação usada é a mesma tanto para o ATU-R como para o ATU-C. Tanto a quantidade de bits quanto a potência por portadora é negociada. Cada canal carrega vários níveis de pulsos que podem representar até 15 bits de dados, e é alocada a potência de transmissão para cada canal individualmente dependendo do nível de ruído e atenuação na transmissão em cada banda.

Quando utilizado com cancelamento de eco o ATU-C dispõe de até 256 subportadoras, enquanto que o ATU-R só pode dispor de no máximo 32 ou seja, neste caso, utiliza 256 portadoras para downstream e 32 portadoras para upstream.

Já quando utilizado com FDM<sup>21</sup> o ATU-C dispõe de até 224 subportadoras, enquanto que o ATU-R só pode dispor de no máximo 32 subportadoras.

O algoritmo de modulação se traduz numa transformada rápida de Fourier inversa no modulador e em uma transformada rápida de Fourier no demodulador situado do outro lado da linha de assinante.

---

<sup>19</sup> DMT - *Discrete MultiTone*

<sup>20</sup> QAM – *Quadrature Amplitude Modulation*

<sup>21</sup>FDM

Na modulação DMT com FDM os espectros dos sinais ascendente e descendente não se sobrepõem, o que simplifica o projeto dos modems, mas reduz a capacidade de transmissão no sentido descendente, não só pelo menor número de portadoras disponíveis, como pelo fato de que as de menor frequência, aquelas para as quais a atenuação do par de cobre é menor não estão disponíveis.

No caso da modulação FDM com EC, baseada em um cancelador de ecos para a separação dos sinais correspondentes aos dois sentidos de transmissão, permite maiores taxas a custo de uma maior complexidade de projeto.

### 2.3.2 Modulação QAM

Na modulação QAM [ITU-T V.22 bis] a informação está contida na fase e na amplitude do sinal modulado.

Na modulação 16 QAM os dados de entrada são agrupados em grupos de 4 (*QUADRIBIT*), o que origina 16 combinações possíveis de entrada. O sinal de saída, modulado, varia tanto em amplitude quanto em fase, segundo o sinal de entrada, podendo assumir 16 condições.

## Capítulo 3 – Pilha de Protocolos

As redes ADSL são suportadas por um conjunto de protocolos que quando combinados adequadamente fornecem a infraestrutura necessária para a implementação deste serviço. Por conta deste motivo, se faz necessária o entendimento e compreensão dos princípios básicos bem como as funcionalidades destes protocolos. Deste modo, as próximas seções deste capítulo fornecem um esboço breve do protocolo ATM [FG99], da família do protocolo TCP/IP [RFC791] [RFC793], e do protocolo PPP[Per89], que estão sendo intensamente desenvolvidas sobre tecnologia DSL [BPK01].

### 3.1 O ATM

O ATM é uma tecnologia de comutação de pacotes, projetada para oferecer uma transmissão flexível para uma grande variedade de serviços tais como voz, vídeo e dados. A tecnologia ATM é um esquema orientado a conexão porque permite que os hosts se comuniquem com seus pares estabelecendo circuitos virtuais ou conexões em uma forma similar às redes de comutação de circuitos. É uma tecnologia baseada em comutação de células. O ATM segmenta toda sua informação em pacotes de tamanhos fixos, o que simplifica a comutação e funções de multiplexação.

A tecnologia ATM também é modelada com a mesma arquitetura hierárquica do modelo OSI<sup>1</sup>, porém somente as camadas mais baixas são utilizadas. Assim como no modelo OSI/ISO<sup>2</sup>, a tecnologia ATM também é estruturada em camadas, que substituem algumas ou uma parte das camadas da pilha original de protocolos. Esta estruturação do sistema ATM é dividida em 3 camadas, que são a Camada Física, Camada ATM e a Camada de Adaptação ATM .

#### 3.1.1 Camada Física

Consiste no transporte físico usado para transferência de células de um nó para outro. Esta camada é muito flexível no sentido de que pode trabalhar com várias categorias de transporte físico.

---

<sup>1</sup> OSI – *Open Standard Interconnection*

<sup>2</sup> ISO - *International Standards Organization*

Existem quatro padrões diferentes para a Camada Física. No entanto, os mais importantes baseiam-se nos modelos SONET<sup>3</sup> e SDH<sup>4</sup>. Esses modelos são praticamente equivalentes, e surgiram como tentativa de adaptação do TDM<sup>5</sup> às grandes frequências de transmissão possibilitadas pelas fibras óticas. Basicamente, as diferenças entre os dois estão no que diz respeito a velocidade de operação, meio físico utilizado e estruturação de dados.

### 3.1.2 Camada ATM

Viabiliza o chaveamento e roteamento das células ATM de acordo com os campos VCI e VPI do cabeçalho, o qual contém informações sobre VC e VP, tipo e controle de erro.

A parte de dados contém realmente as informações que devem ser transmitidos pela rede, sendo que seu tamanho foi definido considerando o compromisso entre a eficiência na transmissão de grandes pacotes de dados e a minimização do tempo atraso devido a processamento, extremamente importante para áudio, vídeo e protocolos sensíveis a este atraso. As células são transmitidas serialmente e se propagam em uma sequência numérica estrita através da rede.

Embora a rede ATM comute pequenas células no nível mais baixo, os programas aplicativos que transferem dados através de uma rede ATM não lêem ou escrevem células. Para isso, os computadores interagem com a rede ATM através de uma camada de adaptação AAL<sup>6</sup> que é parte do padrão ATM [ITUT-I 362].

Existem diferentes tipo de Camada de Adaptação para diferentes tipos de tráfego de acordo com as características de transmissão de um tráfego específico

Ao estabelecer uma conexão, um host deve escolher qual protocolo de camada de aplicação vai utilizar.

---

<sup>3</sup> SONET - *Synchronous Optical Network*

<sup>4</sup> SDH - *Synchronous Digital Hierarchy*

<sup>5</sup> TDM - *Time Division Multiplexing*

<sup>6</sup> ALL - *Adaptation ATM Layer*

### 3.1.3 Camada de Adaptação 1 - AAL1

A AAL1 da rede ATM envia e recebe dados a uma taxa fixa de bits. É utilizada para serviços de vídeo, pois para este caso um serviço de taxa fixa de bits é necessário para assegurar que não ocorram atrasos na transmissão.

### 3.1.4 Camada de Adaptação 5 - AAL5

A AAL5 [GH99] é utilizada para enviar pacotes de dados através de uma rede ATM. Apesar da rede ATM utilizar pequenas células de tamanho fixo no nível mais baixo, a AAL5 possui interface que recebe e envia pacotes grandes e de tamanhos variados através de um circuito virtual da rede ATM.

### 3.1.5 Conexões Lógicas

Para atingir um destino remoto através de uma rede ATM, um *host* deve estabelecer uma conexão virtual. Isto significa que uma conexão virtual necessariamente deve ser estabelecida através da rede ATM antes de qualquer transferência de dados. O ATM oferece duas formas de conexão, que são os Circuitos Virtuais Comutados (SVC), definidos dinamicamente e os Circuitos Virtuais Permanentes (PVC), definidos pelo administrador de rede e comutados permanentemente.

Um *host* comunica-se com o ATM para solicitar o estabelecimento de um SVC. O *host* define o endereço completo de um computador remoto e a qualidade de serviço solicitada. O sistema de sinalização do ATM assume o controle e estabelece um caminho a partir do *host* de origem, através da rede ATM até o computador *host* remoto.

O administrador de rede atua sobre os comutadores da rede ATM para configurar os circuitos virtuais manualmente. Especificam também a origem e o destino do circuito, a qualidade de serviço e os identificadores por cada um dos *hosts* para cada circuito.

A tecnologia ATM oferece dois tipos de conexão de transporte que se completam, que são o VP<sup>7</sup> e o VC<sup>8</sup>.

Um VC é um acesso unidirecional feito da concatenação de uma seqüência de elementos de conexão, já o VP consiste de um grupo destes canais. Deste modo, para cada VP existem vários VCs.

Cada VP e VC possuem um identificador associado, VPI e VCI, respectivamente, que representam o endereço local de uma dada conexão, e que mudam conforme a célula trafega na rede. Cada conexão virtual em um único caminho VP, possui distintos VCIs, porém o mesmo VCI pode vir a ser usado em um outro VP.

### 3.2 Point-to-Point Protocol

O protocolo PPP [**Per89**] é usado atualmente para a maioria das conexões *dial-up* na *Internet*. Pode ser usado também em conexões dedicadas xDSL, e isto é possível porque tanto o provedor de Acesso como o usuário final possuem a mesma configuração de rede. No equipamento do usuário final, pode ser utilizado o PPP sobre *Ethernet*<sup>9</sup> [MLC<sup>+</sup>99] ou PPP sobre ATM [TKB02].

Com os protocolos de camada de rede, as máquinas se comunicam necessitando saber que protocolos de camada da rede estavam disponíveis para suportar uma determinada aplicação do usuário. Além disso, em algumas situações, é desejável negociar as opções a serem usadas durante uma sessão entre duas máquinas. Estas operações eram executadas ocasionalmente usando protocolos proprietários. Entretanto, devido à falta de uniformidade de meios comuns, muitas vezes estas negociações não eram devidamente executados. Isto conduziu geralmente ao desperdício de recursos tais como de largura de faixa. Além disso, até o advento do protocolo ponto a ponto, a indústria não teve meios de definir um protocolo padrão de encapsulamento. O desenvolvimento do PPP resolveu estes dois problemas. O PPP fornece métodos negociando uma grande variedade de opções de operações, bem como meios de encapsulamento padrão dos protocolos da camada de rede. Hoje, o PPP é usado para encapsular datagramas multiprotocolos e para transmiti-los sobre um circuito de comunicação serial.

---

<sup>7</sup> VP - *Virtual Path*

<sup>8</sup> VC - *Virtual Channels*

<sup>9</sup> Ethernet – 802.3



A operação do PPP compreende de três fases principais: o protocolo de controle do link, o protocolo de autenticação e o protocolo de controle de camada de rede.

### 3.2.1 O Protocolo de Controle do Link

Na fase inicial, o LCP<sup>10</sup> detecta se o link físico está pronto para ser usado. No evento *up*, que indica que a camada física está pronta para a transmissão, o LCP em um host estabelece a conexão com seu par com a troca de pacotes de configuração. Outras negociações tais como o comprimento máximo do pacote, uso de compressão, protocolo de autenticação a ser usado, ocorrem entre os pares durante a fase do estabelecimento da ligação. LCP é usado também para encerrar uma ligação a qualquer momento. A terminação de uma ligação pode ocorrer devido à perda de portadora, de falha de autenticação, de falha da qualidade do link, da expiração do período inativo ou do encerramento administrativo da ligação. Quando o PPP está encerrando uma ligação, informa os protocolos de camada de rede de modo que possam fazer exame da ação apropriada.

### 3.2.2 O Protocolo de Autenticação

Embora a autenticação não seja imperativa, pode ser desejável requerer um par de autenticações antes de permitir qualquer troca de pacotes do protocolo de camada de rede. Se execução em um host desejar autenticação, a seguir indica à máquina do par durante a fase do estabelecimento de ligação. Entretanto, é importante que a execução para não permitir a determinação da qualidade dos pacotes do link (durante a fase de LCP) o que atrasa a autenticação indefinidamente. Dai, a autenticação pode ser permitida a ocorrer simultaneamente junto com a determinação da qualidade do link. A execução deve também se assegurar de que nenhum pacote da camada de rede esteja fluindo antes que a autenticação esteja completa. Os vários protocolos de autenticação usados geralmente são o protocolo de autenticação de senha PAP<sup>11</sup>, o CHAP<sup>12</sup> e sua variante, o MS-CHAP<sup>13</sup>.

---

<sup>10</sup> LCP - *Link Control Protocol*

<sup>11</sup> PAP - *Password Authentication Protocol*

<sup>12</sup> CHAP - *Challenge-Handshake Authentication Protocol*

<sup>13</sup> MS-CHAP - *Microsoft-CHAP*

### 3.2.3 O Protocolo de Controle de Camada de Rede

Após o PPP ter completado cada fase (estabelecimento do link e autenticação), cada protocolo de camada de rede é configurado individualmente pelos protocolos de controle de rede apropriado (NCP<sup>14</sup>). Cada NCP pode ser aberto ou fechado a qualquer tempo.

## 3.3 TCP/IP

O TCP/IP é o nome que se dá a toda a família de protocolos utilizados pela *Internet*. Esta família de protocolos foi desenvolvida pela DARPA<sup>15</sup> no Departamento de Defesa dos Estados Unidos.

Este conjunto de protocolos foi desenvolvido para permitir aos computadores compartilharem recursos numa rede. Toda a família de protocolos inclui um conjunto de padrões que especificam os detalhes de como comunicar computadores, assim como também convenções para interconectar redes e rotear o tráfego.

Oficialmente esta família de protocolos é chamada, *Internet Protocol TCP/IP*, comumente referenciada só como TCP/IP, devido a seus dois protocolos mais importantes, o TCP e o IP.

### 3.3.1 Internet Protocol - IP

O Protocolo IP [RFC791] é responsável pela comunicação entre máquinas em uma estrutura de rede TCP/IP. Ele provê a capacidade de comunicação entre cada elemento componente da rede para permitir o transporte de uma mensagem de uma origem até o destino.

O protocolo IP provê um serviço sem conexão e não-confiável entre máquinas em uma estrutura de rede. Qualquer tipo de serviço com estas características deve ser fornecido pelos protocolos de níveis superiores. As funções mais importantes realizadas pelo protocolo IP são a atribuição de um esquema de endereçamento independente do endereçamento da rede utilizada abaixo e independente da própria topologia da rede

---

<sup>14</sup> NCP – *Network Control Protocol*

<sup>15</sup> DARPA - *Defense Advanced Research Project Agency*

utilizada, além da capacidade de rotear e tomar decisões de roteamento para o transporte das mensagens entre os elementos que interligam as redes.

Na arquitetura TCP/IP, os elementos responsáveis por interligar duas ou mais redes distintas são chamados de roteadores. As redes interligadas podem ser tanto redes locais, redes geograficamente distribuídas, redes de longa distância com chaveamento de pacotes ou ligações ponto-a-ponto seriais. Um roteador tem como característica principal a existência de mais de uma interface de rede, cada uma com seu próprio endereço específico. Um roteador pode ser um equipamento específico ou um computador de uso geral com mais de uma interface de rede.

### 3.3.2 Protocolo TCP

O TCP é um protocolo de camada de transporte, a qual reúne os protocolos que realizam as funções de transporte de dados fim a fim, ou seja, considerando apenas a origem e o destino da comunicação, sem se preocupar com os elementos intermediários.

O TCP oferece serviços complexos, que incluem controle de erros e serviço com conexão e envio de fluxo de dados. TCP utiliza o conceito de porta. Para o TCP, uma conexão é formada pelo par endereço IP de origem, porta de origem e endereço IP de destino, porta de destino. Segundo *Comer* [Com03], o TCP oferece as seguintes características:

- **Orientado a Stream.** O TCP pode ser utilizado pela aplicação para fazer *streaming*, pois o receptor pode ir apresentado ao usuário os dados recebidos em seqüência e sem erros, mesmo que a transmissão toda ainda não tenha se completado;
- **Conexão de Circuito Virtual.** As entidades origem e destino guardam informações de sincronismo, número de seqüência, tamanho da janela, dados recebidos, e é através dessas informações que o TCP consegue dar as garantias à aplicação;
- **Transmissão Bufferizada.** Uma conexão TCP procura adaptar seu tráfego à carga da rede e capacidade do receptor, assim, numa transferência de grande volume dados, caso a rede esteja congestionada, o fluxo dessa conexão permanecerá pequeno, por outro lado, caso a rede esteja livre, o fluxo será grande, procurando utilizar a melhor forma à capacidade disponível na rede.

- **Conexão *Full-Duplex*.** O TCP permite que ambas entidades enviem e recebam dados dentro da mesma conexão;

### 3.3.3 Pacote TCP

Entidades TCP trocam dados na forma de segmentos, que consiste num cabeçalho de 20 bytes mais uma parte opcional, seguido de zero ou mais bytes de dados.

A seguir é feita uma descrição de cada campo existente no cabeçalho.

- **Porta de Origem e Porta Destino:** Portas TCP de 16 bits, através das quais será feita a conexão entre as entidades de transporte. Contem os números das portas que identificam os programas nas extremidades da conexão.
- **Número de Seqüência:** Todo byte tem um número de seqüência, portanto, pode ter um *Acknowledge* relativo a ele. Permite a remontagem dos dados no destino.
- **Número de Reconhecimento:** Especifica o próximo byte aguardado pelo destino, informando também que até o byte anterior foi recebido corretamente.
- **Comprimento do Cabeçalho:** Especifica o comprimento do cabeçalho do segmento, possuindo 32 bits. Essa informação é necessária, pois o campo *Opções* é variável.
- **Flags:**
  - URG: campo Urgent Pointer é válido
  - ACK : campo de reconhecimento *Acknowledge number* é válido
  - PSH: receptor é solicitado a entregar os dados à aplicação imediatamente.
  - RST: Restabelecimento da conexão
  - SYN: usado para sincronizar os números de seqüência da conexão.
  - FIN: usado para encerrar a conexão, informa que o transmissor não tem mais dados a enviar.
- **Janela:** Efetua controle de fluxo através de uma janela deslizante de tamanho variável. Esse campo indica quantos bytes podem ser enviados a partir do byte confirmado (*Acknowledge number*).
- **Soma de Verificação:** A soma de todos bytes mais *checksum* deve dar zero.
- **Ponteiro Urgente:** Indica um *offset* do número de bytes a partir do número de seqüência atual no qual existem dados urgentes;
- **Opções:** possui diversas funções, como, por exemplo, comunicar tamanho de *buffer* durante o período de inicialização.

### 3.3.4 O campo Options

O desempenho do TCP está intimamente ligado às condições momentâneas da rede, que podem variar em função de vários fatores, e principalmente daqueles relacionados com as condições de tráfego naquele momento.

Diversas opções podem ser utilizadas no cabeçalho do TCP. As relevantes ao desempenho incluem:

- ***Maximum-receive-segment-size***: Informa ao receptor o tamanho máximo de segmento que o transmissor quer receber durante a conexão. Esta opção deve ser usada com a descoberta do máximo MTU do caminho com o objetivo de estabelecer o tamanho de segmento a ser utilizado.
- ***Window-scale option***: Aumento do tamanho máximo da janela de transferência do TCP, principalmente em redes com alto produto “atraso largura de banda”. Sem essa opção, o tamanho máximo normal da janela é de 65536 bytes. Com essa opção, o tamanho da janela pode chegar até 1 Gbytes.
- ***SACK option*** [MMFR96]. Quando habilitada permite ao receptor enviar informações de *Ack* para blocos não contínuos de dados, permitindo ao transmissor enviar somente o bloco que faltou. O padrão do TCP é enviar *Ack* somente para o número de seqüência mais alto dos pacotes recebidos corretamente, podendo provocar ao transmissor um reenvio de blocos que chegaram corretamente ao destino

### 3.3.5 Maximum Transmission Unit - MTU

Numa conexão TCP o tamanho dos blocos de dados utilizados é negociado no início da sessão. O transmissor tenta utilizar o maior tamanho de segmento não fragmentado possível para a transferência de dados, dentro dos limites da rede, do transmissor e do receptor. O TCP divide as mensagens enviadas pelo usuário em blocos iguais ou menores que 64 Kbytes, enviando cada bloco separadamente. Se o segmento passar por uma rede com MTU menor, deve ser fragmentado, recebendo um novo cabeçalho TCP e IP, com acréscimo de 40 bytes, comprometendo o desempenho do sistema. O pacote consiste de cabeçalho e dados, sendo que o tamanho dos dados é

denominado de MSS<sup>16</sup> [RFC879], e define o máximo de dados que podem ser transmitido em um pacote TCP.

### 3.3.6 Round Rip Time

Outro fator a ser levado em consideração no desempenho do TCP é o RTT<sup>17</sup>. Em enlaces de satélite, ou em conexões assimétricas, no caso no sentido de menor largura de faixa, para tráfego de dados, a situação fica mais crítica, onde o atraso do enlace é da ordem de centenas de ms, levando o RTT para valores muito elevados.

---

<sup>16</sup> MSS - *Maximum Segment Size*

<sup>17</sup> RTT- *Round Trip Time*

## Capítulo 4 - O Desempenho do TCP

O desempenho do TCP é diretamente afetado pelo tempo que os pacotes de confirmação enviados pela estação cliente, em resposta aos pacotes de dados enviados pela estação servidora, gastam para percorrer o caminho desde a origem até o destino, o qual dependem desta confirmação para o envio de novos pacotes de dados. O grau de assimetria de largura de banda da conexão, ou até mesmo em caso de largura de banda ser simétrica, porém com tráfego em ambos os sentidos da conexão, ocasionam a degradação do desempenho do TCP.

### 4.1 Disciplina de Filas

O objetivo da disciplina de fila é definir qual pacote deve ser processado após o processamento do pacote anterior. Quando há congestionamento e os recursos necessários para o processamento de novos pacotes encontra-se esgotado, os pacotes em excesso são enfileirados, e caso seja atingido o tamanho máximo da fila, os novos pacotes entrantes serão descartados. Disciplinas de fila permitem o gerenciamento da contenção de recursos, adquirindo importância no tratamento ofertado ao tráfego. As principais características a serem observadas na implementação e escolha de uma disciplina de fila são a latência<sup>1</sup> induzida, o *jitter*<sup>2</sup>, a escolha dos pacotes a serem descartados, o isolamento e a justiça entre diferentes classes de tráfego [BKG01].

Tanto a latência induzida quanto o *jitter* obtidos por uma determinada classe de tráfego estão diretamente relacionados com o tempo de espera dos pacotes nas filas. A partir da disciplina de fila configurada, quanto maior o tamanho das filas, maior serão a latência e o *jitter* médios dos pacotes encaminhados [BKG01]. Determinadas classes de tráfego podem ter seu encaminhamento privilegiado em relação a outras, a partir da configuração da disciplina da fila.

Também pode ser incluído na disciplinas de fila o gerenciamento das funções de descarte de pacotes. Isto implica que quando a fila do roteador se encontra cheia e um novo pacote chegar, o algoritmo de disciplina de fila deve escolher se o novo pacote deve ser descartado ou se algum outro pacote já enfileirado deve ser descartado para que

---

<sup>1</sup> Latência:

<sup>2</sup> Jitter:

este pacote possa ser processado. Normalmente, quanto menor o tamanho da fila, maior será o descarte de pacotes em caso de congestionamento.

As disciplinas de fila podem conter algoritmos que garantam uma alocação mínima de recursos a determinadas classes de tráfego. A alocação de recursos e garantia aplica-se quando o tráfego corresponde a um perfil previamente definido. Este perfil de tráfego corresponde as características que o tráfego deve possuir para que as garantias das disciplinas de filas possam ser atendidas.

Outra característica importante é como a justiça (fairness) é implementada pela disciplina de filas. O termo justiça é definido como a forma como a disciplina de filas divide os recursos entre requisições concorrentes e de tamanho variado, de modo a obter-se um compartilhamento por igual dos recursos disponíveis entre as varias requisições [APS99] . A forma como o compartilhamento por igual é obtido pode variar, sendo dependente do algoritmo utilizado pela disciplina de fila.

#### 4.1.1 FIFO

A disciplina de filas FIFO consiste simplesmente em processar os pacotes segundo sua ordem de chegada, de modo que os pacotes que sejam recebidos primeiro sejam encaminhados primeiro (First-In-First-Out), ou seja o primeiro que entra é o primeiro que sai. Sua maior vantagem é a simplicidade, que permite que sua implementação. Entretanto, a disciplina de filas FIFO não permite nenhum tipo de diferenciação no tratamento oferecido a pacotes, isolamento entre classes de tráfego ou a existência de justiça em seu encaminhamento.

#### 4.1.2 Priority Round Robin

Este tipo de disciplina de fila é baseado no conceito de que certos tipos de tráfego podem ser identificados e conceitualmente movidos para o início da fila de saída, de modo que classes de tráfego de maior prioridade sejam sempre transmitidas antes de outros tipos de tráfego.

O funcionamento do algoritmo do *Priority Round Robin* baseia-se na premissa que os pacotes são classificados de acordos com regras preestabelecidas e colocados em



filas de serviço associadas. Para cada fila é definida uma prioridade, de forma que exista uma ordem de prioridade entre as filas existentes. Por fila, o processamento de pacotes só ocorre caso as filas de maior prioridade estiverem vazias[APS99] .

O *Priority Round Robin* é utilizado quando se deseja minimizar a latência ou o *jitter* para determinadas classes de serviço. Com este tipo de enfileiramento, o tráfego de alta prioridade apresenta latência induzida mínima, o que o torna adequado para oferecer tratamento especial ao tráfego de aplicações sensíveis ao atraso.

Porém, *Priority Round Robin* também possui várias desvantagens, destacando-se que não há isolamento entre as classes de serviço ou justiça na alocação de recursos. Desta forma, se o tráfego de alta prioridade for elevado os recursos disponíveis para o tráfego de menor prioridade serão mínimos, levando a congestionamentos nas filas associadas e conseqüente degradação no tratamento ofertado ao tráfego das mesmas. Outro problema refere-se a escalabilidade deste tipo de enfileiramento, visto que quanto maior o número de prioridades utilizadas, maiores serão os recursos necessários para o processamento de pacotes.

### 4.1.3 Weighted Round Robin

A disciplina de fila *Weighted Round Robin* também conhecida como Class Based Queuing é uma disciplina de fila de grande aceitação na indústria. *Weighted Round Robin* é uma variação do *Priority Round Robin*, tendo sido proposta para permitir a priorização de certas classes de serviço e, ao mesmo tempo, evitar a completa negação de recursos para as classes de menor prioridade. O *Weighted Round Robin* oferece o isolamento entre as classes de serviço com uma garantia de recursos de processamento de pacotes mínima para cada classe de serviço definida [APS99] .

De forma semelhante ao *Priority Round Robin*, o algoritmo de escalonamento de pacotes do *Weighted Round Robin* permite a definição de diversas classes de serviço através da criação de várias filas associadas com cada classe. À medida que cada pacote é recebido, passa a ser classificado e encaminhado para uma das filas existentes. Entretanto, diferentemente do que ocorre com o *Priority Round Robin*, no *Weighted Round Robin* o escalonador realiza iterações processando pacotes de todas as filas. A quantidade de pacotes a serem processados em cada fila por iteração e o tamanho de cada fila são configuráveis, permitindo a distinção das filas de maior e menor

prioridade. Para as filas de menor prioridade, há sempre a garantia de que a cada iteração um determinado número de pacotes será processado, evitando a completa negação de serviço em favor das filas de maior prioridade a qual pode ocorrer no *Priority Round Robin*.

## 4.2 O problema da Assimetria

Redes assimétricas podem causar impacto na performance de protocolos baseados em confirmação, como é o caso do TCP. A razão para isto, é que embora a conexão no sentido direto do fluxo de dados apresente-se descongestionada, pode haver congestionamento na direção oposta, descontinuando o fluxo de pacotes de confirmação. Esta descontinuidade pode levar a uma baixa performance no fluxo de pacotes de dados. No caso do TCP, a informação de retorno é feita na forma de pacotes de confirmação (*Acks*). O TCP transmissor depende da recepção dos *Acks* para o envio de novos pacotes de dados no sentido direto da conexão.

### 4.2.1 Assimetria de Faixa

Quando a largura de faixa é assimétrica tal que o trajeto reverso usado pelos *Acks* em confirmação aos pacotes de dados no sentido direto da conexão, encontra-se congestionado, o retorno lento ou a perda de *Acks*, ocasiona a degradação do desempenho do TCP no sentido direto [ FSS<sup>+</sup>01] [BPK99] [KVR98].

### 4.2.2 Assimetria de Tráfego

Mesmo quando a largura de faixa da conexão é simétrica, em determinadas situações devido as características de tráfego, principalmente quando utilizado para tráfego bidirecional, o tempo de retorno dos *Acks* devido à interação com os pacotes de dados pode ser variável.

### 4.3.3 Tráfego Unidirecional

Se a conexão reversa possuir uma fila de armazenamento de tamanho adequado, de modo que os *Acks* não sejam descartados no trajeto reverso, então o desempenho

segundo *Lakshman* [LMS97], se dá em função de um índice  $k$  chamado de relação de largura de faixa normalizado, onde  $k$  é a relação de largura de faixa em ambos os sentidos dividida pela relação de tamanho de pacote usado nos dois sentidos.

Isto implica que se o receptor reconhecer mais que um *Ack* a cada  $k$  pacotes de dados, a conexão reversa começará a saturar antes que a conexão direta, limitando a vazão no sentido direto, ou seja, quando os *Acks* chegam na conexão de sentido reverso em uma taxa mais rápida do que o link pode suportar, estes começam a ser enfileirados.

#### 4.3.4 Tráfego Bidirecional

Agora consideramos o caso em que as transferências do TCP ocorrem simultaneamente em sentidos opostos sobre uma rede assimétrica. Um possível exemplo é o caso em que o usuário efetua simultaneamente envio e recebimento de pacotes, neste caso com a transferência de arquivos simultâneos nos dois sentidos da conexão. Na presença de tráfego bidirecional os efeitos apresentados são mais pronunciados porque parte da largura de faixa da conexão é usada para transferência reversa de pacotes de dados, o que ocasiona aumento significativo do grau de assimetria da largura de faixa para transferência direta. Neste caso evidencia-se as deficiências do TCP afetado pelas características da assimetria de faixa somadas aos efeitos da assimetria de tráfego.

Suponha que a conexão reversa é iniciada primeiramente e isto ocupa a conexão com os pacotes de dados no sentido reverso, e então a conexão direta é iniciada. Há elevada probabilidade que muitos *Acks* da conexão direta sofram atraso na conexão reversa que se encontra saturada, ou seja, *Acks* da conexão direta podem ser freqüentemente enfileirados atrás de pacotes de dados, o que ocasiona longos tempos de transmissão. Somente quando a conexão reversa perde pacotes é que a conexão direta possui a oportunidade de efetuar progresso no envio de pacotes acima de sua janela de congestionamento. O enfileiramento de *Acks* atrás do tráfego de pacotes de dados, por períodos de tempos apreciáveis, ocasiona rajadas de tráfego de *Acks*, o que resulta em um efeito conhecido como compressão do *Ack* [FSS<sup>+</sup>01], reduzindo a vazão do TCP.

#### 4.4 Soluções para Problemas de Assimetria

Existem técnicas de disciplina de filas atuam na programação dos *Acks* que fluem no sentido reverso da conexão. Algumas atuam na fila atribuindo prioridade em relação

aos pacotes de dados, outras removendo alguns Acks, ou então removendo e posteriormente reconstruindo na extremidade oposta da conexão reversa. Por outro lado, isto possui o efeito de alterar a característica favorável de afastamento temporal do transmissor TCP onde a transmissão de novos pacotes de dados é provocada por *Acks* entrantes.

Estas questões pretendem ser resolvidas por soluções locais de camada de rede ou por técnicas fim a fim. Algumas destas técnicas requerem mudanças no TCP cliente e outras requerem mudanças no TCP transmissor. Estas técnicas podem ser usadas independentemente, porém algumas técnicas são complementares.

#### 4.4.1 Ack Filtering

*Ack Filtering* [BA00] é uma técnica de disciplina de fila de camada de rede, que reduz o número de *Acks* emitidos na conexão reversa. A técnica deve garantir que o servidor não pare de enviar pacotes de dados e fique esperando *Acks*, o que pode acontecer se os *Acks* forem removidos indiscriminadamente. Quando um *Ack* no cliente está a ponto de ser enfileirado na interface da conexão reversa, é verificada a fila a ser transmitida e se existe um *Acks* mais antigo que pertence à mesma conexão do TCP. Se *Acks* forem encontrados, alguns (ou todos) serão removidos da fila, reduzindo o número de *Acks*. Alguns *Acks* possuem outras funções em TCP, e não devem ser suprimidos para assegurar a operação normal. Um conjunto de políticas de filtros de *Acks* pode ser usado. Estas políticas podem ser determinísticas ou aleatórias. Os algoritmos foram sugeridos também para assegurar uma taxa de *Acks* mínimo para garantir incremento da janela de transmissão e para limitar o número de pacotes de dados reconhecidos por um afastamento de *Ack*. Por fluxo o estado necessita ser mantido somente para conexões com ao menos um pacote na fila.

#### 4.4.2 Acks-First Scheduling

*Acks-First Scheduling* [KVR97] é uma técnica experimental utilizada para melhorar o desempenho da transferência bidirecional. Nestes casos os pacotes de dados

e *Acks* competem pelos recursos da conexão reversa [FH99]. Uma simples fila FIFO<sup>3</sup> para pacotes dos dados e *Acks* pode causar impacto no desempenho na transferência direta.

Uma aproximação possível para minimizar este efeito seria programar diferentemente dados e *Acks* na fila. O algoritmo usado neste caso é o *Acks-First Scheduling*, que aplica uma prioridade mais elevada aos *Acks* sobre os pacotes dos dados. A motivação para esta programação é minimizar o momento inativo da conexão direta diminuindo o tempo em que os *Acks* gastam enfileirados atrás de pacotes de dados na conexão reversa. O tempo de transmissão de *Acks* torna-se pequeno bastante tal que o impacto em pacotes subsequentes de dados é mínimo. Este esquema de programação não requer que a conexão reversa do transmissor identifique ou mantenha explicitamente o estado para conexões individuais do TCP. *Acks-First Scheduling* não evita atraso devido a um pacote de dados na transmissão e permite para casos em que  $k \gg 1$  que a transmissão de pacotes se torne nula na conexão reversa, o que ocorre devido a vazão na conexão ser menor que a quantidade de *Acks* esperando na fila para serem transmitidos.

#### 4.4.3 Ack Filtering com Acks-First Scheduling

Quando utilizadas em conjunto em conexões com grau médio de assimetria ( $K > 1$ ) *Ack Filtering* e *Acks-First Scheduling* proporcionam acréscimo significativo de performance tanto para tráfego unidirecional como para tráfego bidirecional nas conexões direta e reversa. Este aumento de performance é substancialmente maior do que o obtido quando utilizada qualquer uma das técnicas isoladamente.

Este acréscimo de performance ocorre pela combinação das características principais de cada técnica, ou seja, *Acks-First Scheduling* permite alto desempenho na transferência na conexão direta por garantir prioridade nos *Acks* da conexão reversa sobre os pacotes de dados, enquanto que *Ack Filtering* otimiza a transferência na conexão reversa por diminuir a quantidade de *Acks* enviados na conexão, aproveitando a característica cumulativa de *Acks* do TCP. Porém, para casos em que o grau de assimetria é muito elevado ( $K \gg 1$ ), devido às características da técnica *Acks-First Scheduling*, a transmissão de pacotes torna-se nula na conexão reversa, devido a vazão

---

<sup>3</sup> FIFO - *First-In First-Out*

na conexão ser menor que a quantidade de *Acks* esperando na fila para serem transmitidos.

#### 4.4.4 ACK Reconstruction

O *Ack Reconstruction* [BPK99] é uma modificação experimental executada no caminho reverso da conexão de baixa velocidade. Quando um afastamento entre *Acks* é percebido, o algoritmo *Ack Reconstruction* introduz *Acks* adicionais preenchendo as lacunas na seqüência de *Acks*. O reconstrutor determina o número de *Acks* adicionais, estimando o número de *Acks* filtrados. Para isto usa informações implícita recebida dos *Acks*, observando o número de seqüência de cada *Ack* recebido.

O principio do método é que o reconstrutor faça a medida da taxa de chegada de *Acks* usando uma estimativa média de peso exponencial móvel. Esta taxa depende da taxa de saída da conexão no sentido reverso e da presença de outro tráfego que compartilhe a conexão. A saída estimada indica o afastamento temporal médio para o *Acks*. Isto pode ser usado para o *Ack Reconstructor* ajustar o afastamento temporal do *Acks* reconstruídos. O esquema pode também ser usado em combinação com uma adaptação do TCP transmissor, como o *Ack Filtering*. A vantagem está em obter menos rajadas do TCP transmissor, e uma taxa melhor do aumento da janela de congestionamento, com uma redução na variação do RTT.

#### 4.4.5 Técnica Proposta

Este trabalho propõe um sistema de disciplina de filas para a conexão reversa, aplicado a situações de alto grau de assimetria, ocasionada pela combinação de assimetria de largura de faixa somada a assimetria de tráfego, analisados para os casos de tráfego unidirecional e bidirecional. Pretende além de minimizar os efeitos de congestionamento de *Acks* na conexão reversa, enfileirados atrás de pacotes de dados, com conseqüente queda de desempenho da conexão direta, garantir um tráfego substancial de pacotes de dados no sentido reverso da conexão.

A técnica aqui apresentada, é um trabalho experimental desenvolvido a partir das características principais da técnica *Acks-First Scheduling*, utilizando também a prioridade dos *Acks* sobre os pacotes de dados, mas propondo uma disciplina

controlada sobre esta prioridade, de modo que a cada conjunto de *Acks* enviados, verifica-se a fila e caso existam pacotes de dados, obriga-se o envio de um pacote de dados, garantindo com este procedimento uma vazão mínima de dados no sentido reverso.

Em conjunto à este procedimento, utilizamos também a técnica *Ack Filtering* e experimentamos uma variação no tamanho dos pacotes de dados enviados no sentido reverso da conexão, sendo estes pacotes de 1500 bytes, 1000 bytes e 500 bytes, com conseqüente influência no desempenho do TCP.

## Capítulo 5 - Análise de Modelos

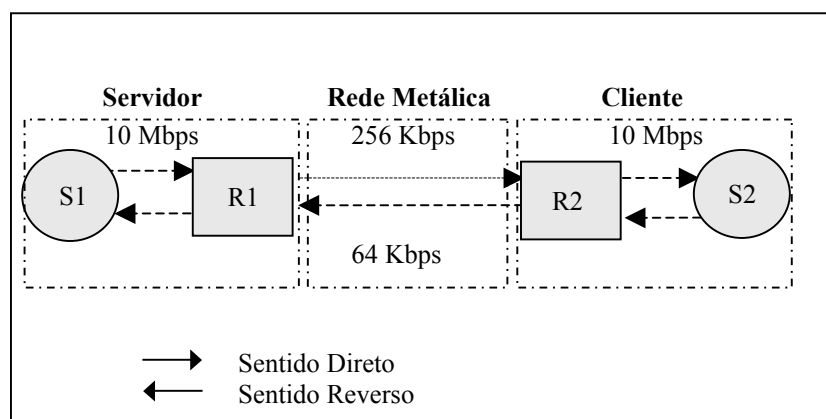
Como elemento de comparação ao Modelo Proposto, analisamos inicialmente os resultados obtidos a partir da coleta de dados de uma rede real em operação com assimetria de faixa e assimetria de tráfego, e posteriormente em experimentos implementados em ambiente gerado através de simulador de rede.

### 5.1 Modelo Experimental

Nesta experiência usamos um trecho limitado de uma rede assimétrica (*ADSL*) em operação para avaliar a performance do TCP, e para isto utilizamos combinação de tráfego unidirecional e bidirecional. Os dados são coletados nos equipamentos servidores das extremidades da rede. Esta prática pretende verificar a performance do TCP em um ambiente real de operação, em situação de alto grau de assimetria, o qual é composto pela assimetria de largura de faixa acentuada pela assimetria de tráfego, ocasionada pelo tráfego bidirecional..

#### 5.1.1 Topologia

A topologia mostrada na Figura 2, representa um acesso ADSL típico, composto pelos elementos de redes usualmente encontrados para este tipo de serviço.



**Figura 2:** Topologia utilizada no modelo experimental

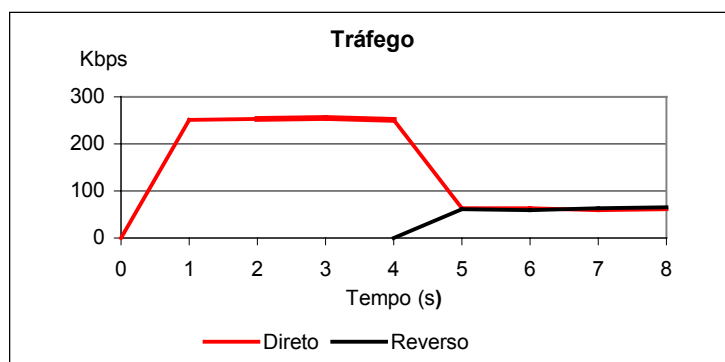
#### 5.1.2 Desempenho

O desempenho obtido, está relacionado com as condições de assimetria da rede, ou seja, depende do grau de assimetria de faixa e também das condições de assimetria de tráfego, como podemos observar no gráfico da Figura 3. Aplicamos tráfego de dados



no sentido direto da conexão a partir de tempo 't = 0s'. Após o tempo 't = 4s' aplicamos tráfego de pacotes de dados no sentido reverso da conexão.

Para o tráfego de pacotes de dados apenas no sentido direto, com intervalo de tempo de 't > 0 s a t < 4 s' obtem-se alto desempenho com a ocupação total da conexão no sentido direto. Já quando iniciamos o tráfego de pacotes de dados no sentido reverso da conexão com tempo 't > 4 s', o desempenho no sentido direto tem uma queda brusca, o que ocorre devido interação entre os pacotes de dados e *Acks* na conexão reversa, o que estaremos examinando em detalhes a partir de análises em modelos equivalentes obtidos a partir de simulações.



**Figura 3:** Tráfego unidirecional e bidirecional

## 5.2 Network Simulator

Neste trabalho estamos utilizando o simulador *ns2 Network Simulator*<sup>1</sup> de Berkeley / LBNL/VINT (versão 2) para realizar as experiências de TCP. Os parâmetros utilizados para a simulação, tais como largura de faixa da conexão, taxa de atraso da rede e tamanho dos pacotes e a própria topologia são derivados de redes típicas assimétricas (ADSL). O *ns2 Network Simulator* é uma ferramenta de simulação amplamente utilizada nos meios acadêmicos, escrita primeiramente em C++<sup>2</sup> com objeto Tcl<sup>3</sup> (OTcl) do MIT<sup>4</sup> como uma linguagem de comandos e de configuração. Tem uma arquitetura orientada a objeto que permite a adição modular de componentes novos, e o aumento de funcionalidade de componentes existentes utilizando conceitos

<sup>1</sup> Simulador de Rede

<sup>2</sup> C++ - Linguagem de Programação Orientada a Objeto

<sup>3</sup> Tcl -

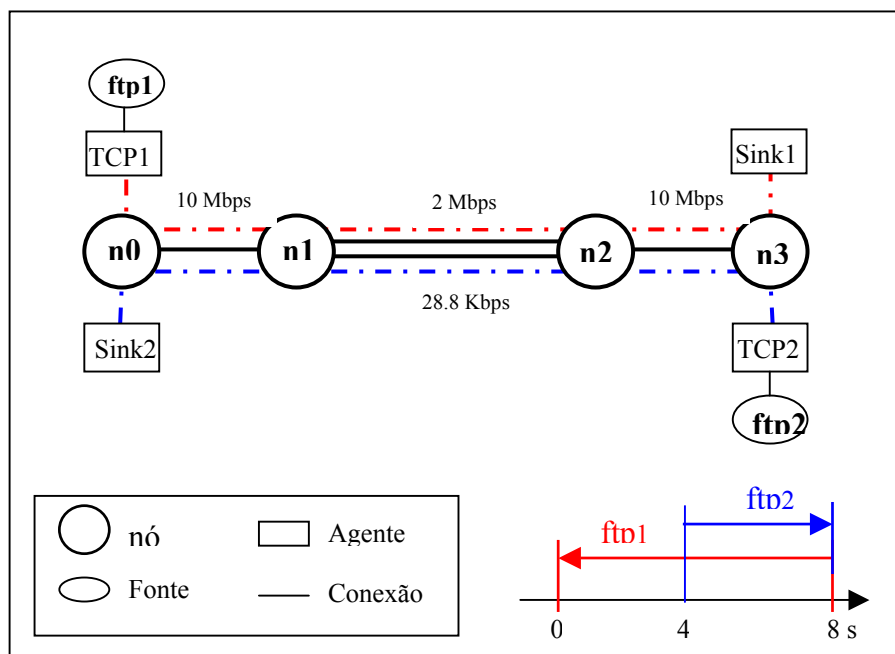
<sup>4</sup> MIT - *Massachusetts Institute of Technology*

de herança. As simulações são configuradas e executadas em OTcl. Isto é possível devido ao acoplamento entre objetos de C++ e de OTcl usando objetos compartilhados.

O simulador ns2 fornece abstração para as conexões, a filas e os nós da rede. Possui execuções de diversos protocolos de rede comuns e aplicações.

### 5.2.1 Topologia

Utiliza-se para este experimento uma topologia básica como mostra a Figura 4, composta de dois nós ADSL,  $n1$  e  $n2$ . Possui conexão TCP a dois servidores  $n0$  e  $n3$ , de acordo com o sentido do tráfego TCP1 ( $n0, n3$ ) e TCP2 ( $n3, n0$ ). Inicialmente utilizamos fluxo de dados FTP1 no sentido direto de alta velocidade, com tempo inicial 'ti = 0s' até tempo final 'tf = 10s', e então iniciamos o FTP2 no sentido reverso, de baixa velocidade com tempo inicial em 'ti = 4s'. Os parâmetros de medida de desempenho utilizados são a taxa de transferência de pacotes de dados no sentido direto, taxa de transferência de pacotes no sentido reverso e taxa de transferência de pacotes de confirmação (Acks) no sentido reverso.



**Figura 4:** Topologia básica utilizada nas simulações

### 5.2.2 Configuração NS2

- Descrição do link:

`$ns simplex-link $n1 $n2 2Mbps 5ms DropTail`

`$ns simplex-link $n2 $n1 28,8 kbps 50ms DropTail`

- Tamanho da fila no roteador: `$ns queue-limit $n0 $n1 20`
- Tamanho da janela no transmissor: 20 pacotes
- Agente: `set src [new Agent/TCP/FullTcp]`
- Tráfego: `$ns at 0 "$ftp1 start"` e `$ns at 4 "$ftp2 start"`

Os parâmetros de configuração bem como a topologia empregada na simulação possuem o propósito de investigar os efeitos da assimetria na rede, neste caso composta pela assimetria de largura de faixa e assimetria de tráfego. Os parâmetros de configuração das conexões nos sentidos direto e reverso pretendem simular valores encontrados em redes assimétricas reais (ADSL) quando em condições de alto grau de assimetria.

## Capítulo 6 - Implementação

Para o caso de tráfego unidirecional, analisamos o desempenho de cada técnica aplicada tanto para o sentido reverso como no sentido direto da conexão. Já para o caso de tráfego bidirecional, além do desempenho nos sentidos reverso e direto da conexão, também é feita uma análise na entrada e saída de pacotes na fila do nó *n1*.

### 6.1 FIFO

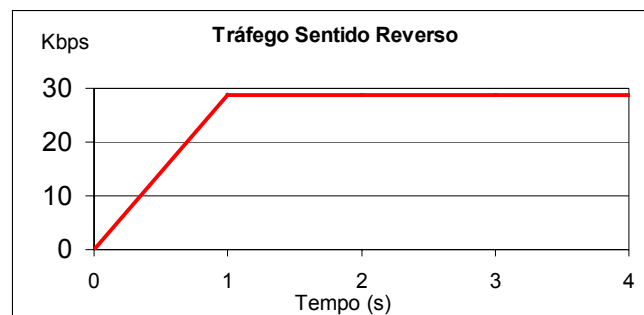
Aplica a técnica FIFO para a disciplina de saída para a fila do nó *n1* na transferência de *Acks* no sentido reverso, onde o primeiro *Ack* que chega a fila é o primeiro que sai, e quando há ocupação total da fila, os novos *Acks* entrantes são descartados.

#### 6.1.1 Tráfego Unidirecional

Quando utilizada em casos de tráfego unidirecional com alto grau de assimetria a técnica *FIFO* apresenta o seguinte comportamento:

##### 6.1.1.1 Desempenho no Sentido Reverso

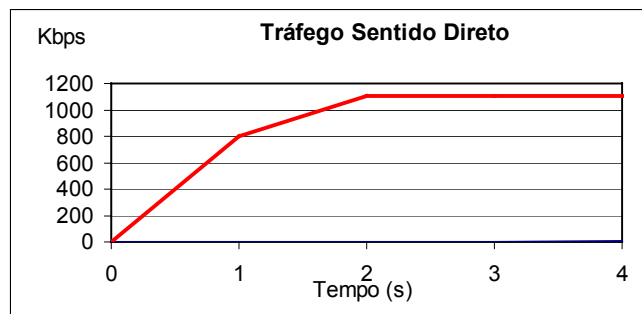
Apresenta ocupação total da conexão com o tráfego de *Acks* em resposta aos pacotes de dados da conexão no sentido direto conforme mostrado na Figura 5.



**Figura 5** – Ocupação da conexão no sentido Reverso

### 6.1.1.2 Desempenho no Sentido direto

Apesar da largura de faixa de 2Mbps da conexão no sentido direto, devido ao longo tempo de espera pelos *Acks* de confirmação aos pacotes de dados enviados, oferece desempenho máximo de 1.2 Mbps.



**Figura 6** – Ocupação da conexão no sentido Direto

### 6.1.2 Tráfego Bidirecional

Quando utilizada em casos de tráfego bidirecional com alto grau de assimetria a técnica *Drop Tail* apresenta para o gerenciamento da fila o seguinte comportamento:

#### 6.1.2.1 Análise da Fila

Verificamos que a programação dos pacotes na fila não sofrem alteração na ordem de entrada e saída, porém há um tempo de atraso considerável, como podemos verificar nos gráficos das Figuras 7 e 8. Os *Acks* 255 e 256 que possuem um afastamento temporal de 23 ms na entrada e de 872 ms na saída da fila, o que ocorre devido ao tempo de espera necessário para o envio de dois pacotes de dados, intermediários aos *Acks*.

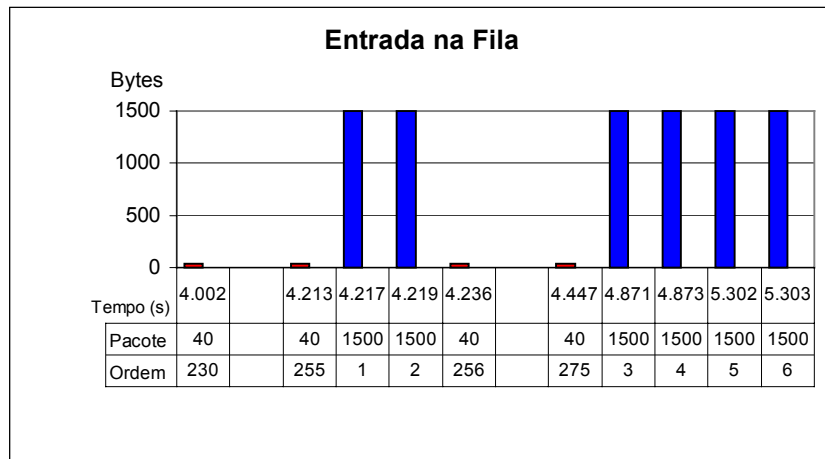


Figura 7: Seqüência de *Acks* e pacotes de dados que entram na fila

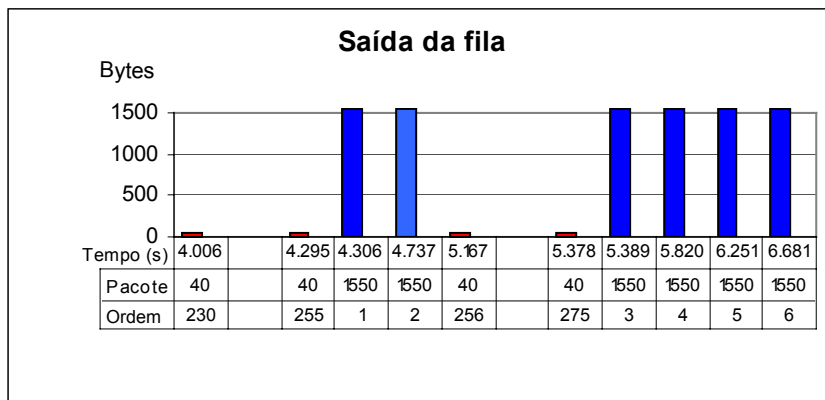


Figura 8: Seqüência de *Acks* e pacotes de dados na saída da fila

### 6.1.2.2 Desempenho no Sentido Reverso

Verificamos a ocupação total da conexão no sentido reverso com pacotes de dados e *Acks*, sendo que há uma queda acentuada na quantidade de *Acks* enviados imediatamente após o início do tráfego no sentido direto.

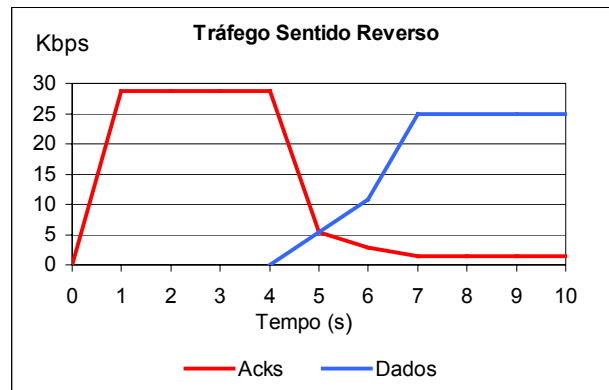


Figura 9 – Desempenho no sentido Reverso

### 6.1.2.3 Desempenho no Sentido Direto

Devido a alta ocupação com pacotes de dados da conexão de baixa velocidade no sentido reverso, há um enfileiramento de *Acks* atrás deste pacotes de dados, ocasionando um longo tempo de espera para confirmação e envio de novos pacotes de dados o que degrada o desempenho no sentido direto, como mostrado no gráfico da Figura 10.

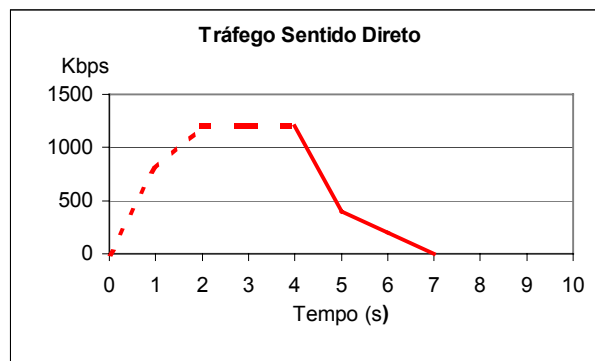


Figura 10 – Desempenho da conexão no sentido direto

## 6.2 Ack Filtering

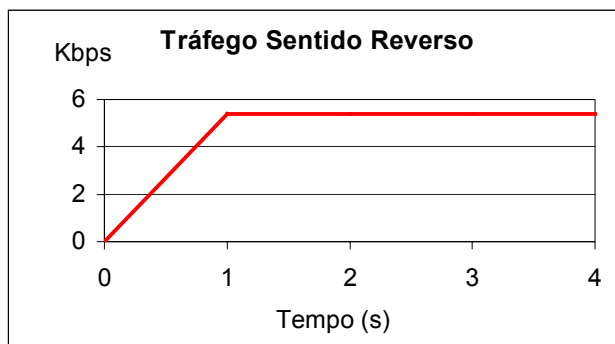
Aplica a técnica *Ack Filtering* na disciplina de saída da fila do nó *n1* para a transferência de *Acks* no sentido reverso. Verifica a fila e envia o *Ack* de última ordem descartando todos os outros.

### 6.2.1 Tráfego Unidirecional

Quando utilizada em casos de tráfego unidirecional com alto grau de assimetria a técnica *Ack Filtering* apresenta o seguinte comportamento:

#### 6.2.1.1 Desempenho no Sentido Reverso

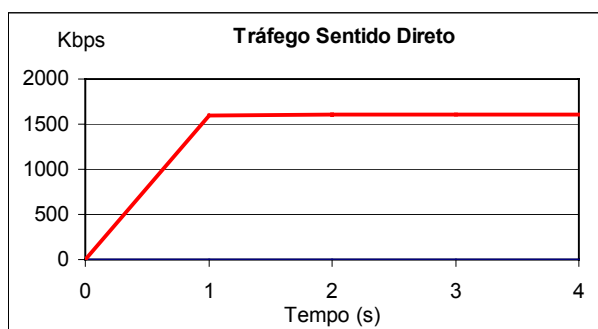
Apresenta baixa ocupação da conexão com o tráfego de *Acks* em resposta aos pacotes de dados da conexão no sentido direto como mostrado na Figura 11. Neste caso, visto que só existem *Acks* na fila para serem enviados, o processo de filtragem torna-se bastante eficaz, reduzindo consideravelmente a quantidade de *Acks* a serem enviados na conexão reversa.



**Figura 11** – Ocupação da conexão no sentido Reverso

#### 6.2.1.2 Desempenho no Sentido direto

Apresenta acréscimo de desempenho na transferência de pacotes de dados no sentido direto, de alta velocidade, quando comparada a técnica de disciplina de filas FIFO.



**Figura 12:** Ocupação da conexão no sentido Direto

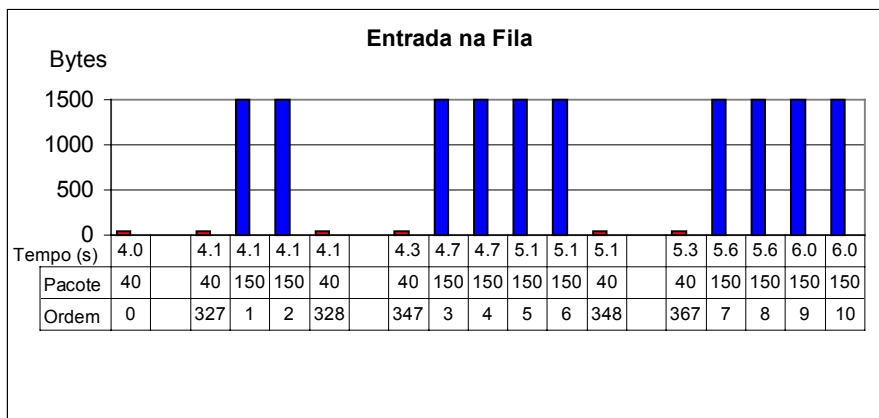


### 6.2.2 Tráfego Bidirecional

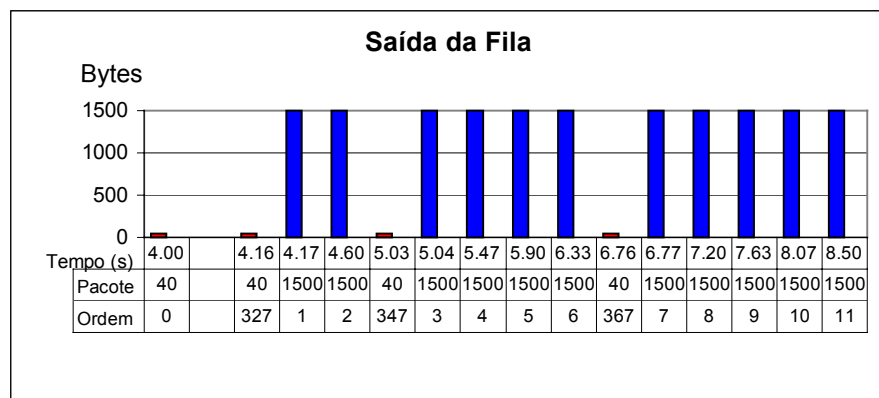
Quando utilizada em casos de tráfego bidirecional com alto grau de assimetria a técnica *Ack Filtering* apresenta o seguinte comportamento:

#### 6.2.2.1 Análise da Fila

Verificamos que a programação dos pacotes de dados na fila não sofre qualquer tipo de alteração, porém para o caso dos *Acks*, quando comparada a entrada com saída da fila, observa-se que alguns *Acks* foram suprimidos, como pode ser observado nos gráficos das Figuras 13 e 14.



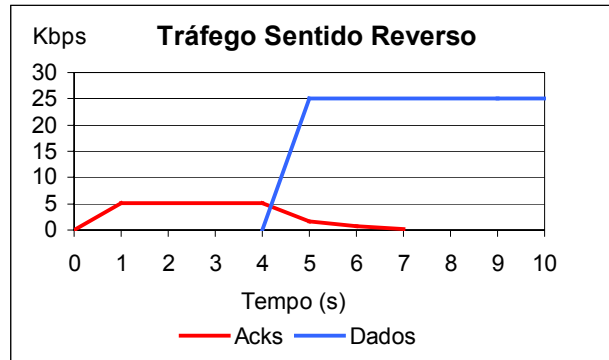
**Figura 13** – Sequência de Pacotes de dados e *Acks* que entram na fila



**Figura 14** – Sequência de Pacotes de dados e *Acks* que saem da fila

### 6.2.2.2 Desempenho no Sentido Reverso

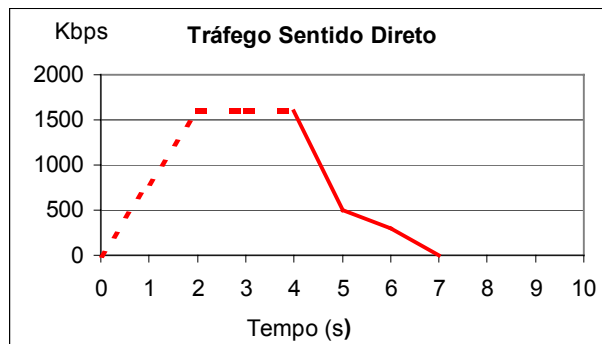
Devido a pequena quantidade de *Acks* que trafegam na conexão, consegue-se alta ocupação da conexão no sentido reverso com pacotes de dados (figura 15).



**Figura 15:** Ocupação no sentido Reverso

### 6.2.2.3 Desempenho no Sentido Direto

Devido a alta ocupação com pacotes de dados da conexão de baixa velocidade no sentido reverso, há um enfileiramento de *Acks* atrás destes pacotes de dados, ocasionando um longo tempo de espera para confirmação e envio de novos pacotes de dados o que degrada o desempenho no sentido direto (figura 16).



**Figura 16 –** Ocupação da conexão no sentido direto

### 6.3 Acks -First Scheduling

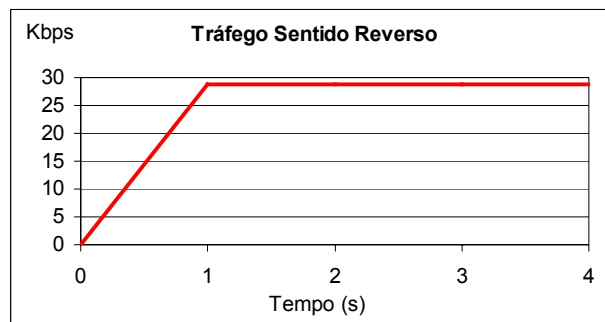
Aplica a técnica *Acks-First Scheduling* na disciplina de saída da fila do nó *n1* para a transferência de *Acks* no sentido reverso. Verifica a fila e envia o próximo *Ack* com prioridade sobre qualquer pacote de dados.

#### 6.3.1 Tráfego Unidirecional

Quando utilizada em casos de tráfego bidirecional com alto grau de assimetria a técnica *Acks-First Scheduling* apresenta o seguinte comportamento:

##### 6.3.1.1 Desempenho no Sentido Reverso

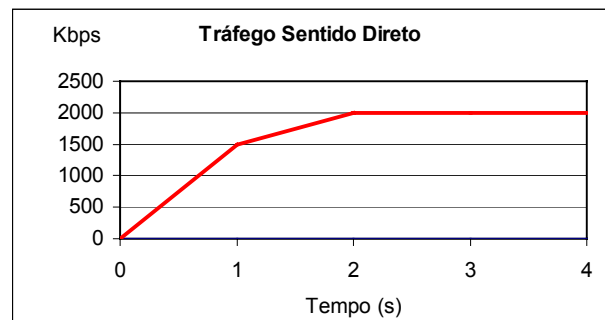
Apresenta ocupação total da conexão com o tráfego de *Acks* em resposta aos pacotes de dados da conexão no sentido direto, como pode ser verificado na Figura 17.



**Figura 17** – Ocupação da conexão no sentido Reverso

##### 6.3.1.2 Sentido direto

Propicia acréscimo de desempenho na transferência de pacotes de dados no sentido direto, de alta velocidade, quando comparada a técnica de disciplina de filas *Ack Filtering*.



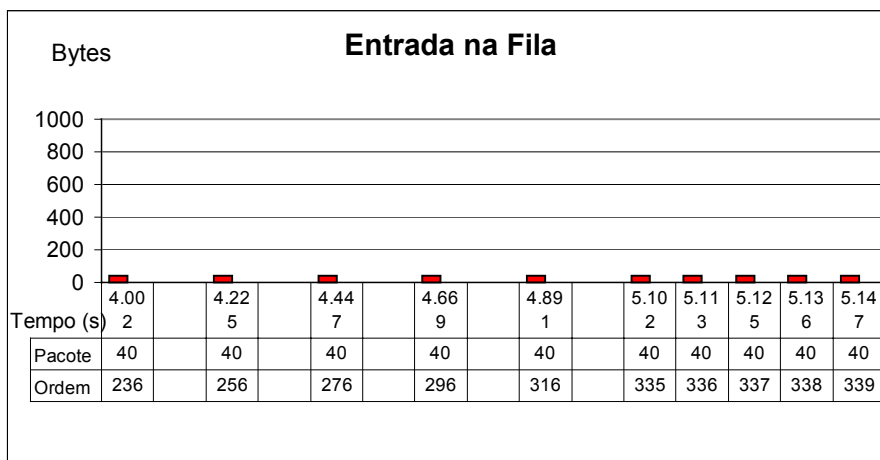
**Figura 18** – Ocupação da conexão no sentido Direto

### 6.3.2 Tráfego Bidirecional

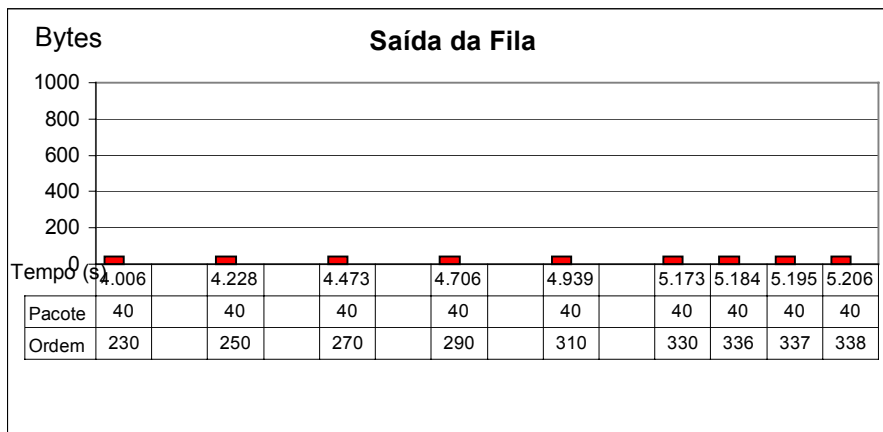
Para o caso de transferência bidirecional com alto grau de assimetria de tráfego, o comportamento e desempenho da técnica *Acks-First Scheduling* apresenta o seguinte comportamento:

#### 6.3.2.1 Análise da Fila

Devido ao elevado grau de assimetria de faixa verificamos que existem somente *Acks* na fila esperando para serem enviados.



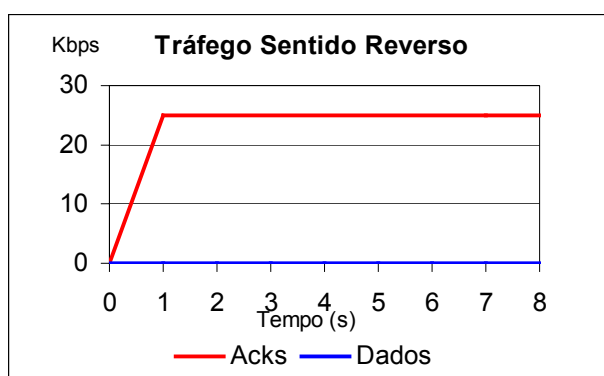
**Figura 19** – Sequência de *Acks* que entram na fila



**Figura 20** – Sequência de *Acks* que saem da fila

### 6.3.2.2 Desempenho no Sentido Reverso

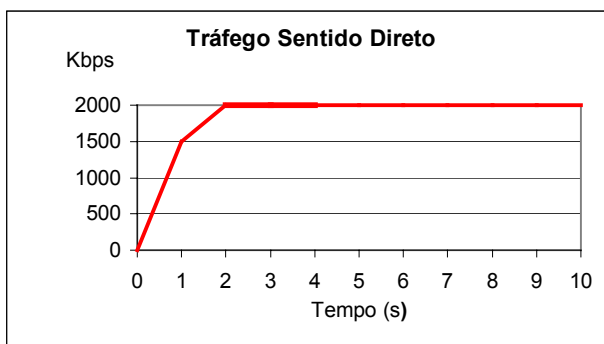
Verificamos queda abrupta de desempenho na transferência de pacotes de dados no sentido reverso devido a prioridade dada aos *Acks* sobre os pacotes de dados, podendo chegar a transferência nula de pacotes de dados quando o grau de assimetria for muito elevado ( $k \gg 1$ ). Isto ocorre se a quantidade de *Acks* entrantes na fila for grande o suficiente para haver sempre um *Ack* esperando para ser transmitido, o que faz com que os pacotes de dados esperem indefinidamente sua transmissão e sejam descartados.



**Figura 21** – Ocupação da conexão no sentido Reverso

### 6.3.2.3 Desempenho no Sentido Direto

Devido a prioridade dada aos *Acks* sobre os pacotes de dados na gerência da fila no sentido reverso, a técnica *Acks-First Scheduling* apresenta desempenho elevado na transferência de dados no sentido direto, como apresentado na Figura 22.



**Figura 22:** Ocupação no sentido direto

## 6.4 Solução Proposta

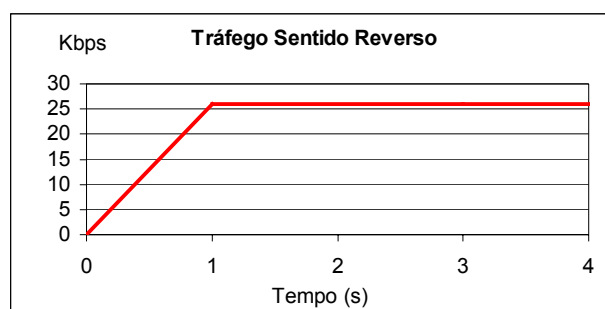
A técnica aqui proposta pretende corrigir as deficiências verificadas nas técnicas *Ack Filtering* e *Acks-First Scheduling* bem como aproveitar os ganhos apresentados por estas técnicas quando utilizadas em conexões com situação de alto grau de assimetria.

### 6.4.1 Tráfego Unidirecional

Quando utilizada em casos de tráfego bidirecional com alto grau de assimetria a Técnica Proposta apresenta o seguinte comportamento:

#### 6.4.1.1 Desempenho no Sentido Reverso

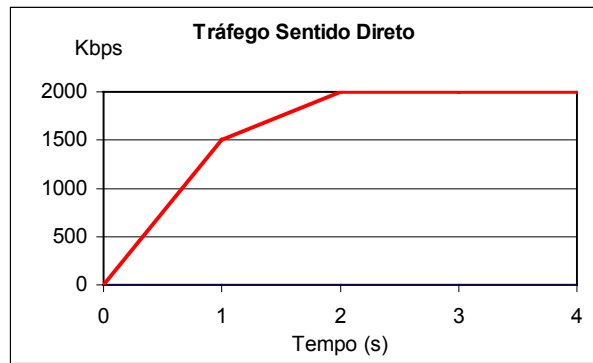
Apresenta ocupação total da conexão com o tráfego de *Acks* em resposta aos pacotes de dados da conexão no sentido direto.



**Figura 23** – Ocupação da conexão no sentido Reverso

#### 6.4.1.2 Desempenho no Sentido direto

Propicia acréscimo de desempenho na transferência de pacotes de dados no sentido direto, de alta velocidade, com desempenho semelhante a técnica de disciplina de filas *Acks-First Scheduling*.



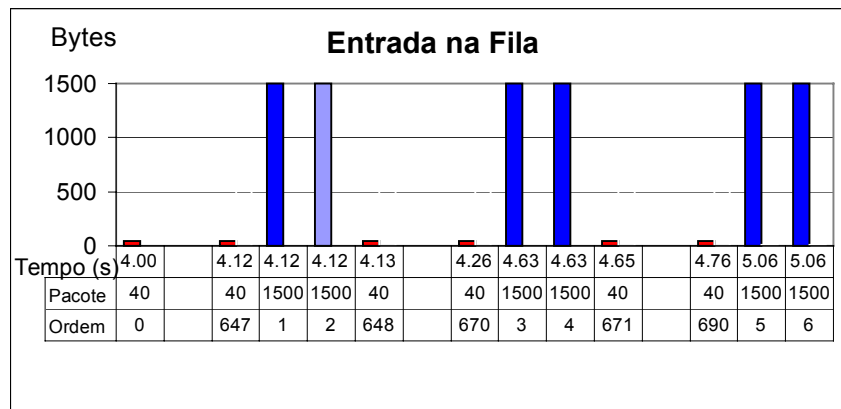
**Figura 24** – Ocupação da conexão no sentido Direto

### 6.4.2 Tráfego Bidirecional

Para o caso de transferência bidirecional com alto grau de assimetria de tráfego, o desempenho da técnica proposta apresentou vantagens em relação as técnicas apresentadas, ou seja, combina o ganho de desempenho da técnica *Ack Filtering* para o tráfego no sentido reverso com o ganho no desempenho da técnica *Acks-First Scheduling* para o tráfego no sentido direto.

#### 6.4.2.1 Análise da Fila

Verificamos que a programação dos pacotes de dados na fila sofre alteração, visto que neste caso somente grupos de dois pacotes apresentam-se para serem transmitidos, e para o caso dos *Acks*, quando comparada a entrada com saída da fila, observa-se que alguns *Acks* foram suprimidos, restando apenas um *Ack* a cada pacote de dados a ser transmitido.



**Figura 25** – Seqüência de Pacotes que entram na fila

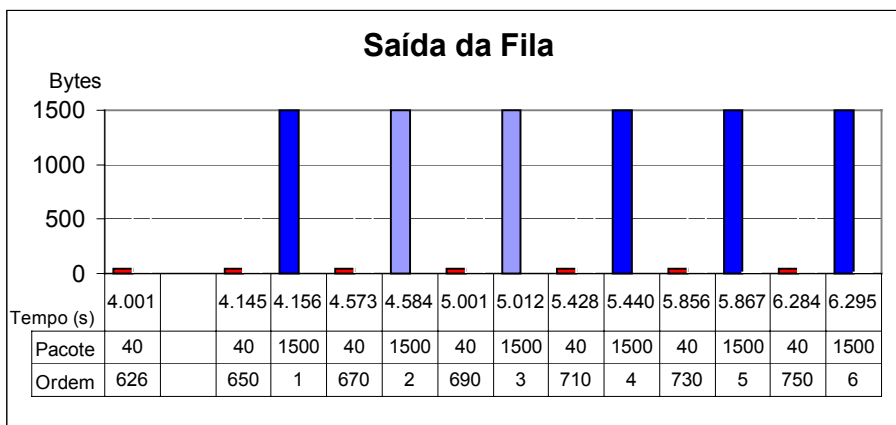
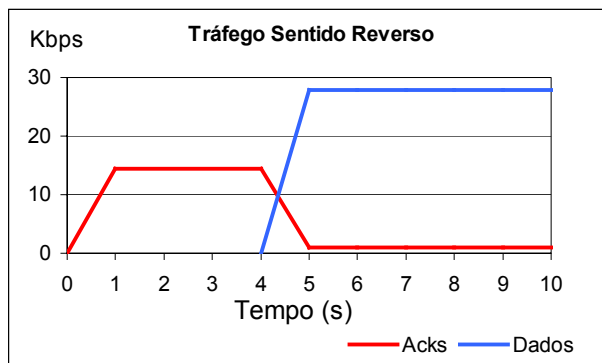


Figura 26 – Seqüência de Pacotes que saem da fila

### 6.4.2.2 Desempenho no Sentido Reverso

Assim como *Acks-First Scheduling* a técnica propõe a prioridade dos *Acks* em relação aos pacotes de dados, porém se preocupa em evitar que a transferência de pacotes de dados seja nula quando o grau de assimetria for muito elevado ( $k \gg 1$ ). Para isto implementa um algoritmo de disciplina de fila que controla a prioridade atribuída aos *Acks*. Verifica se há pacotes de dados esperando para serem enviados, e caso existam, implementa uma regra de prioridade controlada de *Acks* sobre os pacotes de dados, ou seja, envia uma quantidade de *Acks* determinada em relação ao comprimento da fila e então obrigatoriamente envia um pacote de dados. Com esta ação temos na saída da fila uma seqüência de vários *Acks* seguidos de um pacote de dados a cada comprimento da fila. Quando utilizada em conjunto com *Ack Filtering*, os *Acks* enfileirados a frente de cada pacote de dados são filtrados e então temos na fila uma seqüência de um *Ack* seguido por um pacote de dados. Com a diminuição na quantidade de *Acks* enviados, consegue-se um aumento na taxa de transferência de pacotes de dados no sentido reverso.

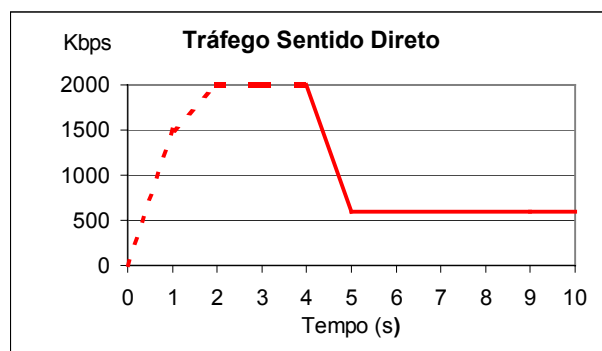




**Figura 27** – Ocupação da conexão no sentido Reverso

### 6.4.2.3 Desempenho no Sentido Direto

Assim como a técnica *Acks First Scheduling*, a solução proposta implementa a prioridade dos *Acks* sobre os pacotes de dados na gerência da fila no sentido reverso. Com isto consegue desempenho elevado na transferência de dados no sentido direto. Este desempenho está relacionado com o tempo que os *Acks* gastam para trafegar no sentido reverso da conexão e com tempo gasto para transferir pacotes de dados que também trafegam na conexão. O tempo gasto com a transferência de pacotes de dados está relacionado com o tamanho destes pacotes. Sendo assim, o desempenho para a transferência de pacotes de dados da conexão no sentido direto, está relacionado com o tamanho dos pacotes de dados que são enviados na conexão no sentido reverso.



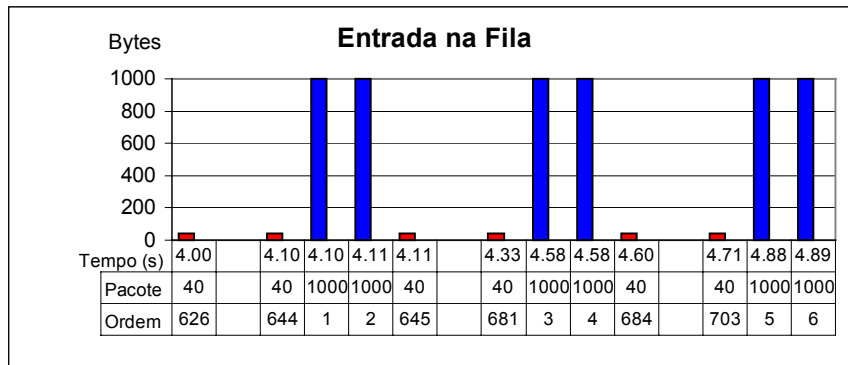
**Figura 28** – Ocupação da conexão no sentido Direto

### 6.4.3 Análise para pacotes de 1 Kbytes

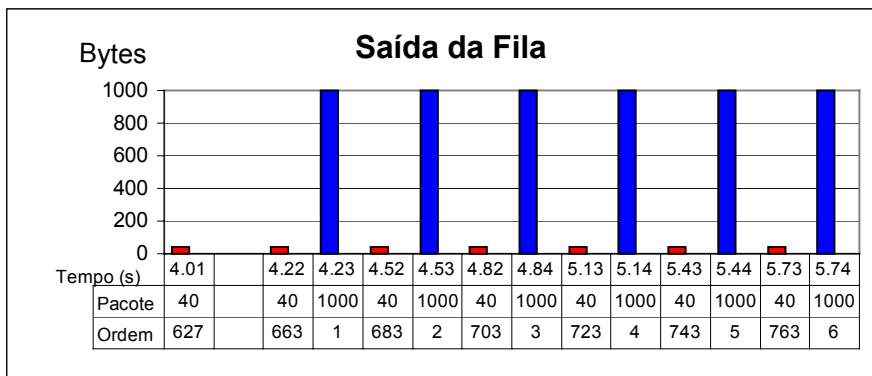
Utilizando pacotes de 1000 bytes no sentido reverso da conexão a Técnica Proposta apresenta o seguinte comportamento:

### 6.4.3.1 Análise da Fila

Apresenta comportamento semelhante ao verificado para o caso de pacotes de dados de 1,5 Kbytes, diferindo apenas nos tempos de entrada e saídas dos *Acks* e pacotes de dados.



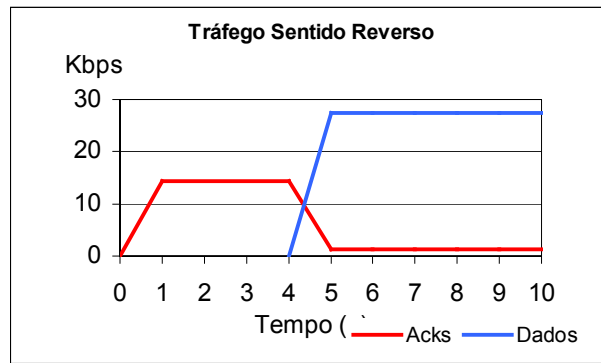
**Figura 29** – Seqüência de Pacotes que entram na fila



**Figura 30** – Seqüência de Pacotes que saem da fila

### 6.4.3.2 Desempenho no Sentido Reverso

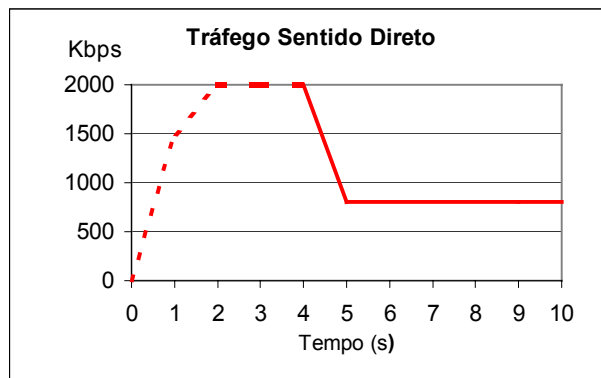
Apresenta desempenho semelhante ao oferecido quando do uso de pacotes de 1,5 kbytes.



**Figura 31:** Ocupação da conexão no sentido Reverso

### 6.4.3.3 Desempenho no Sentido Direto

Apresenta acréscimo de performance quando comparada com ao caso em que utiliza pacotes de 1,5 Kbytes.



**Figura 32 –** Ocupação da conexão no sentido Reverso

### 6.4.4 Análise para pacotes de 500 bytes

Utilizando pacotes de 500 bytes no sentido reverso da conexão a Técnica Proposta apresenta o seguinte comportamento:

#### 6.4.4.1 Análise da Fila

Apresenta comportamento semelhante ao verificado para o caso de pacotes de dados de 1,5 Kbytes, diferindo apenas nos tempos de entrada e saídas dos *Acks* e pacotes de dados.

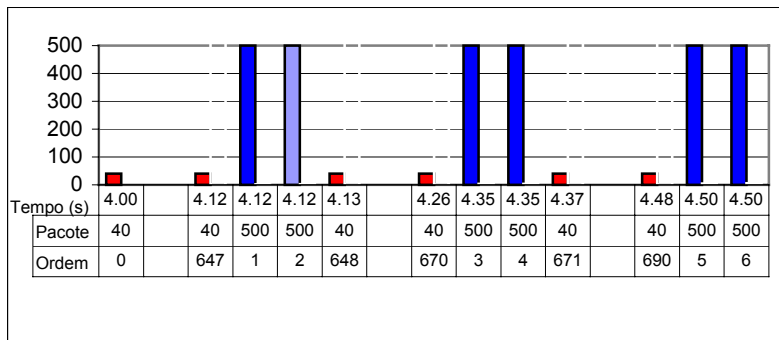


Figura 33 – Seqüência de Pacotes que entram na fila

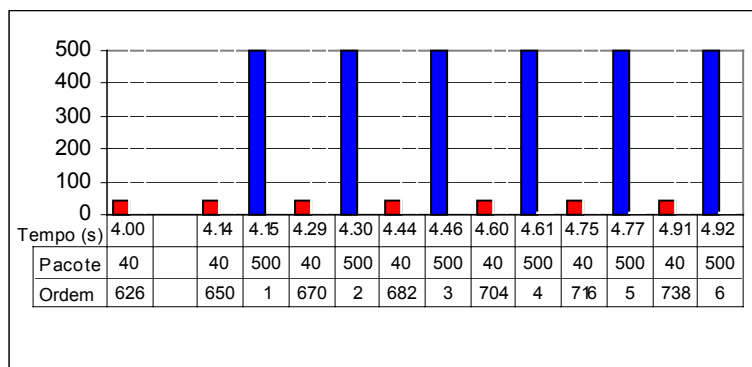


Figura 34 – Seqüência de Pacotes que saem da fila

### 6.4.4.2 Desempenho no Sentido Reverso

Apresenta desempenho semelhante ao oferecido quando do uso de pacotes de 1 kbytes.

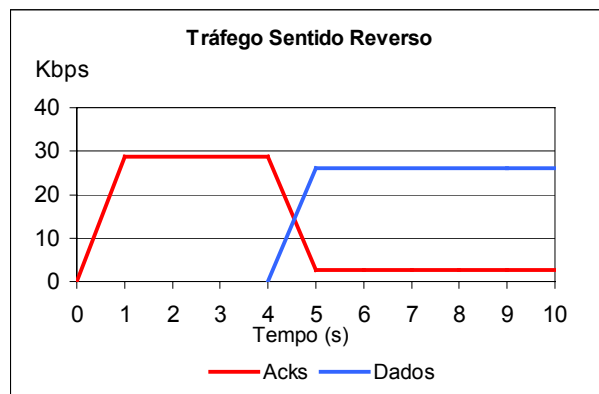


Figura 35 – Ocupação da conexão no sentido Reverso

### 6.4.4.3 Desempenho no Sentido Direto

Apresenta acréscimo de performance quando comparada com ao caso em que utiliza pacotes de 1,0 Kbytes.

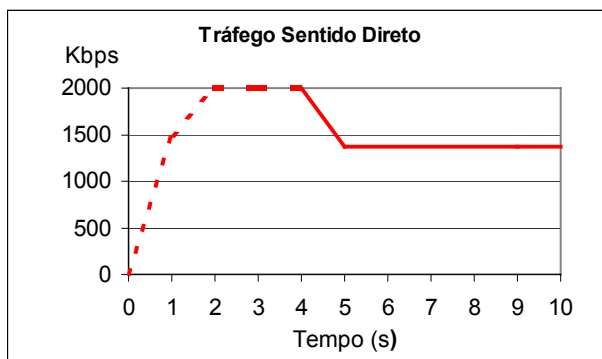


Figura 36 – Ocupação da conexão no sentido Direto

## 6.5 Comparativo de Desempenho

A tabela da Figura 37, apresenta um comparativo de desempenho entre as técnicas estudadas tanto para o caso de tráfego unidirecional quanto para o caso de tráfego bidirecional.

Comparativo de Desempenho			
Tráfego	Unidirecional		Bidirecional
Sentido	Direto (Dados - Mbps)	Reverso (Dados - Kbps)	Direto (Dados -Mbps)
<i>FIFO</i>	1,2	28,8	0,0
<i>Ack Filtering</i>	1,5	28,8	0,0
<i>Acks-First Scheduling</i>	2,0	0,0	2,0
Proposta - MTU 1500 bytes	2,0	26,0	0,6
Proposta - MTU 1000 bytes	2,0	26,0	0,8
Proposta - MTU 500 bytes	2,0	26,0	1,2

Figura 36 – Tabela comparativa de desempenho

Neste comparativo podemos observar que a técnica proposta apresenta desempenho elevado para tráfego unidirecional, com um desempenho equivalente as outras técnicas aqui estudadas

Já para o caso de tráfego bidirecional, onde acentua-se a característica de assimetria de faixa devido ao tráfego nos dois sentidos da conexão, podemos verificar que a técnica proposta apresenta desempenho elevado no sentido reverso aliado a garantia de transferência de pacotes de dados no sentido direto da conexão.

O desempenho no sentido direto esta relacionado com o tamanho dos pacotes de dados enviados. A medida que diminuimos o tamanho destes pacotes, obtemos um acréscimo na performance, devido ao menor tempo gasto no enfileiramento dos Acks atrás do pacote de dados.

# Capítulo 7 - Conclusão

## 7.1 Conclusão

Neste trabalho analisamos a partir de resultados de simulações de modelos, algumas técnicas de disciplina de filas. Estas técnicas foram desenvolvidas com o propósito de corrigir as deficiências do TCP para conexões com grau elevado de assimetria.

A partir do estudo do comportamento do tráfego em conexões que apresentam grau médio de assimetria, quando aplicadas estas técnicas para disciplina de saída da fila no sentido reverso da conexão assimétrica, verificamos um acréscimo na performance na transferência de pacotes de dados. A técnica *Ack Filtering* apresenta ganho de desempenho no sentido reverso, enquanto que a técnica *Acks-First Scheduling* apresenta ganho no sentido direto da conexão.

Já quando utilizamos a conexão para tráfego com alto grau de assimetria, como é o caso do tráfego bidirecional, onde acrescentamos na assimetria de largura de faixa a assimetria provocada pelo tráfego, verificamos uma queda acentuada no tráfego de pacotes de dados no sentido direto para a técnica *Ack Filtering* e queda acentuada no tráfego no sentido reverso para a técnica *Acks-First Scheduling*.

Em decorrência da análise das características de desempenho de tráfego apresentadas nos resultados das simulações neste trabalho implementadas, buscamos desenvolver um novo modelo com o propósito de garantir as vantagens das técnicas aplicadas e simultaneamente corrigir as deficiências por elas apresentadas.

Com este objetivo, é aplicado neste trabalho um algoritmo de disciplina de filas para a saída da fila, no sentido reverso da conexão assimétrica, desenvolvido a partir do conceito básico da técnica *Acks First Scheduling*. Também propõe a garantia de prioridade dos *Acks* sobre os pacotes de dados, porém não na totalidade, ou seja, há uma programação desta prioridade.

O desafio é permitir que os *Acks* exerçam a prioridade sobre os pacotes de dados, mas neste caso, pelo menos um pacote de dados é enviado a cada comprimento de fila, garantindo desta forma tanto a transferência de pacotes de dados no sentido direto da conexão, como também a garantia de transferência de pacotes de dados no sentido reverso.

Neste trabalho utilizamos conjuntamente com o modelo proposto a técnica *Ack Filtering*, com o objetivo de otimizar o uso da conexão no sentido reverso, e somado a isto experimentamos também uma variação no tamanho dos pacotes de dados enviados.

Os resultados obtidos a partir das simulações do modelo proposto apresentaram ganhos significativos em relação as técnicas estudadas, ou seja, foram preservadas as vantagens de cada técnica e corrigidas parte das deficiências por elas apresentadas.

## 7.2 Sugestão para futuros trabalhos

Como sugestão para trabalhos futuros, consideramos interessante o aprofundamento na análise da técnica proposta em relação ao desempenho no sentido direto, a preocupação com os *Acks* infrequentes e a questão da compressão do *Ack*. Também consideramos promissor a utilização da técnica proposta em conjunto com outras técnicas existentes, e em particular com a técnica *Ack Reconstruction* [BP01].



## Referências Bibliográficas

- [ADG<sup>+</sup>00] Allman, M., Dawkins, S., Glover, D., Griner, J., Henderson, T., Heidemann, J., Kruse, H., Ostermann, S., Scott, K., Semke, J., Touch, J., and D. Tran, *"Ongoing TCP Research Related to Satellites"*, RFC2760. February 2000.
- [ADSL] ADSL Fórum Web Site: <http://www.adsl.com>
- [AFP02] M. Allman, S. Floyd, C. Partridge. *"Increasing TCP's Initial Window"*, RFC3390, October 2002.
- [AGPS] M. Allman Glenn/Sterling, V. Paxson, W. Stevens, *"TCP Congestion Control"*, RFC2581, April 1999.
- [ANSI98] ANSI Standard T1.413, *"Network to Customer Installation Interfaces - Asymmetric Digital Subscriber Lines (ADSL) Metallic Interface"*, November 1998.
- [APS99] Allman, M., Paxson, V., and W. Stevens, *"TCP Congestion Control"*, RFC2581, April 1999
- [Atk94] R. Atkinson, *"Default IP MTU for use over ATM AAL5"*, RFC1626, May 1994.

- [BA00] Chadi Barakat & Eitan Altman, *"On ACK Filtering on a Slow Reverse Channel"*, INRIA Sophia Antipolis - France Workshop on Quality of Future Internet Services September, 2000, Berlin, Germany.
- [Bal98] H. Balakrishnan, *"Challenges to Reliable Data Transport over Heterogeneous Wireless Networks"*, Ph.D. Thesis, University of California at Berkeley, USA, August 1998.
- [BAFW03] E. Blanton, M. Allman, K. Fall, L. Wang. *"A Conservative Selective Acknowledgment (SACK)-based Loss Recovery Algorithm for TCP"*, RFC3517, April 2003.
- [BBD<sup>+</sup>01] Bormann, C. Burmeister, C. Degermark, M. Fukushima, H. Hannu, H. Jonsson, E. Hakenberg, R. Koren, T., Le, K., Liu, Z., Martensson, A., Miyazaki, A., Svanbro, K., Wiebke, T., Yoshimura, T., Zheng, H., *"RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP ESP and uncompressed"*, RFC3095, July 2001.
- [BKG01] Border, J., Kojo, M., Griner, J., Montenegro, G., and Z. Shelby, *"Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations"*, RFC3135, June 2001.
- [BL99] G. Bathrick, F. Ly, *"Definitions of Managed Objects for the ADSL Lines"*, RFC2662 August 1999.
- [Bor99] Bormann, C. *"The Multi-Class Extension to Multi-Link PPP"*, RFC2686, Sept 1999.

- [BP01] H. Balakrishnan, V. N. Padmanabhan, *"How Network Asymmetry Affects TCP"*, IEEE Communications Magazine, April 2001.
- [BPFS02] H. Balakrishnan, V. N. Padmanabhan, G. Fairhurst, M. Sooriyabandara. *"TCP Performance Implications of Network Path Asymmetry"*, RFC3449, December 2002.
- [BPK99] H. Balakrishnan, V. N. Padmanabhan, and R. H. Katz, *"The Effects of Asymmetry on TCP Performance,"* ACM MONET, 1999.
- [Cj99] S. Casner, V. Jacobson, *"Compressing IP/UDP/RTP Headers for Low-Speed Serial Links"*, RFC2508, February 1999.
- [Com88] Comer, Douglas E., *"Interligação em Rede com TCP/IP, Princípios, protocolos e arquitetura"*, vol. 1, 3<sup>a</sup> Edição, Ed. Campus, Rio de Janeiro, 1988.
- [DDI<sup>+</sup>01] Duros, E., Dabbous, W., Izumiyama, H., Fujii, N., and Y. Zhang, *"A link Layer tunneling mechanism for unidirectional links"*, RFC3077, March 2001.
- [DMK<sup>+</sup>01] Dawkins, S., Montenegro, G., Kojo, M. and V. Magret, *"End- to-end Performance Implications of Slow Links"*, BCP 48, RFC3150, July 2001.
- [DNP99] Degermark, M., Nordgren, B., and Pink, S., *"IP Header Compression"*, RFC2507, February 1999.
- [FB00] Victor Firoiu and Marty Borden. *"A Study of Active Queue Management for Congestion Control"*, Infocom, pages 1435-1444, 2000.

- [FG99] B. Fox, B. Gleeson, "*Virtual Private Networks Identifier*", RFC2685, September 1999.
- [Fh99] S. Floyd, T. Henderson. "*The New Reno Modification to TCP's Fast Recovery Algorithm*", RFC2582, April 1999.
- [FLH<sup>+</sup>00] Farinacci, D., Li, T., Hanks, S., Meyer, D., Traina, P., "*Generic Routing Encapsulation (GRE)*", RFC2784, March 2000.
- [FRS01] Sally Floyd, Ramakrishna Gummadi, and Scott Shenker, "*Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management*", AT&T Center for Internet Research at ICSI, August 1, 2001.
- [FSS<sup>+</sup>01] Fairhurst, G., Samaraweera, N.K.G, Sooriyabandara, M., Harun, H., Hodson, K., and R. Donardio, "*Performance Issues in Asymmetric Service Provision using Broadband Satellite*", IEE Proc. Commun, Vol.148, No.2, April 2001.
- [GH99] D. Grossman, J. Heinanen, "*Multiprotocol Encapsulation over ATM Adaptation Layer 5*", RFC2684, September 1999.
- [Hua01] Hua Qin, "*A New Approach To Improving TCP Performance Over ADSL*", A Thesis MASTER OF SCIENCE, Rensselaer Polytechnic Institute, Troy, New York July 2001.
- [ITU01] ITU-T Recommendation E.681, "*Traffic Engineering Methods For IP Access Networks Based on Hybrid Fiber/Coax System*", September 2001.

- [ITU99] ITU-T Recommendation G.992.1, "*Asymmetrical Digital Subscriber Line (ADSL) Transceivers*", July 1999.
- [Jac88] V. Jacobson, "*Congestion Avoidance and Control*," *Proc. ACM SIGCOMM 88*, Aug. 1988.
- [Jac89] Jacobson, V., "*Compressing TCP/IP Headers for Low-Speed Serial Links*". RFC1144, November 1989.
- [Jac90] V. Jacobson, "*Compressing TCP/IP headers for low speed serial links*", RFC1144, February 1990.
- [Jac92] Jacobson, V., Braden, R., Borman, D., "*TCP Extensions for High Performance*", RFC 1323, Mai 1992.
- [JN84] John Nagle, "*Congestion Control in IP/TCP Internetworks*", RFC896, January 1984.
- [KCB03] T. Koren, S. Casner, C. Bormann. "*IP Header Compression over PPP*", RFC3544, July 2003.
- [Kom00] Komer, "*TCP/IP*", The 9th IEEE International Conference on Computer Communication and Networks (IC3N'00), Las Vegas, Nevada, October 2000.
- [KVR98] L. Kalampoukas, A. Varma and K.K. Ramakrishnan, "*Improving TCP Throughput over Two-Way Asymmetric Links: Analysis and Solutions*" in *Proc. of Sigmetrics'98*, June 1998.'
- [Kwo99] T.C. Kwok, "*Residential Broadband Architecture over ADSL and G.Lite (G.992.2): PPP over ATM*," *IEEE Communications Magazine*, May 1999.

- [LMS97] Lakshman, T.V., Madhow, U., and B. Suter, "*Window-based Error Recovery and Flow Control with a Slow Acknowledgement Channel: A Study of TCP/IP Performance*", Proc. IEEE INFOCOM, Vol.3, Kobe, Japan, 1997.
- [LS00] T. V. Lakshman and Bernhard Suter, "*TCP/IP Performance with Random Loss and Bidirectional Congestion*", IEEE/ACM TRANSACTIONS ON NETWORKING, Vol. 8, no 5, october 2000.
- [McG92] G. McGregor "*The PPP Internet Protocol Control Protocol (IPCP)*", RFC1332 May 1992.
- [MD90] . Mogul, S. Deering. "*Path MTU Discovery*", RFC1191, November 1990.
- [MDLB00] Martin May, Christophe Diot, Bryan Lyles, Jean Bolot, "*Influence of Active Queue Management Parameters on Aggregate Traffic Performance*", Institut National de Recherche en Informatique et en Automatique, August 2000, France.
- [MLECS99] L. Mamakos, K. Lidl, J. Evarts ,D. Carrel, D. Simone, R. Wheeler, "*A Method for Transmitting PPP Over Ethernet (PPPoE)*",RFC2516, February 1999.
- [MMF<sup>+</sup>96] M. Mathis J. Mahdavi ,S. Floyd, A. Romanow, "*TCP Selective Acknowledgment Options*", RFC2018, October 1996.
- [NS04] NS, UCB/LBNL/VINT Network Simulator - ns (version 2), November 2004, <http://www.isi.edu/nsnam/ns>.
- [Pad98] V. N. Padmanabhan, "*Addressing the Challenges of Web Data Transport*" Ph.D. thesis, University of California at Berkeley, USA, Sept. 1998.

- [PAD<sup>+</sup>99] Paxson, V., Allman, M., Dawson, S., Heavens, I., and B. Volz, "*Known TCP Implementation Problems*", RFC2525, March 1999.
- [Per89] D.Perkins, "*The Point-to-Point Protocol: A Proposal for Multi-Protocol Transmission of Datagrams Over Point-to-Point Links*", RFC1134, November 1989.
- [PSK00] K. Phanse, L. DaSilva and K. Kidambi, "*Characterization of Performance of TCP/IP over PPP over ATM over Asymmetric Links*," to appear at the IEEE, I3CN00.
- [RA02] B. Ray, R. Abbi "Definitions of Managed Objects for High Bit-Rate DSL - 2nd generation (HDSL2) and Single-Pair High-Speed Digital Subscriber Line (SHDSL) Lines", RFC3276, May 2002.
- [RA04] B. Ray, R. Abbi "*Definitions of Managed Objects for Very High Speed Digital Subscriber Lines (VDSL)*", RFC3728, February 2004.
- [RBGD02] J. Reynolds, R. Braden, S. Ginoza, A. De La Cruz. "*Internet Official Protocol Standards*" RFC3300, November 2002.
- [RFC791] Defense Advanced Research Projects Agency, "*Internet Protocol*", RFC791, September 1981.
- [RFC793] Defense Advanced Research Projects Agency, "*Transmission Control Protocol*", RFC793, September 1981.
- [RFC2481] Ramakrishnan K., and S. Floyd, "*A Proposal to add Explicit Congestion Notification (ECN) to IP*", RFC2481.

- [RFC3135] J. Border, M. Kojo, J. Griner, G. Montenegro, Z. Shelby, “*Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations*”, RFC3135, June 2001.
- [Sva02] K. Svanbro, “*Lower Layer Guidelines for Robust RTP/UDP/IP Header Compression*”, RFC3409, December 2002.
- [Sim92] W. Simpson “*The Point-to-Point Protocol (PPP) for the Transmission of Multi-protocol Datagrams over Point-to-Point Links*”, RFC1331, May 1992.
- [Sim94] W. Simpson, Ed., “*The Point-to-Point Protocol (PPP)*” RFC1661, July 1994.
- [Sta01] Stallings William, “*Data & Computer Communications*”, Prentice Hall, sixth Edition, 2001.
- [Ste97] W. Stevens “*TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms*”, RFC 2001, January 1997.
- [STT<sup>+</sup>02] A.Singh, R. Turner, R. Tio, S. Nanji, “*Layer Two Tunnelling Protocol (L2TP) Over ATM Adaptation Layer 5 (AAL5)*”, RFC3355, August 2002.
- [TKB02] B. Thompson, T. Koren, B. Buffam. “*PPP Over Asynchronous Transfer Mode Adaptation Layer 2 (AAL2)*”, RFC3336, December 2002.
- [TSC02] Y. TJoens, B. Sales, P. Crivellari, “*Layer Two Tunnelling Protocol (L2TP): ATM access network extensions*”, RFC 3301, June 2002



- [Xia00] Xiaoning He, "*Tcp Performance Analysis of a DMT-based ADSL System, The Pennsylvania State University*", Proc. of the 25th Annual Conference on Local Computer Networks, LCN'00, Tampa, FL, November 2000.

## ANEXO 1

### Parâmetros de configuração da conexão – NS2

```
# TCP_ASSIM.TCL
```

```
source util2.tcl
```

```
set ns [new Simulator]
```

```
set n0 [$ns node]
```

```
set n1 [$ns node]
```

```
set n2 [$ns node]
```

```
set n3 [$ns node]
```

```
$ns color 0 blue
```

```
$ns color 1 red
```

```
set f [open out3.tr w]
```

```
$ns trace-all $f
```

```
set nf [open out.nam w]
```

```
$ns namtrace-all $nf
```

```
set redtrace [open red.tr w]
```

```
set upwin 0
```

```
set downwin 100
```

```
set ackSize 40
```

```
set rbw 28.8Kb
```

```
set window 100
```

```
set rbw 0
```

```
set rqsize 20
```

```
set qsize 10
```

```
set rgw "DropTail"
```

```
set nonfifo true
```

```
set ackstest true
```

```
set reconsacks false
```

```
set filteracks true
```

```
# -----
```

```
$ns duplex-link $n0 $n1 10Mb 1ms DropTail
```

```
$ns simplex-link $n1 $n2 28.8kb 50ms DropTail
```

```
$ns simplex-link $n2 $n1 2Mb 5ms DropTail
```

```
$ns duplex-link $n2 $n3 10Mb 1ms DropTail
```

```
$ns duplex-link-op $n0 $n1 orient right
```

```
$ns simplex-link-op $n1 $n2 orient right
```

```
$ns simplex-link-op $n2 $n1 orient left
```

```
$ns duplex-link-op $n2 $n3 orient left
```

```
#$ns simplex-link-op $n1 $n2 queuePos 0.5
```

```
# -----
```

```
set tcp [new Agent/TCP]
```

```
$tcp set class_ 1
```

```
$tcp set packetSize_ 1450
```

```

set sink [new Agent/TCPSink]
$sink set packetSize_ 40
$ns attach-agent $n0 $tcp
$ns attach-agent $n3 $sink
$ns connect $tcp $sink

set ftp [new Application/FTP]
$ftp attach-agent $tcp

set tcp3 [new Agent/TCP]
$tcp3 set class_ 1
$tcp3 set packetSize_ 1450

set sink3 [new Agent/TCPSink]
$sink3 set packetSize_ 40
$ns attach-agent $n3 $tcp3
$ns attach-agent $n0 $sink3
$ns connect $tcp3 $sink3

set ftp3 [new Application/FTP]
$ftp3 attach-agent $tcp3

$ns at 0.0 "$ftp3 start"

$ns at 4.0 "$ftp start"

#configQueue $ns $n1 $n2 $sacksfirst
# -----

```

## Técnica Proposta – NS2

```

#semantic-packetqueue.cc

Packet*
SemanticPacketQueue::deque_ackstest() {
    int i = 0;
    Packet* p = head_;
    Packet* pp = NULL;
    Packet* pk = head_;
    Packet* pkk = NULL;
    packet_t type;
    // Se for somente download - data_count == 0
    if (ack_count > 0 && data_count == 0) {
        while (p) {
            type = hdr_cmn::access(p)->ptype_;
            if (type == PT_ACK) {
                break; }

            pp = p;

```

```

        p = p->next_;
    }

    PacketQueue::remove(p, pp);
}
// Se upload - data_count != 0
else if (ack_count > 0 && count < 20) {
    while (p) {
        type = hdr_cmn::access(p)->ptype_;
        if (type == PT_ACK) {
            break;
        }
        pp = p;
        p = p->next_;
    }
    if (!p)
        PacketQueue::remove(p, pp);
    count ++;
}

// Se somente upload - ack_count = 0
else if (data_count > 0 ) {
    while (pk) {
        type = hdr_cmn::access(pk)->ptype_;
        if (type != PT_ACK) {
            break;
        }

        pkk = pk;
        pk = pk->next_;
    }

    if (!pk)

        pk = PacketQueue::deque();
        fprintf(stderr, "Packet: %d, pk: %d , pkk: %d \n",
data_count, pk, pkk);
        count = 0;
    }
    return p;
}

```