

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

IGOR RAMOS DE OLIVEIRA

Mecanismos e formas de acesso à
QoS (Quality of Service) em ambientes
Microsoft Windows XP e Windows Server 2003

Trabalho de Graduação.

Prof. Dr. Sérgio Luis Cechin
Orientador

Prof. Dr. João César Netto
Coordenador do Curso

Porto Alegre, dezembro de 2009.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitora de Graduação: Profa. Valquíria Linck Bassani

Diretor do Instituto de Informática: Prof. Flávio Rech Wagner

Coordenador da COMGRAD em Ciência da Computação: Prof. João César Netto

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

AGRADECIMENTOS

Agradeço à minha família por todo o apoio e incentivo dados durante os difíceis anos onde tive sempre de conciliar os estudos do curso de graduação com as atividades de trabalho e a vida social.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	5
LISTA DE FIGURAS	7
LISTA DE TABELAS	9
RESUMO	10
ABSTRACT	11
1 INTRODUÇÃO	9
2 BREVE HISTÓRICO DAS REDES DE COMPUTADORES	11
3 QUALITY OF SERVICE (QOS)	15
3.1 Métricas	15
3.1.1 Vazão.....	16
3.1.2 Atraso (<i>Delay</i>).....	17
3.1.3 Jitter.....	17
3.1.4 Taxa de erros.....	18
3.2 Modelos de implementação	18
3.2.1 Serviços Integrados (<i>IntServ – Integrated Services</i>).....	19
3.2.2 Serviços Diferenciados (<i>DiffServ – Differentiated Services</i>).....	21
3.2.3 IEEE 802.1p.....	24
4 QUALIDADE DE SERVIÇO EM AMBIENTES MICROSOFT WINDOWS XP E WINDOWS SERVER 2003	28
4.1 Arquitetura de QoS	28
4.1.1 Controle de tráfego.....	28
4.1.2 Componentes.....	30
5 TESTES DE IMPLEMENTAÇÃO	36
5.1 Ferramentas utilizadas	36
5.2 Configuração dos equipamentos	36
5.3 Pré-requisitos	37
5.4 Testes	40
6 CONCLUSÃO	53
REFERÊNCIAS	55

LISTA DE ABREVIATURAS E SIGLAS

AF	<i>Assured Forwarding</i>
API	<i>Application Programming Interface</i>
ARPA	<i>Advanced Research Projects Agency</i>
ARPANET	<i>Advanced Research Projects Agency Network</i>
CBQ	<i>Class Based Queuing</i>
CFI	<i>Canonical Format Indicator</i>
CFQ	<i>Class Based Fair Queuing</i>
DDR2	<i>Double Data Rate 2</i>
DiffServ	<i>Differentiated Services</i>
DNS	<i>Domain Name System</i>
DS	<i>Differentiated Service</i>
DSCP	<i>Differentiated Service Code Point</i>
EF	<i>Expedited Forwarding</i>
FIFO	<i>First In First Out</i>
FTP	<i>File Transfer Protocol</i>
Gbps	<i>Giga bit por segundo</i>
GPC	<i>Generic Packet Classifier</i>
GPedit	<i>Group Policy Editor</i>
GQoS	<i>Generic Quality of Service</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IETF	<i>Internet Engineering Task Force</i>
IntServ	<i>Integrated Services</i>
IP	<i>Internet Protocol</i>
IPv4	<i>Internet Protocol version 4</i>
IPv6	<i>Internet Protocol version 6</i>
Kbps	<i>Kilo bit por segundo</i>
LAN	<i>Local Area Network</i>

LCD	<i>Liquid Crystal Display</i>
Mbps	<i>Mega bit por Segundo</i>
ms	<i>Milisegundo</i>
NSF	<i>National Science Foundation</i>
NSFNET	<i>National Science Foundation Network</i>
OSI	<i>Open Systems Interconnection</i>
PARC	<i>Palo Alto Research Center</i>
Perfmon	<i>Windows Performance Monitor</i>
PHB	<i>Per Hop Behavior</i>
QoS	<i>Quality of Service</i>
RED	<i>Random Early Detection</i>
RFC	<i>Request for Comments</i>
RSVP	<i>Resource Reservation Protocol</i>
RTP	<i>Real-time Transport Protocol</i>
SFQ	<i>Stochastic Fair Queuing</i>
SLA	<i>Service Level Agreement</i>
SRI	<i>Stanford Research Institute</i>
TC	<i>Traffic Control</i>
TCMon	<i>Traffic Control Monitor</i>
TCP	<i>Transmission Control Protocol</i>
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>
Telnet	<i>Telecommunication network</i>
TOS	<i>Type of Service</i>
TPID	<i>Tag Protocol Identifier</i>
UCLA	<i>University of California, Los Angeles</i>
UCSB	<i>University of California, Santa Barbara</i>
UTP	<i>Unshielded Twisted Pair</i>
VID	<i>VLAN ID</i>
VLAN	<i>Virtual Local Area Network</i>
VoIP	<i>Voice over Internet Protocol</i>
WFQ	<i>Weighted Fair Queuing</i>
Winsock2	<i>Windows Sockets 2.0</i>
WRED	<i>Weighted Random Early Detection</i>

LISTA DE FIGURAS

FIGURA 2.1: Crescimento da ARPANET. (a) Dezembro de 1969. (b) Julho de 1970. (c) Março de 1971. (d) Abril de 1972. (e) Setembro de 1972. [TAN03].....	12
FIGURA 2.2: Desenho feito por Robert Metcalfe em 1976 ilustrando o conceito do padrão Ethernet [CHA09].	13
FIGURA 3.1: Efeito do <i>jitter</i> [SAN09].....	18
FIGURA 3.2: Alguns blocos funcionais da arquitetura IntServ. [POZ01].	20
FIGURA 3.3: Reserva de recursos com o protocolo RSVP. [POZ01].....	20
FIGURA 3.4: Campo TOS (<i>Type of Service</i>) do cabeçalho do protocolo IPv4. [DEL09].	22
FIGURA 3.5: Blocos funcionais do modelo DiffServ. [SAN09].....	24
FIGURA 3.6: Quadro Ethernet modificado pelo padrão IEEE 802.1q. [HPT09].....	25
FIGURA 4.1: Modelo arquitetural do TCP/IP implementado no <i>Windows XP</i> e <i>Windows Server 2003</i> . [TCP09].....	31
FIGURA 5.1: Propriedades avançadas do <i>driver</i> da placa de rede da estação transmissora. ..	39
FIGURA 5.2: Aba <i>Parameters</i> do TCMon instalado na estação transmissora.	39
FIGURA 5.3: Atuação das estações durante os testes.....	41
FIGURA 5.4: Fluxos criados no TCMon.	42
FIGURA 5.5: Parâmetros da interface de rede.....	42
FIGURA 5.6: Métricas do fluxo para a porta 5001 (mesmas métricas para os outros fluxos).43	
FIGURA 5.7: Filtro do fluxo para a porta 5001 (nos outros fluxos muda a porta de destino).43	
FIGURA 5.8: Adição de contadores no Perfmon.....	44
FIGURA 5.9: Perfmon após a adição do contador “Bytes transmitidos/s”.....	44
FIGURA 5.10: Perfmon mostrando a taxa de transferência em KB/s dos três fluxos.	45
FIGURA 5.11: Conteúdo do pacote capturado com o <i>Wireshark</i>	46
FIGURA 5.12: Métricas do fluxo para a porta UDP 5001.....	47
FIGURA 5.13: Métricas do fluxo para a porta UDP 5002.....	47
FIGURA 5.14: Métricas do fluxo para a porta UDP 5003.....	48

FIGURA 5.15: TCMon com as novas métricas.	48
FIGURA 5.16: Perfmon mostrando a taxa de transferência em KB/s dos três fluxos com as novas métricas.	49
FIGURA 5.17: Conteúdo de um pacote do fluxo para a porta UDP 5001, com o campo DSCP preenchido com o valor 0x30, correspondente a classe de serviço <i>Network Control</i>	50
FIGURA 5.18: Conteúdo de um pacote do fluxo para a porta UDP 5002, com o campo DSCP preenchido com o valor 0x18, correspondente a classe de serviço <i>Controlled load</i>	50
FIGURA 5.19: Conteúdo de um pacote do fluxo para a porta UDP 5003, com o campo DSCP preenchido com o valor 0x00, correspondente a classe de serviço <i>Best-effort</i>	51
FIGURA 5.20: Taxas de transferência dos fluxos alteradas dinamicamente no TCMon.	52

LISTA DE TABELAS

TABELA 3.1: Requisitos de uma aplicação VoIP	16
TABELA 3.2: Exemplos de vazão de algumas aplicações de rede.....	16
TABELA 3.3: Valores do campo DSCP para classes do AF	23
TABELA 3.4: Valores do campo <i>User Priority</i> recomendados pelo IEEE	26
TABELA 4.1: Valores padrão do <i>Windows</i> para as classes de serviço do IEEE 802.1p.....	33
TABELA 4.2: Valores padrão do <i>Windows</i> para as classes de serviço do DiffServ.....	33
TABELA 4.3: Configurações das <i>Group Policies</i> associadas ao <i>QoS Packet Scheduler</i>	34
TABELA 5.1: Configuração dos equipamentos.....	37

RESUMO

Este trabalho visa apresentar um estudo dos métodos e formas de obtenção de qualidade de serviço em redes locais (LANs - *Local Area Networks*) suportadas pelas plataformas *Microsoft Windows XP* e *Windows Server 2003*.

Nele, fizemos um breve relato da história das redes de computadores e dos principais eventos ocorridos nos últimos 50 anos que ajudaram a tornar as redes computacionais tão imprescindíveis nos dias de hoje. Também são destacados conceitos importantes relacionados aos mecanismos e tecnologias envolvidos na qualidade de serviço, como métricas e modelos de implementação.

Finalmente, são feitos testes e simulações de implementação da qualidade de serviço em um ambiente suportado pela plataforma *Windows XP*.

Palavras-Chave: Redes de Computadores, Qualidade de Serviço, Serviços Diferenciados, Serviços Integrados, IEEE 802.1p, Windows XP, Windows Server 2003.

TITLE: “Mechanisms and means of access to QoS (Quality of Service) in Microsoft Windows XP and Windows Server 2003”.

ABSTRACT

This work has as goal to present a study of methods and ways of achieving quality of service in local area networks (LANs) supported by Microsoft Windows XP and Windows Server 2003 platforms.

In this document, we made a brief of computer networks history and the main events of the last 50 years that helped make computer networks so essential today. Important concepts related to the mechanisms and technologies involved in the quality of service, such as metrics and implementation models are also highlighted.

Finally, tests and simulations of quality of service implementation in an environment supported by the Windows XP platform are performed.

Keywords: Computer Networks, Quality of Service, Differentiated Services, Integrated Services, IEEE 802.1p, Windows XP, Windows Server 2003.

1 INTRODUÇÃO

O protocolo IP (*Internet Protocol*) foi desenvolvido na década de 70 sob o slogan “IP sobre Tudo”, ou seja, o principal objetivo na concepção do protocolo era que o mesmo pudesse ser utilizado com as diversas tecnologias e meios físicos que existiam na época. Sendo assim, as decisões arquiteturais tomadas na concepção do protocolo visavam a simplicidade, o que fez com que o protocolo IP fosse concebido com algumas restrições técnicas, limitando o seu uso para aplicações com poucos requisitos de operação, como por exemplo, aplicações que permitiam a perda de pacotes e a existência de atrasos [SAN09].

Durante muito tempo o serviço tradicional oferecido pelo protocolo IP, conhecido como melhor-esforço (*best-effort*), o qual trata o tráfego de diferentes fluxos de dados de forma igualitária, foi adequado para as aplicações existentes até então: FTP (*File Transfer Protocol*), Telnet (*Telecommunication network*), aplicações de *e-mail*, etc.

Porém, com a popularização das redes de computadores e a evolução dos serviços e dispositivos de comunicação, com conseqüente aumento no tráfego de dados, surgiu a necessidade de um tratamento diferenciado para os fluxos de dados dos novos tipos de aplicações, principalmente as aplicações multimídias, as quais possuem exigências de tempo e sincronização para que operem com qualidade. Esse novo cenário fez com que surgisse a visão de convergência de serviços, acrescentando à premissa básica do “IP sobre Tudo” uma nova perspectiva: “Tudo sobre IP”.

É neste contexto que surge o QoS (*Quality of Service*), um conjunto de tecnologias que permitem o gerenciamento do tráfego de rede, possibilitando a entrega priorizada do tráfego de aplicações de tempo real que exigem mais confiabilidade que o *best-effort*.

Este trabalho está organizado da seguinte forma:

No capítulo 2 é feito um breve resumo da história das redes de computadores, focando principalmente nos fatos e acontecimentos responsáveis pelo surgimento e popularização da Internet e das redes locais.

No capítulo 3 são apresentados os conceitos de qualidade de serviço, suas principais métricas e modelos de implementação.

No capítulo 4 são descritos alguns componentes e suas formas de atuação dentro da arquitetura de QoS implementada nos sistemas operacionais *Windows XP* e *Windows Server 2003*.

No capítulo 5 são descritos os resultados de alguns testes realizados em um ambiente montado com o objetivo de comprovar o funcionamento da implementação da qualidade de serviço na plataforma *Windows XP*.

E, finalmente, no capítulo 6 são apresentadas as conclusões obtidas com este estudo.

2 BREVE HISTÓRICO DAS REDES DE COMPUTADORES

A busca por comunicação é antiga na sociedade humana. Nossos ancestrais já buscavam meios de se comunicar uns com os outros e, conforme as civilizações foram crescendo e se espalhando, a necessidade de comunicação a longa distância fez com que fossem desenvolvidos métodos que quebrassem as barreiras geográficas: sinais de fumaça, pombos-correio, etc. [SOA95].

A descoberta de que a comunicação poderia se dar através de sinais elétricos, com a invenção do telégrafo em 1835 por Samuel Morse, foi o embrião da comunicação moderna para longas distâncias. Quarenta anos depois, Alexander Graham Bell viria a inventar o telefone, dispositivo que é usado até hoje.

Passado o desafio da comunicação de voz a longas distâncias, um novo desafio começou a surgir no horizonte a partir do final da década de 1950: a comunicação de dados a longas distâncias. Assim como muitos dos avanços tecnológicos da humanidade, as redes de computadores tiveram como motivação inicial fins militares.

No auge da Guerra Fria e da corrida espacial, todas as comunicações militares dos Estados Unidos passavam pela rede de telefonia pública, considerada vulnerável. O Departamento de Defesa dos Estados Unidos começou então a buscar a construção de uma rede de comunicação altamente distribuída e tolerante a falhas. Após vários anos, alguns projetos descartados e ainda sem contar com um sistema melhor de comando e controle, a situação se agravou em outubro de 1957, quando a União Soviética lançou o primeiro satélite artificial, o *Sputnik*. [TAN03].

O presidente norte-americano na época, Dwight D. Eisenhower, preocupado com a situação, cria então uma organização de pesquisa de defesa: a ARPA (*Advanced Research*

Projects Agency). A ARPA mantinha contratos com algumas universidades e centros de pesquisa norte-americanos e, após alguns anos, em 1969 entrou no ar uma rede experimental composta por quatro nós: UCLA (*University of California, Los Angeles*), UCSB (*University of California, Santa Barbara*), SRI (*Stanford Research Institute*) e *University of Utah*. Esta rede ficou conhecida como ARPANET. A ARPANET cresceu rapidamente nos três primeiros anos, conforme mostra a figura 2.1:

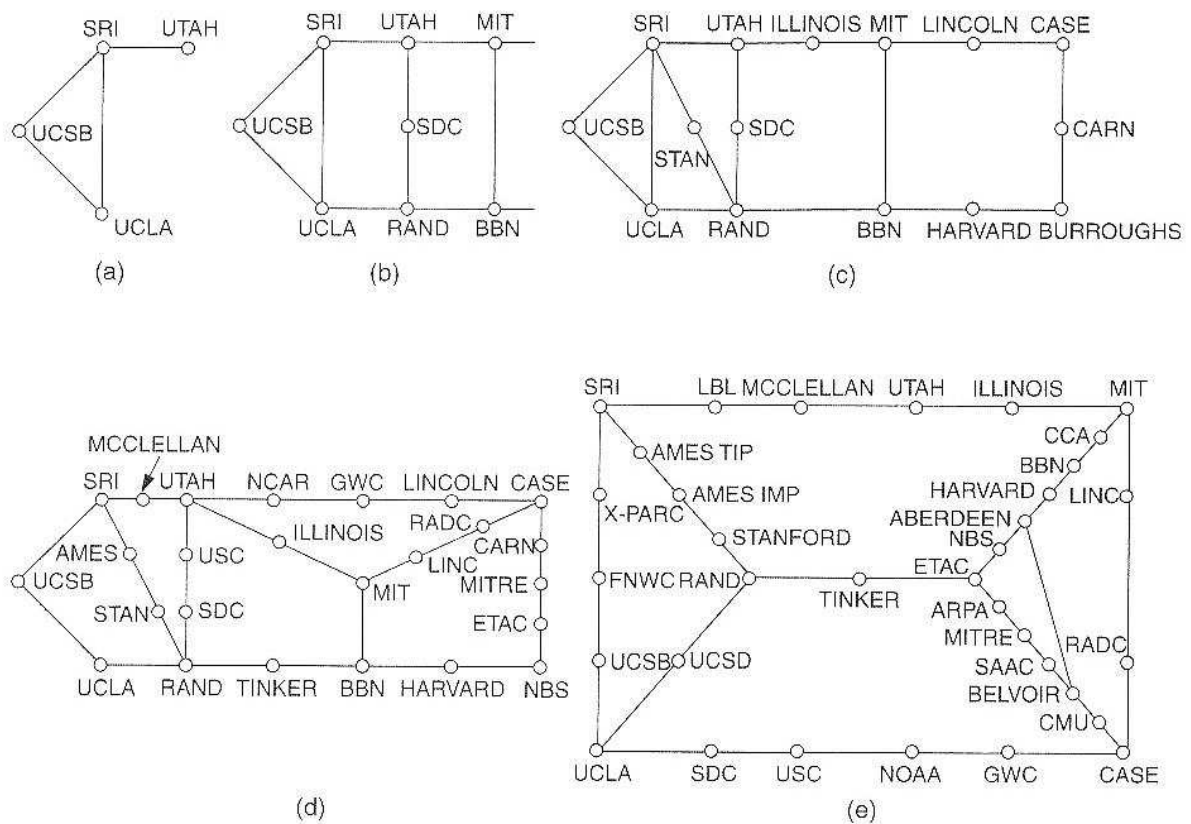


FIGURA 2.1: Crescimento da ARPANET. (a) Dezembro de 1969. (b) Julho de 1970. (c) Março de 1971. (d) Abril de 1972. (e) Setembro de 1972. [TAN03].

Com o uso, percebeu-se que os protocolos da ARPANET não eram totalmente adequados para execução em redes múltiplas. Essa constatação levou ao desenvolvimento do protocolo TCP/IP, o qual tornou mais fácil a conexão das LANs (*Local Area Networks*) à ARPANET, propiciando um crescimento ainda maior da rede. Esse crescimento fez com que a instalação de novos *hosts* se torna-se muito dispendiosa, o que gerou a necessidade da criação de um sistema que organizasse as máquinas em domínios e fizesse o mapeamento dos

nomes dos *hosts* em seus respectivos endereços IP. Foi então que surgiu o DNS (*Domain Name System*). [TAN03].

Para entrar na ARPANET, uma universidade precisava ter um contrato de pesquisa com o Departamento de Defesa dos Estados Unidos, um privilégio para poucas. Dessa forma, a NSF (*National Science Foundation*), percebendo a importância que a ARPANET estava tendo para as pesquisas universitárias, resolveu criar uma rede similar a ARPANET, porém aberta a todos os grupos de pesquisa universitários. Foi então que em 1986 surgiu a NSFNET. [TAN03]. Com a extinção da ARPANET em 1990, a NSFNET viria a se tornar o que hoje conhecemos por Internet.

Paralelamente ao desenvolvimento da Internet, foi desenvolvida e aperfeiçoada uma importante tecnologia a qual foi um dos principais saltos tecnológicos que permitiram a popularização das redes: a Ethernet. O nascimento desta tecnologia foi propiciado pela necessidade que muitas organizações (universidades, empresas, etc.) tinham em interligar não somente seus computadores em redes geograficamente distribuídas, mas também os computadores em suas redes locais.

A Ethernet foi criada em 1973 por Robert Metcalfe nos laboratórios do PARC (*Palo Alto Research Center*) da Xerox. Ela foi desenvolvida tendo como base o trabalho do pesquisador Norman Abramson da *University of Hawaii*, o qual havia inventado uma solução por rádio de ondas curtas para conectar usuários situados em ilhas próximas ao computador principal em Honolulu. Esse sistema era chamado de ALOHANET. [TAN03].

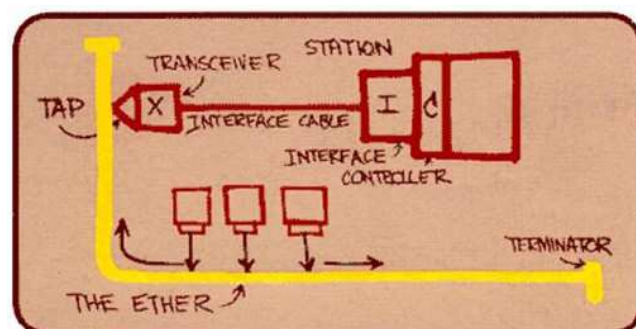


FIGURA 2.2: Desenho feito por Robert Metcalfe em 1976 ilustrando o conceito do padrão Ethernet [CHA09].

Embora a Ethernet tenha sido bem-sucedida, a Xerox acabou mostrando pouco interesse. Dessa forma, Metcalfe deixou a empresa em 1979 para promover o uso de redes locais e formou sua própria empresa, a 3Com. Em 1983 a tecnologia Ethernet foi padronizada pelo IEEE (*Institute of Electrical and Electronics Engineers*) através da especificação 802.3.

A ARPANET/NSFNET e o padrão Ethernet foram os grandes responsáveis pelo sucesso atual das redes de computadores, dando origem à Internet e às LANs, respectivamente. Podemos até mesmo dizer que revolucionaram a computação, pois talvez conseguíssemos viver sem processadores *quad-core*, memórias DDR2 e monitores de LCD, mas viver sem as redes locais e a Internet nos dias de hoje é praticamente impossível.

3 QUALITY OF SERVICE (QOS)

O modelo de serviço normalmente oferecido pelas tecnologias mais usadas em redes de computadores, tanto na camada de rede (protocolo IP) como na camada de enlace (protocolo Ethernet), é o *best-effort* (melhor esforço), onde todos os pacotes são tratados sem distinção com base no FIFO (*First In First Out*). Porém, com a demanda de tráfego cada vez maior devido ao surgimento de novas aplicações, principalmente as aplicações multimídia, criou-se a necessidade de mecanismos que possibilitassem um maior controle sobre os recursos da rede. Esses mecanismos deveriam implementar algum tipo de priorização, o qual permitisse que o fluxo de determinadas aplicações (como aplicações de missão crítica) tivesse prioridade de entrega sobre o fluxo de aplicações não tão críticas.

Qualidade de Serviço (*Quality of Service*), ou QoS, é o termo genérico empregado para designar o conjunto de mecanismos e tecnologias capazes de fornecer tratamento diferenciado para os diferentes tipos de tráfego presente nas redes. Sua utilização visa um controle mais eficiente dos recursos da rede e, como consequência, uma otimização no uso da banda. [BLO09].

3.1 Métricas

Existem aplicações que possuem requisitos que devem ser atendidos para que elas possam executar com qualidade. Esses requisitos são expressos através de algumas métricas (vazão, atrasos, perdas, etc.), cujos valores devem estar dentro de limites bem definidos (valor mínimo, valor máximo). Como exemplo, pode ser mencionada uma aplicação de Voz sobre IP (VoIP) com algumas centenas de canais de voz simultâneos, a qual possui os seguintes requisitos: [SAN09]

TABELA 3.1: Requisitos de uma aplicação VoIP

Métrica	Valor
Vazão	≥ 2 Mbps
Atraso	≤ 250 ms
Disponibilidade	$\geq 99,5\%$

Fonte: [SAN09].

As métricas de QoS usadas para expressar os requisitos das aplicações são as seguintes:

- Vazão;
- Atraso (*Delay*);
- *Jitter*;
- Taxa de erros.

3.1.1 Vazão

Vazão (ou taxa de transferência) é a quantidade de informação que pode ser transferida em um determinado espaço de tempo. Normalmente é expressa em Kbps, Mbps ou Gbps. A largura de banda de uma rede deve acomodar a taxa de transferência de todas as aplicações que ela se propõe a suportar.

O atendimento deste requisito deve ser levado em conta no momento do projeto da rede. A tabela 3.2 ilustra a vazão típica de algumas aplicações:

TABELA 3.2: Exemplos de vazão de algumas aplicações de rede

Aplicação	Vazão
Voz	10 Kbps a 100 Kbps

Vídeo	100 Kbps a 1 Mbps
Imagens Médicas	10 Mbps a 100 Mbps
Realidade Virtual	80 Mbps a 150 Mbps

Fonte: [SAN09].

3.1.2 Atraso (*Delay*)

Atraso (ou latência) é definido como o tempo decorrido entre o envio de um pacote e sua entrega ao destino. Em termos gerais, o atraso total de uma rede é o resultado da soma da influência de três fatores:

- Atraso de propagação do meio de transmissão: tempo necessário para que o sinal (elétrico, óptico, rádio) propague-se no meio de transmissão (par trançado, fibra óptica, ar);
- Taxa de transmissão: também conhecida como largura de banda, é a capacidade máxima de transmissão de dados suportada pelas tecnologias (interfaces de rede, meios de transmissão, etc.) presentes na rede. Normalmente é expressa em Kbps, Mbps ou Gbps;
- Tempo de processamento dos equipamentos: é o tempo gasto pelos equipamentos de rede (hosts, *hubs*, *switches*, roteadores) para realizar o processamento dos pacotes (controle de fluxo, roteamento, etc.).

3.1.3 Jitter

Pode ser definido como a variação dos tempos de atraso na entrega dos pacotes, podendo até mesmo ocasionar a entrega fora de ordem dos mesmos (este problema pode ser solucionado com o uso de um serviço de entrega confiável, como o TCP (*Transmission Control Protocol*) ou o RTP (*Real-time Transport Protocol*)). A figura 3.1 ilustra o efeito do *jitter*:

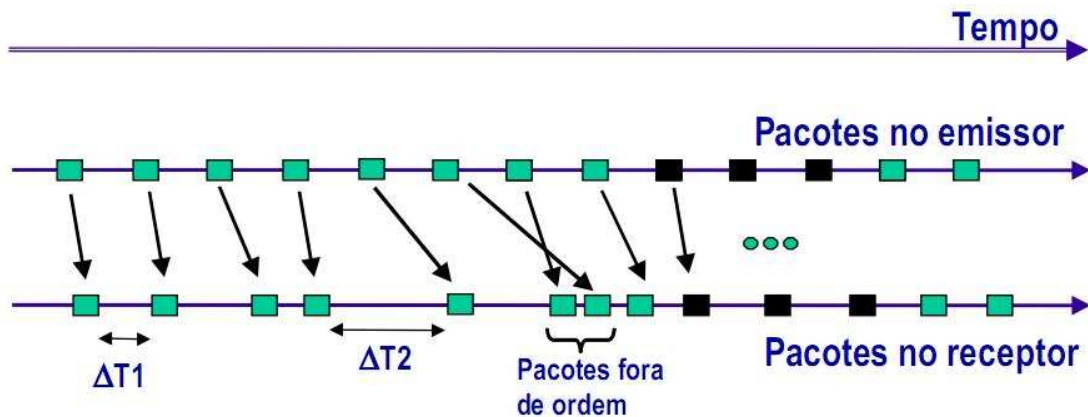


FIGURA 3.1: Efeito do *jitter* [SAN09].

Como pode ser visto na figura 3.1, embora os pacotes sejam transmitidos pelo emissor em intervalos regulares, eles são entregues ao receptor com uma periodicidade variável ($\Delta T1$ e $\Delta T2$) devido à variação na latência da rede.

3.1.4 Taxa de erros

Esta métrica pode ser definida como sendo a razão entre a quantidade de pacotes que não foram entregues ao destino e a quantidade total de pacotes, medidos em um determinado período de tempo.

A falha na entrega dos pacotes pode ocorrer devido a vários fatores, tais como: descarte pelos roteadores/*switches* (erros nos cabeçalhos), descarte pelo sistema operacional (erros nos dados), descarte devido à insuficiência de banda (estouros de *buffer*), falhas em equipamentos, etc.

3.2 Modelos de implementação

Várias pesquisas ([INT09], [DIF09], [IEE09]) começaram a ser realizadas a partir do momento que se percebeu a importância da qualidade de serviço para as novas aplicações. Estas pesquisas resultaram em alguns modelos de implementação, dentre os quais os mais importantes são:

- Na camada de rede:

- Serviços Integrados (IntServ – *Integrated Services*);
- Serviços Diferenciados (DiffServ – *Differentiated Services*);
- Na camada de enlace:
 - IEEE 802.1p.

3.2.1 Serviços Integrados (IntServ – *Integrated Services*)

O modelo de Serviços Integrados (IntServ – *Integrated Services*) foi proposto por um grupo de pesquisa da IETF (*Internet Engineering Task Force*), através da RFC 1633. Este modelo é caracterizado pela reserva prévia de recursos, a qual deve ser feita explicitamente pela aplicação, através de um mecanismo de solicitação de serviços.

O modelo IntServ adiciona mais duas classes de serviços ao já existente serviço de melhor esforço (*Best-effort*): Serviços Garantidos (*Guaranteed Service*) e Serviços de Carga Controlada (*Controlled Load Services*). A classe de Serviços Garantidos é utilizada por aplicações que exigem garantias rígidas de largura de banda, atraso e perdas de pacotes. Já a classe de Serviços de Carga Controlada é direcionada para aplicações que exigem serviços semelhantes aos oferecidos pela classe de melhor esforço, mas com algumas garantias de condição de tráfego confiável e sem congestionamento. [POZ01].

A arquitetura IntServ é composta por quatro elementos:

- Mecanismo de solicitação de serviços: normalmente o protocolo RSVP (*Resource Reservation Protocol*) é utilizado como mecanismo de solicitação de serviços. As aplicações e os roteadores utilizam esse protocolo de sinalização para trocarem mensagens de controle de alocação de recursos;
- Controle de admissão: responsável pelo gerenciamento dos recursos disponíveis, no que diz respeito às solicitações de reserva de recursos feitas pelas aplicações;

- Classificador de pacotes: responsável pelo processo de classificação dos pacotes de cada fluxo, distribuindo-os entre as classes de serviço/filas adequadas;
- Escalonador de pacotes: responsável por retirar os pacotes das filas para envio, segundo alguma política de filas.

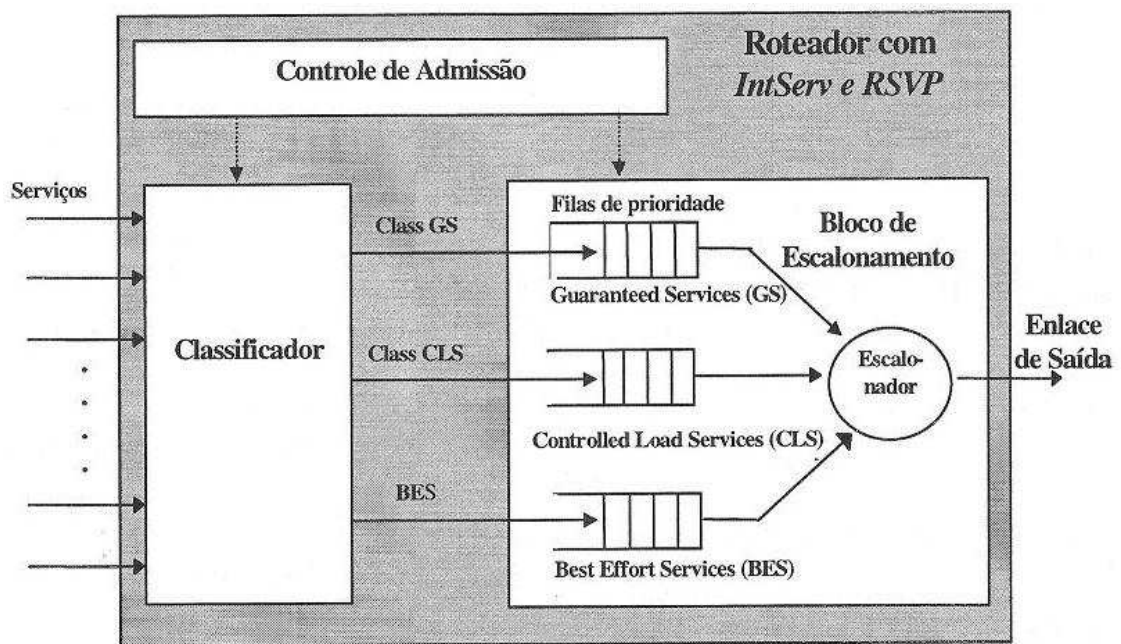


FIGURA 3.2: Alguns blocos funcionais da arquitetura IntServ. [POZ01].

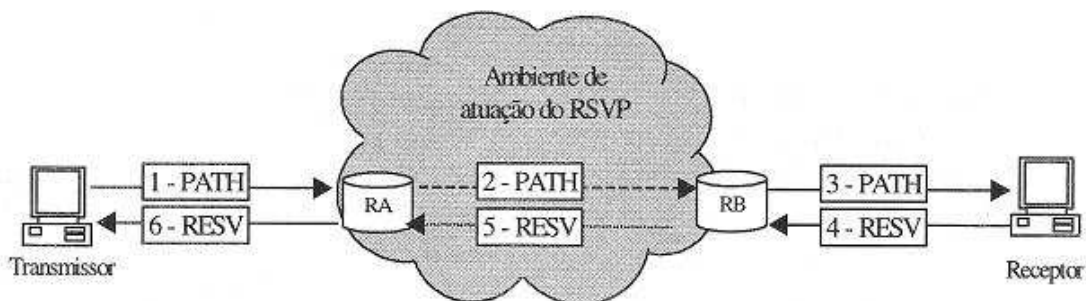


FIGURA 3.3: Reserva de recursos com o protocolo RSVP. [POZ01].

A figura 3.3 ilustra o processo de reserva de recursos realizado com o protocolo RSVP. Primeiramente, a aplicação que irá realizar a transmissão do fluxo de dados envia uma

mensagem denominada PATH para a aplicação que irá receber o fluxo. Essa mensagem contém a especificação das características do fluxo. Após receber a mensagem, a aplicação de destino deve enviar outra mensagem, denominada RESV, a qual irá solicitar a cada roteador no caminho até a aplicação emissora a alocação dos recursos necessários. Se todos os roteadores no caminho aceitarem a solicitação de reserva, a transmissão é realizada. Caso algum roteador não possa atender a solicitação, ele envia uma mensagem de erro à aplicação de destino e a transmissão do fluxo é cancelada.

Devido ao fato de possuir alta granularidade na alocação de recursos (a qual é feita para fluxos individuais), o modelo de Serviços Integrados acaba aumentando o grau de complexidade nos roteadores e dessa forma torna-se pouco escalável. Essa característica inviabiliza a sua utilização em grandes redes.

3.2.2 Serviços Diferenciados (DiffServ – *Differentiated Services*)

Para tentar superar as limitações do modelo IntServ, a IETF propôs, através da RFC 2475, um novo modelo chamado de Serviços Diferenciados (DiffServ – *Differentiated Services*). Neste modelo não há reserva de recursos e a qualidade de serviço é garantida através de mecanismos de priorização de pacotes.

Esse princípio de funcionamento torna o modelo DiffServ muito mais escalável que o IntServ, pois não existe a necessidade do armazenamento das informações de estado para cada fluxo individual de tráfego, diminuindo o nível de processamento exigido nos roteadores.

Na arquitetura DiffServ, os roteadores são chamados de nós DS (*Differentiated Service*) e uma rede formada por um conjunto de nós DS é chamada de domínio DS. Um domínio DS é formado por nós de borda, que interconectam o domínio DS com outro domínio (seja este DS ou não) e atuam como nós de entrada (*ingress*) e saída (*egress*) do domínio; e nós de centro, que conectam os nós no interior de um mesmo domínio DS. [POZ01].

A idéia básica do modelo DiffServ é que os vários fluxos de tráfego gerados pelas diversas aplicações sejam agregados a poucas classes de serviço através da classificação e marcação dos seus pacotes, de acordo com a qualidade de serviço especificada para cada

classe. [SAN09]. Assim, todo o tráfego que entra em um domínio DS é classificado segundo uma classe de serviço.

Para que isto seja possível, é utilizado o campo TOS (*Type of Service*) do cabeçalho do protocolo IPv4 e o campo *Traffic Class* do cabeçalho do protocolo IPv6. Na arquitetura DiffServ, esses campos são chamados de campos DS e contém as informações que irão determinar o tratamento dos pacotes nos nós DS. Com base nessas informações os nós DS podem identificar à qual classe de serviço pertencem os pacotes, evitando um tratamento individualizado. Essa característica é conhecida como comportamento agregado por nó (PHB – *Per Hop Behavior*).

O PHB é determinado pela codificação dos seis primeiros bits, chamados de DSCP (*Differentiated Service Code Point*), do campo DS.

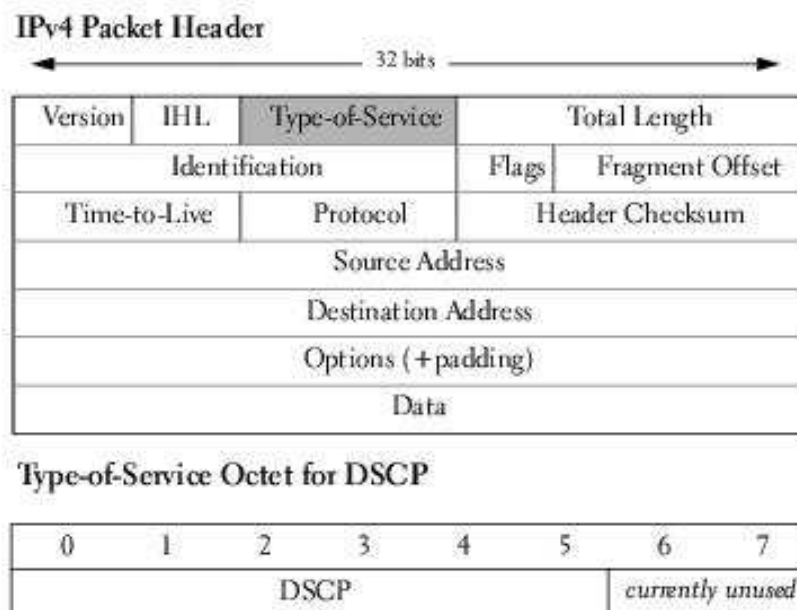


FIGURA 3.4: Campo TOS (*Type of Service*) do cabeçalho do protocolo IPv4. [DEL09].

Teoricamente, poderia haver até 64 (2^6) classes de serviço. Na prática, porém, a RFC 2475 recomenda as seguintes classes de serviço (ou PHBs):

- Comportamento Padrão (Default PHB): Corresponde ao comportamento do *best-effort* e recebe o tráfego que não foi classificado em nenhuma outra classe. Para esta classe, o valor recomendado para o campo DSCP é 000000;
- Encaminhamento Assegurado (AF - Assured Forwarding): Esta classe define quatro sub-classes de serviço com três níveis de descarte cada. Quando existir congestionamento em uma sub-classe, os pacotes são descartados conforme sua precedência de descarte. É melhor do que o *best-effort* pois comporta-se como se fosse uma rede *best-effort* pouco saturada;

TABELA 3.3: Valores do campo DSCP para classes do AF

Prioridade de descarte	Classe 1	Classe 2	Classe 3	Classe 4
Baixa	001010	010010	011010	100010
Média	001100	010100	011100	100100
Alta	001110	010110	011110	100110

Fonte: [SAN09].

- Encaminhamento Expresso (EF - Expedited Forwarding): Esta é a classe que possui a maior prioridade dentre as classes. Os pacotes classificados nesta classe transitam pela rede como se nenhum outro pacote estivesse presente, simulando uma linha dedicada e minimizando os atrasos, perda de pacotes e *jitter*. O valor recomendado para o campo DSCP é 101110.

A figura 3.5 ilustra os principais blocos funcionais presentes nos equipamentos que implementam a solução DiffServ, seguida de uma breve explicação da função de cada bloco:

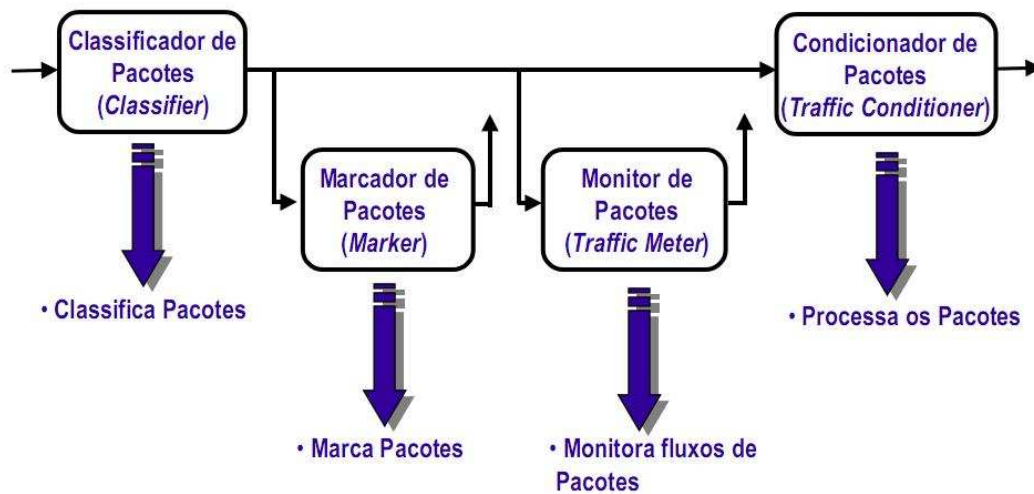


FIGURA 3.5: Blocos funcionais do modelo DiffServ. [SAN09].

- Classificador de Pacotes: responsável pela associação dos pacotes às suas classes de tráfego. Essa classificação pode ser baseada nos campos de endereço de origem e destino, portas de origem e destino, código do protocolo, além do valor do campo DSCP;
- Marcador de Pacotes: escreve o campo DSCP do pacote com o valor apropriado, baseado na classificação feita pelo Classificador de Pacotes;
- Monitor de Pacotes: responsável pela medição do tráfego real e pela comparação com o perfil de tráfego destinado à classe;
- Condicionador de Pacotes: faz o policiamento do tráfego, retardando ou até mesmo descartando os pacotes que não estão de acordo com o perfil da classe à qual pertencem.

3.2.3 IEEE 802.1p

Quando o padrão Ethernet (IEEE 802.3) tornou-se a tecnologia mais usada para redes locais, foi necessário incorporar à camada dois (camada de enlace) do modelo OSI mecanismos para garantir a qualidade de serviço. Foi então que o IEEE, através do grupo de trabalho 802.1, desenvolveu o padrão IEEE 802.1p (a letra “p” vem da palavra *priorization*): protocolo para priorização de tráfego na camada de enlace.

Como originalmente o quadro Ethernet não possui um campo de priorização, o padrão IEEE 802.1p requer outro padrão, o IEEE 802.1q. O IEEE 802.1q é o padrão para VLANs (*Virtual Local Area Networks*), as redes locais virtuais. Ele insere quatro *bytes* (chamados de *VLAN Tag*) ao quadro Ethernet, conforme ilustrado na figura 3.6:

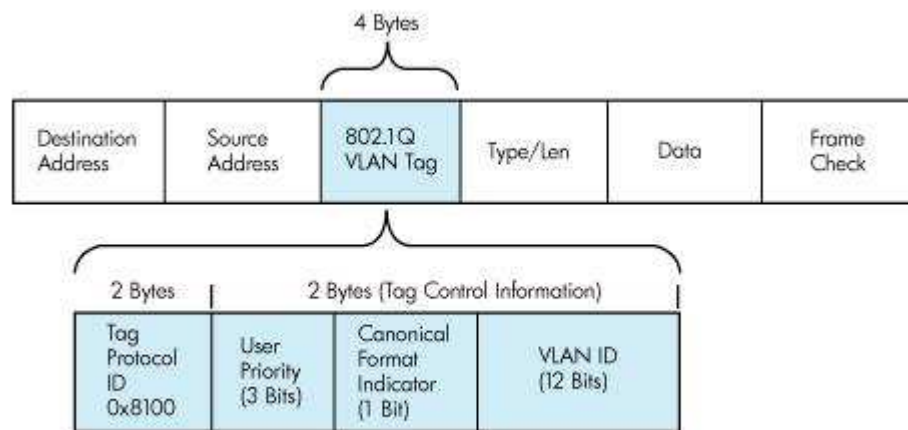


FIGURA 3.6: Quadro Ethernet modificado pelo padrão IEEE 802.1q. [HPT09].

Abaixo segue a descrição de cada campo da *VLAN Tag*:

- ***Tag Protocol Identifier (TPID)***: esse campo possui o valor padrão de 0x8100, que significa que o quadro Ethernet está nos padrões IEEE 802.1q/802.1p;
- ***User Priority***: campo utilizado pelo padrão IEEE 802.1p para definir a prioridade do quadro;
- ***Canonical Format Indicator (CFI)***: utilizado para fins de compatibilidade com redes *Token Ring*. Em redes Ethernet, possui sempre o valor zero;
- ***VLAN ID (VID)***: campo utilizado pelo padrão IEEE 802.1q para identificação de qual VLAN o quadro pertence.

É importante salientar que, em seu formato original, o quadro Ethernet possui o tamanho máximo de 1.518 bytes. Com a inclusão ao quadro dos quatro bytes necessários para implementar o padrão IEEE 802.1q, dispositivos que não implementam ou que não estejam

configurados para utilizar o padrão IEEE 802.1q não reconhecerão o quadro e irão descartá-lo.

O padrão IEEE 802.1p utiliza os três *bits* do campo *User Priority* (o qual não é usado pelo IEEE 802.1q) para agrupar os quadros em oito (2^3) diferentes classes de tráfego. Essas classes são usadas para que os dispositivos de rede possam reconhecer e tratar de maneira diferenciada o tráfego de cada uma delas, dando-lhes uma prioridade maior ou menor. Embora a forma com que o tráfego de cada classe deva ser tratado fique a critério do administrador da rede, o IEEE recomenda que se utilize a seguinte configuração:

TABELA 3.4: Valores do campo *User Priority* recomendados pelo IEEE

<i>User Priority</i>	Tipo de Tráfego
7 (maior prioridade)	<i>Network Management</i>
6	Voz
5	Vídeo
4	Carga Controlada
3	<i>Excellent-Effort</i>
0 (tráfego de rotina)	<i>Best-Effort</i>
2	Não definido
1 (menor prioridade)	<i>Background</i>

Fonte: [HPT09a].

Todos os modelos de implementação de QoS, tanto na camada de rede (IntServ e DiffServ), como na camada de enlace (IEEE 802.1p), são implementados nos dispositivos de rede através da utilização de diversos mecanismos [SAN09]. Abaixo constam alguns deles, seguidos de exemplos:

- Protocolos de sinalização
 - RSVP – Resource Reservation Protocol: protocolo especialmente desenvolvido para o modelo IntServ através da RFC 2205, permite que as próprias aplicações requeiram da rede reservas de recursos necessários para seus fluxos;
- Algoritmos de prioridade
 - Priority Queuing: esquema formado por filas distintas para diferentes classes de tráfego, onde a transmissão tem início pelo tráfego de maior prioridade e é realizada de forma exaustiva, podendo causar *starvation* nas filas de menor prioridade;
- Algoritmos de escalonamento
 - CBQ – Class Based Queuing: esse mecanismo é caracterizado pela existência de uma fila para cada tipo de tráfego, as quais são servidas de forma cíclica, onde especifica-se o percentual de banda do canal ou ainda o número de *bytes* a ser transmitido a cada ciclo;
- Algoritmos de controle de filas
 - SFQ – Stochastic Fair Queuing: nesse mecanismo as filas são servidas um pacote de cada vez, utilizando o *Round-Robin* como estratégia de escalonamento;
- Algoritmos de congestionamento
 - RED – Random Early Detection: o principal objetivo deste algoritmo é limitar o tamanho das filas, controlando o atraso médio introduzido na transmissão de pacotes.

4 QUALIDADE DE SERVIÇO EM AMBIENTES *MICROSOFT WINDOWS XP E WINDOWS SERVER 2003*

Em decorrência da especificidade do assunto, o conteúdo deste capítulo está baseado quase que totalmente nas informações contidas em [TEC09].

A *Microsoft* introduziu mudanças no suporte à QoS dos sistemas operacionais *Windows XP* e *Windows Server 2003* em relação ao seu antecessor, o *Windows 2000*. O *Windows 2000* suporta tanto o modelo IntServ quanto o modelo DiffServ e inclui mecanismos de reserva de recursos, controle de admissão e controle de tráfego. Já o *Windows XP* e o *Windows Server 2003* suportam somente o modelo DiffServ e mecanismos de controle de tráfego. O principal motivo para essas mudanças foi o fato, já mencionado anteriormente, que o modelo IntServ é complexo e pouco escalável.

4.1 Arquitetura de QoS

4.1.1 Controle de tráfego

No *Windows XP* e no *Windows Server 2003* a qualidade de serviço é implementada através de mecanismos de controle de tráfego, como priorização e *traffic shaping* (conformação de tráfego), os quais permitem a separação do tráfego em classes de serviço e regulam a forma como os pacotes são enviados pela rede. As etapas envolvidas no processo de controle de tráfego são as seguintes:

- Classificação;
- Conformação;

- Escalonamento;
- Marcação.

4.1.1.1 Classificação

Nessa etapa os pacotes são classificados em uma determinada classe de serviço e encaminhados para sua respectiva fila de saída. No *Windows XP* e no *Windows Server 2003* existem as seguintes classes de serviço:

- Melhor esforço (*Best-effort*): classe de serviço padrão que não oferece garantias de desempenho;
- Carga controlada (*Controlled load*): essa classe de serviço oferece um tratamento similar ao da classe de melhor esforço, porém simulando uma rede des congestionada. Esse comportamento proporciona uma baixa taxa de perda de pacotes e, em geral, baixos atrasos na entrega (mas não são oferecidas garantias em relação aos atrasos);
- Serviço garantido (*Guaranteed service*): classe que provê uma alta taxa de entregas bem sucedidas com garantia de um atraso mínimo. Devido a essas características, essa classe tem um impacto significativo na rede e deve ser usada exclusivamente para aplicações críticas;
- Controle de rede (*Network control*): classe com a mais alta prioridade em relação às demais. Usada para o tráfego das aplicações de gerenciamento da rede;
- Serviço qualitativo (*Qualitative service*): classe destinada a aplicações que exigem um tratamento priorizado para seu tráfego, mas que ao invés de expressarem de forma quantificada suas necessidades, expressam de forma qualitativa (ex.: “baixo atraso”). Neste caso, os próprios elementos da rede é que irão determinar como os pacotes deverão ser tratados.

4.1.1.2 Conformação

Durante esta etapa, o tráfego de saída é moldado através da suavização das rajadas de pacotes, com o objetivo de criar um fluxo mais uniforme.

4.1.1.3 Escalonamento

Nesta etapa, o *QoS Packet Scheduler* atua no controle da prioridade de acesso à rede de cada fila de saída.

4.1.1.4 Marcação

Na marcação, é explicitamente definida a prioridade de cada pacote. Para isso, é escrito um código de identificação de classe no cabeçalho dos pacotes. Essa marcação irá mapear o pacote em uma determinada classe de serviço e pode ser feita tanto na camada de enlace (camada 2 do modelo OSI), através da utilização do IEEE 802.1p, quanto na camada de rede (camada 3 do modelo OSI), através da utilização do modelo DiffServ.

4.1.2 Componentes

Os componentes da arquitetura de qualidade de serviço do *Windows XP* e do *Windows Server 2003* são o *Generic Packet Classifier (GPC)*, o *QoS Packet Scheduler*, e as APIs (*Application Programming Interfaces*) *Generic Quality of Service (GQoS)* e *Traffic Control (TC)*.

Na figura 4.1, pode ser visto o posicionamento dos componentes (as APIs GQoS e TC fazem parte da API *Windows Sockets*) dentro do modelo arquitetural do TCP/IP implementado no *Windows XP* e no *Windows Server 2003*:

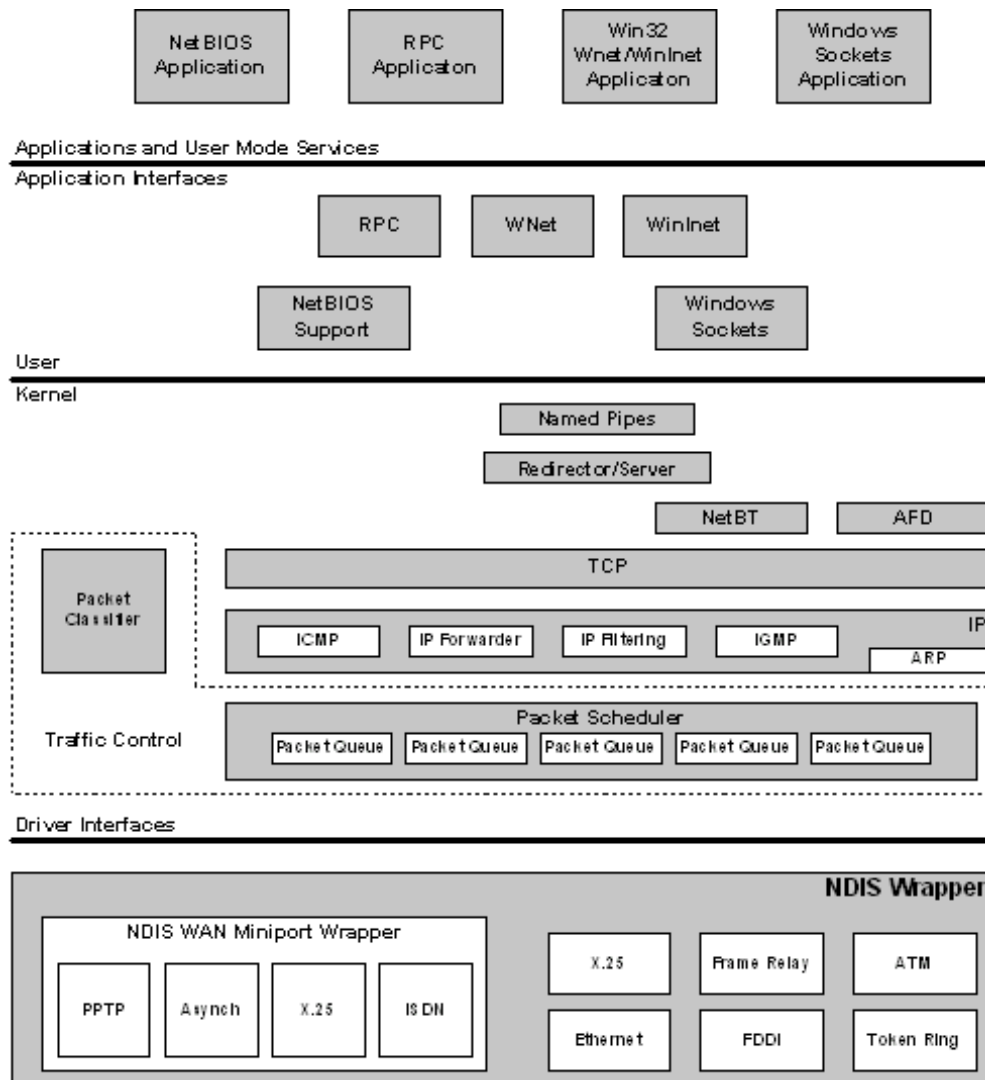


FIGURA 4.1: Modelo arquitetural do TCP/IP implementado no *Windows XP* e *Windows Server 2003*. [TCP09].

4.1.2.1 Generic Quality of Service (GQoS)

No *Windows 2000*, esta API prove acesso aos mecanismos de reserva de recursos e de controle de tráfego. Já no *Windows XP* e no *Windows Server 2003*, ela suporta somente o acesso aos mecanismos de controle de tráfego.

Os desenvolvedores podem usar a API GQoS para desenvolver aplicações *QoS-aware*, ou seja, aplicações que são capazes de aplicar métricas de qualidade de serviço ao seu tráfego.

A API GQoS faz parte da API *Windows Sockets 2.0* (Winsock2).

4.1.2.2 Traffic Control (TC)

A API *Traffic Control* (traffic.dll) é utilizada para prover acesso aos mecanismos de controle de tráfego no nível de *host*, diferentemente da API GQoS, que é utilizada no nível de aplicação. Por ser uma API de mais baixo nível, a TC requer privilégios administrativos.

Aplicações que fazem uso da GQoS, ou seja, aplicações *QoS-aware*, utilizam indiretamente a API TC. Para aplicações que não sejam *QoS-aware*, os administradores de rede podem usar ferramentas de gerência de rede que invocam funções e métodos da API *Traffic Control* para aplicar as métricas de qualidade de serviço.

Ao contrário da GQoS, que considera como sendo um fluxo o tráfego definido por endereço IP de origem e destino, porta de origem e destino e protocolo (TCP/UDP), a API *Traffic Control* permite que o tráfego de saída das várias aplicações de um *host* sejam agregados em um único fluxo, como por exemplo, todo o tráfego tendo como destino um determinado endereço IP.

Para classificar o tráfego, a *Traffic Control* utiliza as seguintes especificações:

- *Flowspec*: lista contendo as métricas de QoS (classe de serviço, taxa de transferência, etc.) que devem ser aplicadas a um fluxo;
- *Filterspec*: lista dos atributos (IP e porta de origem, IP e porta de destino, etc.) que definem um fluxo.

4.1.2.3 Generic Packet Classifier (GPC)

O *Generic Packet Classifier* (msgpc.sys) atua classificando os pacotes em classes de serviço (definidas pelo *flowspec*) e mapeando-os para as suas respectivas filas de saída.

4.1.2.4 QoS Packet Scheduler

O *QoS Packet Scheduler* (psched.sys) é o componente do kernel que controla e aplica as métricas de QoS nas filas de saída. Para escalonar o acesso das filas à rede, ele baseia-se na prioridade e na taxa de transferência de cada fluxo. A prioridade é usada durante períodos de congestionamento para determinar a ordem de entrega dos pacotes à rede e a taxa de

transferência é usada para definir o ritmo da transmissão dos pacotes, evitando assim a premissa “enviar tudo agora”, padrão das transmissões IP.

Para garantir o tratamento preferencial ao longo da rede, o *QoS Packet Scheduler* utiliza o padrão IEEE 802.1p. Para isso, ele informa ao *driver* da placa de rede e este escreve no campo *User Priority* dos pacotes de cada fila o valor correspondente da classe de serviço à qual eles pertencem, permitindo assim que os dispositivos de rede que atuam na camada de enlace possam tratar os pacotes adequadamente. O *QoS Packet Scheduler* garante a qualidade de serviço somente na camada de enlace. Na camada de rede a qualidade de serviço é garantida através do modelo DiffServ e quem realiza a escrita do valor correspondente da classe no campo DSCP é o próprio protocolo TCP/IP.

Nas tabelas 4.1 e 4.2 podem ser vistos os valores padrão do *Windows XP* e do *Windows Server 2003* para as classes de serviço dos modelos IEEE 802.1p e DiffServ, respectivamente.

TABELA 4.1: Valores padrão do *Windows* para as classes de serviço do IEEE 802.1p

Classe de serviço	Valor
Pacotes sem <i>flowspec</i>	0
<i>Best-effort</i>	0
<i>Controlled load</i>	4
<i>Guaranteed service</i>	5
<i>Network control</i>	7
<i>Qualitative</i>	0

Fonte: [QOS09a].

TABELA 4.2: Valores padrão do *Windows* para as classes de serviço do DiffServ

Classe de serviço	Valor
Pacotes sem <i>flowspec</i>	0

<i>Best-effort</i>	0
<i>Controlled load</i>	24
<i>Guaranteed service</i>	40
<i>Network control</i>	48
<i>Qualitative</i>	0

Fonte: [QOS09a].

Tanto os valores padrão do IEEE 802.1p como os do DiffServ podem ser modificados através da edição das *Group Policies* (Políticas de Grupo) do *host*. Para isso, deve-se acessar a ferramenta GPedit (*Group Policy Editor*) com o comando *gpedit.msc* e editar as configurações que estão em “*Computer Configuration\Administrative Templates\Network\QoS Packet Scheduler*”.

Além da configuração dos valores que representam as classes de serviço, outras configurações associadas à qualidade de serviço podem ser alteradas através da edição das *Group Policies*. Na tabela 4.3 podemos ver alguns exemplos:

TABELA 4.3: Configurações das *Group Policies* associadas ao *QoS Packet Scheduler*

Configuração	Descrição
<i>Limit outstanding</i>	Define o número máximo de pacotes pendentes permitido. Quando esse limite é atingido, o <i>QoS Packet Scheduler</i> adia o envio dos pacotes para os adaptadores de rede até que o número de pacotes pendentes reduza abaixo deste valor.
<i>Limit reservable bandwidth</i>	Especifica a porcentagem máxima de largura de banda da conexão que as aplicações <i>QoS-aware</i> podem reservar no <i>host</i> . Esta reserva não é garantida pois as aplicações que não são <i>QoS-aware</i> podem consumir uma banda maior que a remanescente.
<i>Set timer resolution</i>	Determina a menor unidade de tempo (em microsegundos) que o <i>QoS Packet</i>

	<i>Scheduler</i> usa ao escalonar pacotes para transmissão.
--	---

Fonte: [QOS09a].

5 TESTES DE IMPLEMENTAÇÃO

Para testar o funcionamento da implementação da qualidade de serviço nas plataformas *Microsoft Windows XP/2003*, foi montado um ambiente de teste no qual duas estações foram conectadas através de um cabo UTP (*Unshielded Twisted Pair*) do tipo *crossover* e onde uma delas atuava como geradora do tráfego sob o qual foram aplicadas as políticas de QoS. Todas as medições e coletas de dados foram realizadas na interface de rede da estação geradora do tráfego.

5.1 Ferramentas utilizadas

Para a realização dos testes foram necessárias as seguintes ferramentas:

- *Traffic Control Monitor (TCMon)*: ferramenta que integra o *Windows Server 2003 Resource Kit Tools* e é utilizada para interagir com a arquitetura de QoS do *Windows* através da API *Traffic Control*;
- *Iperf (versão: 1.7.0)*: ferramenta de linha de comando utilizada para gerar tráfego e para medir a banda e a qualidade de um link;
- *Windows Performance Monitor (Perfmon)*: ferramenta que integra os sistemas *Windows* e é usada para coletar dados de performance;
- *Wireshark (versão: 1.2.2)*: ferramenta de análise de protocolos de rede.

5.2 Configuração dos equipamentos

As estações envolvidas nos testes possuíam as seguintes configurações:

TABELA 5.1: Configuração dos equipamentos

	Estação Transmissora	Estação Receptora
Marca:	<i>HP Compaq</i>	<i>HP Compaq</i>
Modelo:	<i>dc5000 SFF</i>	<i>dc5000 SFF</i>
Processador:	<i>Pentium 4 - 2.80GHz</i>	<i>Pentium 4 - 2.80GHz</i>
Memória:	<i>512 MB</i>	<i>512 MB</i>
Placa de rede:	<i>Broadcom NetXtreme Gigabit Ethernet for hp</i>	<i>Broadcom NetXtreme Gigabit Ethernet for hp</i>
Driver da placa de rede	<i>Versão 6.34.0.0</i>	<i>Versão 6.34.0.0</i>
Sistema operacional:	<i>Windows XP SP3</i>	<i>Windows XP SP3</i>

As estações foram conectadas a 1 Gbps através de um cabo UTP categoria 5e de 10 metros do tipo *crossover*.

5.3 Pré-requisitos

Para que os testes pudessem ser realizados foram necessárias a instalação e configuração das ferramentas, conforme descrito a seguir:

- Traffic Control Monitor (TCMon)
 - *Download* do *Windows Server 2003 Resource Kit Tools* do site <http://www.microsoft.com/downloads/details.aspx?familyid=9d467a69-57ff-4ae7-96ee-b18c4790cffd&displaylang=en> e instalação na estação transmissora;
 - Execução do arquivo “Tcmon.bat”, que se encontra na pasta de instalação do *Windows Server 2003 Resource Kit Tools*, para instalação dos componentes necessários para a execução do TCMon;

- Iperf
 - *Download* do arquivo “iperf.exe” do site <http://www.noc.ucf.edu/Tools/Iperf/> para as estações transmissora e receptora;

- Wireshark
 - *Download* do site <http://www.wireshark.org/download.html> e instalação na estação transmissora.

Como o sistema operacional utilizado durante os testes foi o *Windows XP*, não houve a necessidade de instalar o *QoS Packet Scheduler* (psched.sys), pois o mesmo já vem instalado por padrão. Nos sistemas operacionais *Windows Server 2003* este componente deve ser instalado manualmente.

Para que o *QoS Packet Scheduler* pudesse escrever no campo *User Priority*, foi necessário se certificar de que o *driver* da placa de rede da estação transmissora suportava o padrão IEEE 802.1p e que essa funcionalidade estava habilitada. Isso pôde ser verificado nas propriedades avançadas do *driver* da placa de rede, bem como na aba *Parameters* do TCMon.

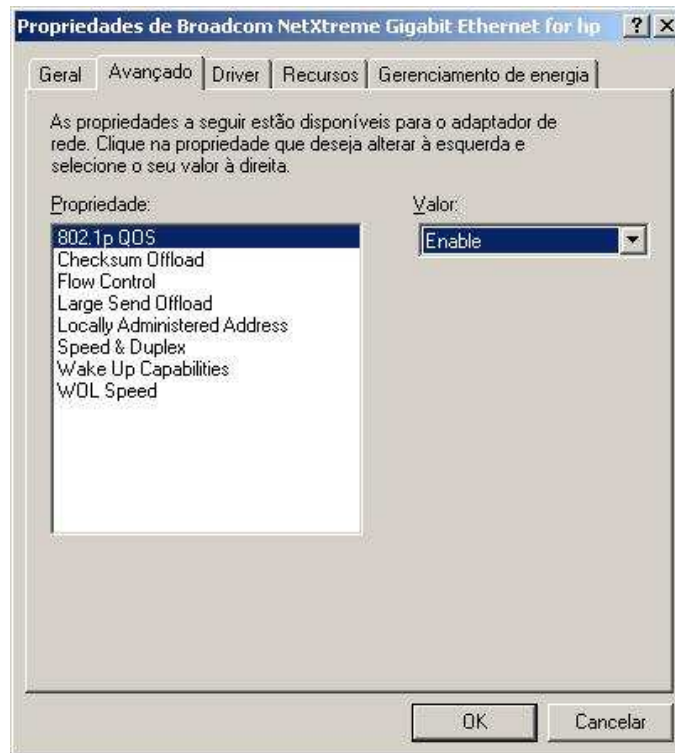


FIGURA 5.1: Propriedades avançadas do *driver* da placa de rede da estação transmissora.

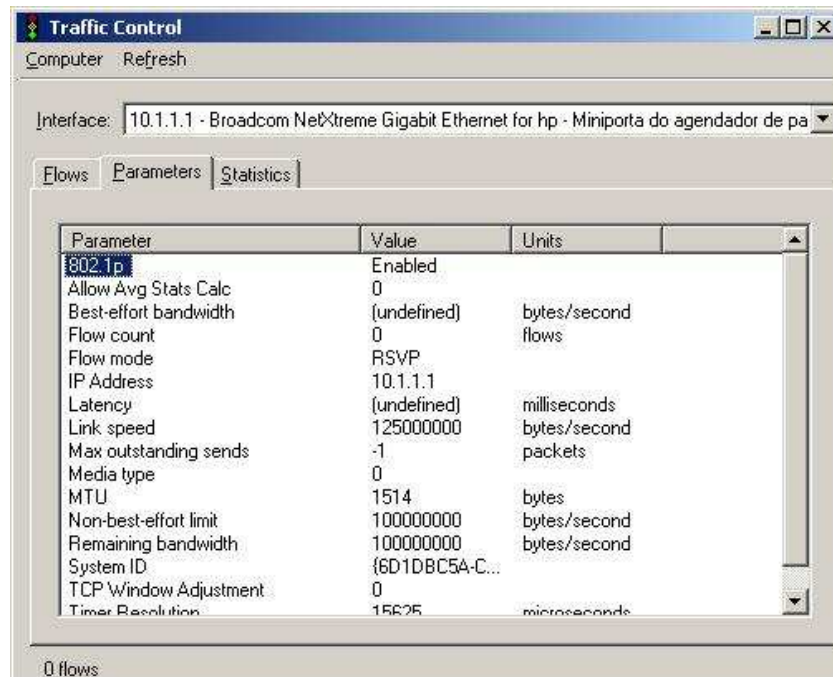


FIGURA 5.2: Aba *Parameters* do TCMon instalado na estação transmissora.

5.4 Testes

Os testes simulam os casos de tráfegos gerados por aplicações não *QoS-aware*, ou seja, aplicações que não são capazes de aplicar métricas de qualidade de serviço ao seu tráfego. Para gerar tais tráfegos foi utilizada a ferramenta Iperf atuando como cliente na máquina transmissora (endereço IP 10.1.1.1/24) e como servidor na máquina receptora (endereço IP 10.1.1.2/24). Para facilitar, foi criado um arquivo em lote (cliente.bat) que continha os seguintes comandos do Iperf (os quais geram tráfego para as portas UDP 5001, UDP 5002 e UDP 5003 da máquina receptora):

- *iperf.exe -c 10.1.1.2 -u -i 1 -p 5001 -f m -b 300.0M -n 300.0M*
- *iperf.exe -c 10.1.1.2 -u -i 1 -p 5002 -f m -b 300.0M -n 300.0M*
- *iperf.exe -c 10.1.1.2 -u -i 1 -p 5003 -f m -b 300.0M -n 300.0M*

onde:

- **-c 10.1.1.2** especifica que o Iperf está atuando como cliente e irá enviar o tráfego para o servidor 10.1.1.2;
- **-u** especifica que o tráfego será do tipo UDP;
- **-i 1** especifica que os relatórios serão fornecidos a cada um segundo;
- **-p 5001** especifica a porta de comunicação do servidor;
- **-f m** especifica que o relatório mostrará os dados em *megabits*;
- **-b 300.0M** especifica a taxa de transferência máxima do fluxo (300 Mbit/sec);
- **-n 300.0M** especifica a quantidade de dados que serão transferidos (300 MB).

Na máquina receptora (servidor), foi criado o arquivo em lote (servidor.bat) contendo os seguintes comandos:

- *iperf.exe -s -u -i 1 -p 5001 -f m*

- `iperf.exe -s -u -i 1 -p 5002 -f m`
- `iperf.exe -s -u -i 1 -p 5003 -f m`

onde:

- `-s` especifica que o Iperf está atuando como servidor.

A figura 5.3 ilustra as atuações das máquinas transmissora e receptora:

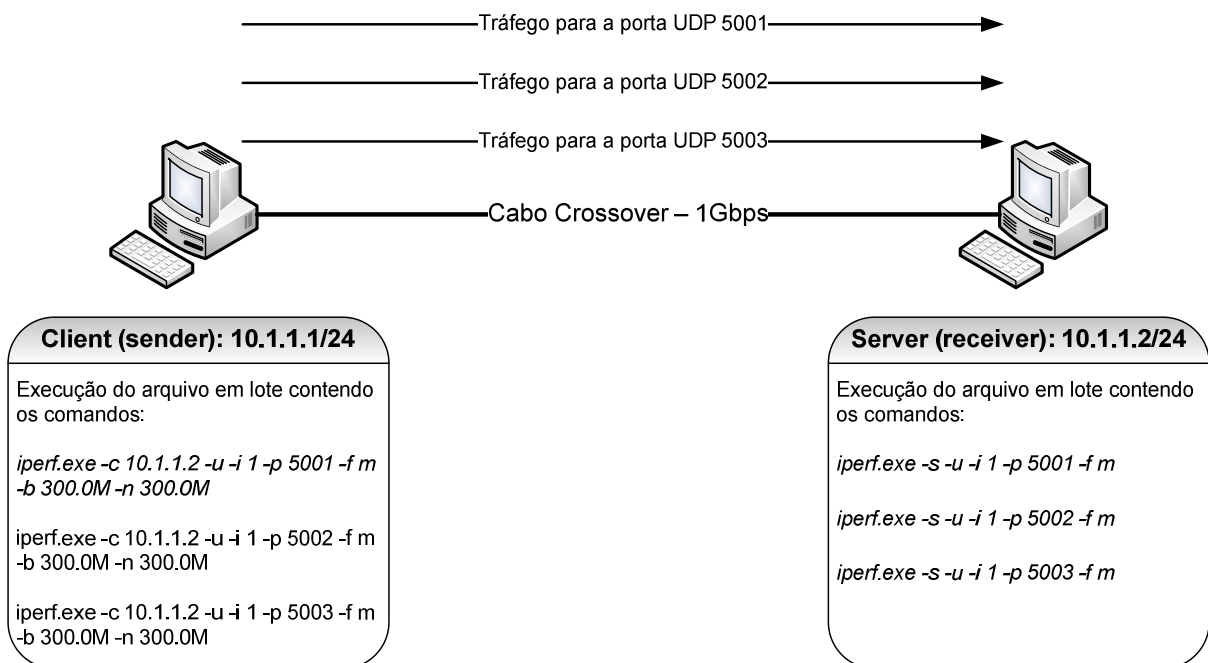


FIGURA 5.3: Atuação das estações durante os testes.

Com a utilização da ferramenta TCMon foram criados três fluxos da máquina transmissora para a receptora (portas UDP 5001, UDP 5002 e UDP 5003). Inicialmente foram setadas as mesmas métricas para os três fluxos.

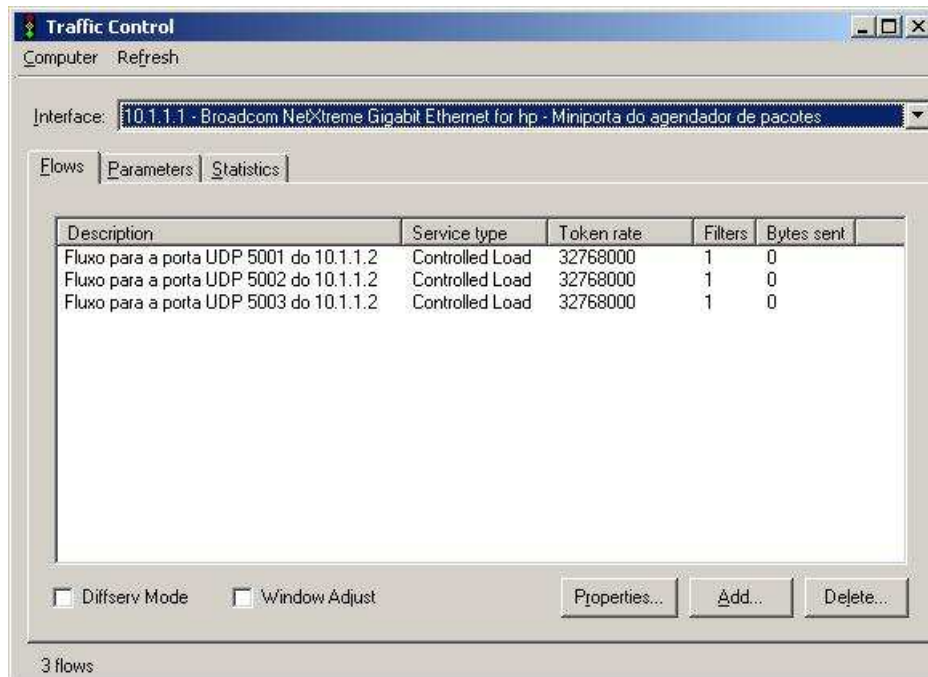


FIGURA 5.4: Fluxos criados no TCMon.

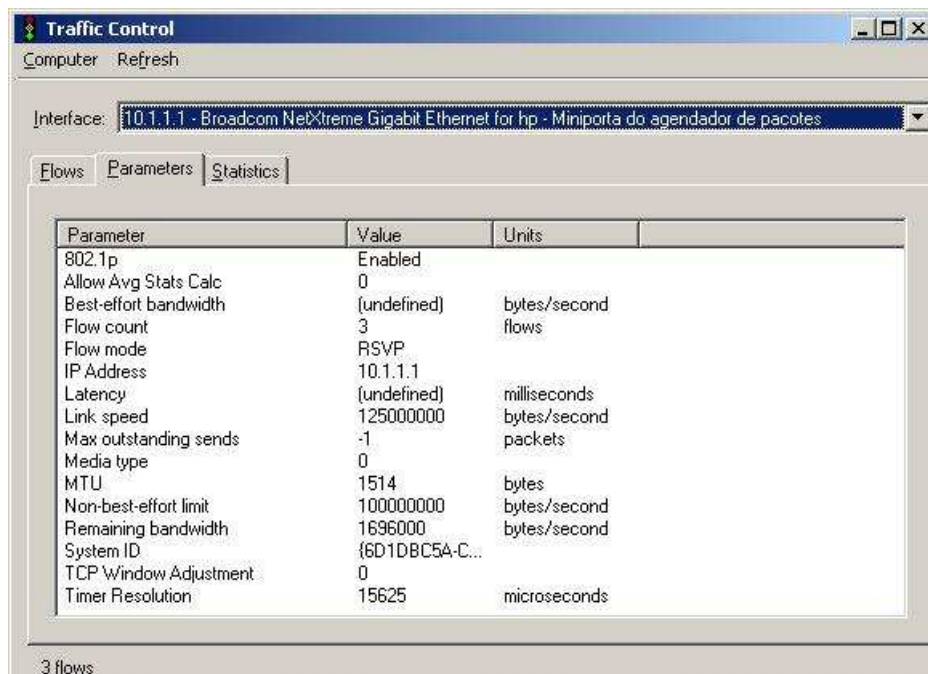


FIGURA 5.5: Parâmetros da interface de rede.

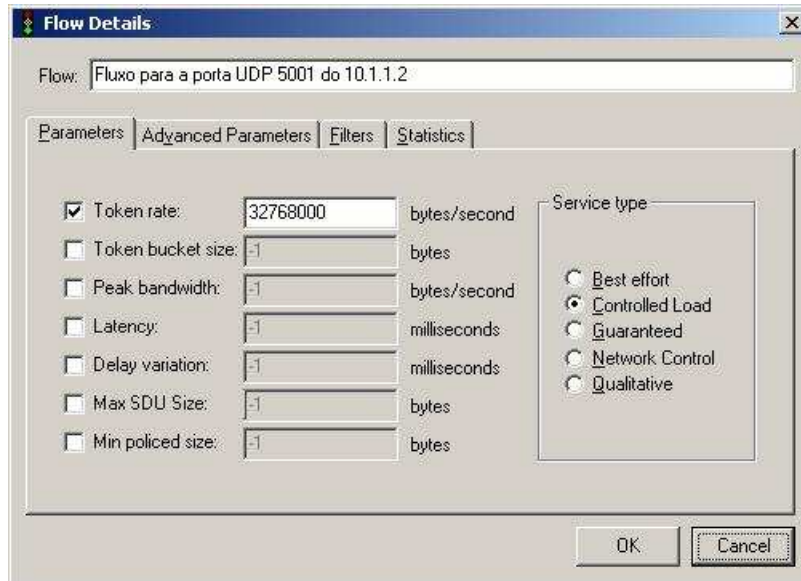


FIGURA 5.6: Métricas do fluxo para a porta 5001 (mesmas métricas para os outros fluxos).

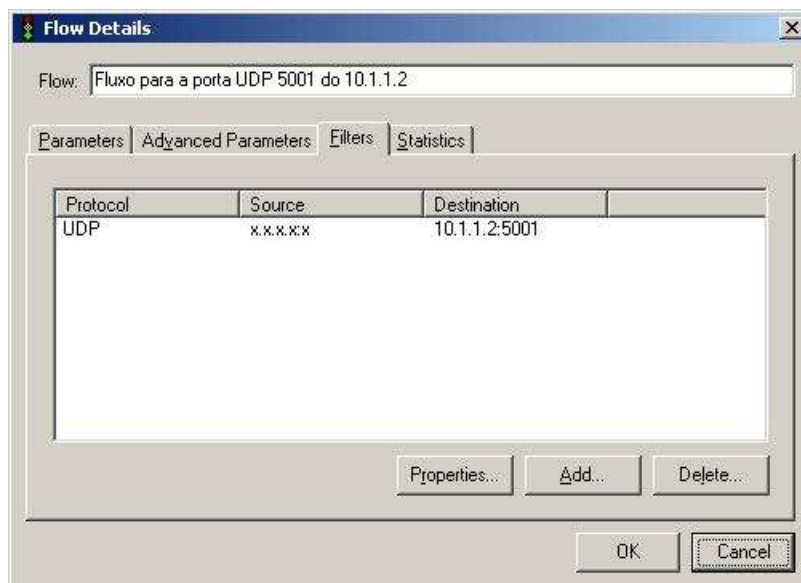


FIGURA 5.7: Filtro do fluxo para a porta 5001 (nos outros fluxos muda a porta de destino).

Para a visualização gráfica dos tráfegos, foi adicionado na ferramenta *Windows Performance Monitor* (Perfmon) da máquina transmissora o contador “Bytes transmitidos/s” (pertencente ao objeto de desempenho “Fluxo PSched”) para os três fluxos (instâncias), conforme figura 5.8:

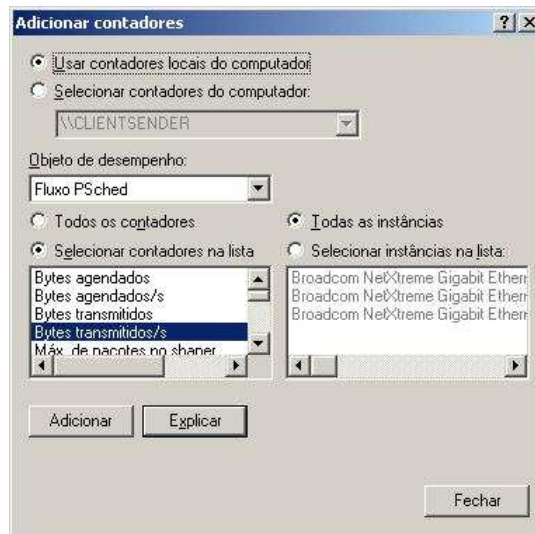


FIGURA 5.8: Adição de contadores no Perfmon.

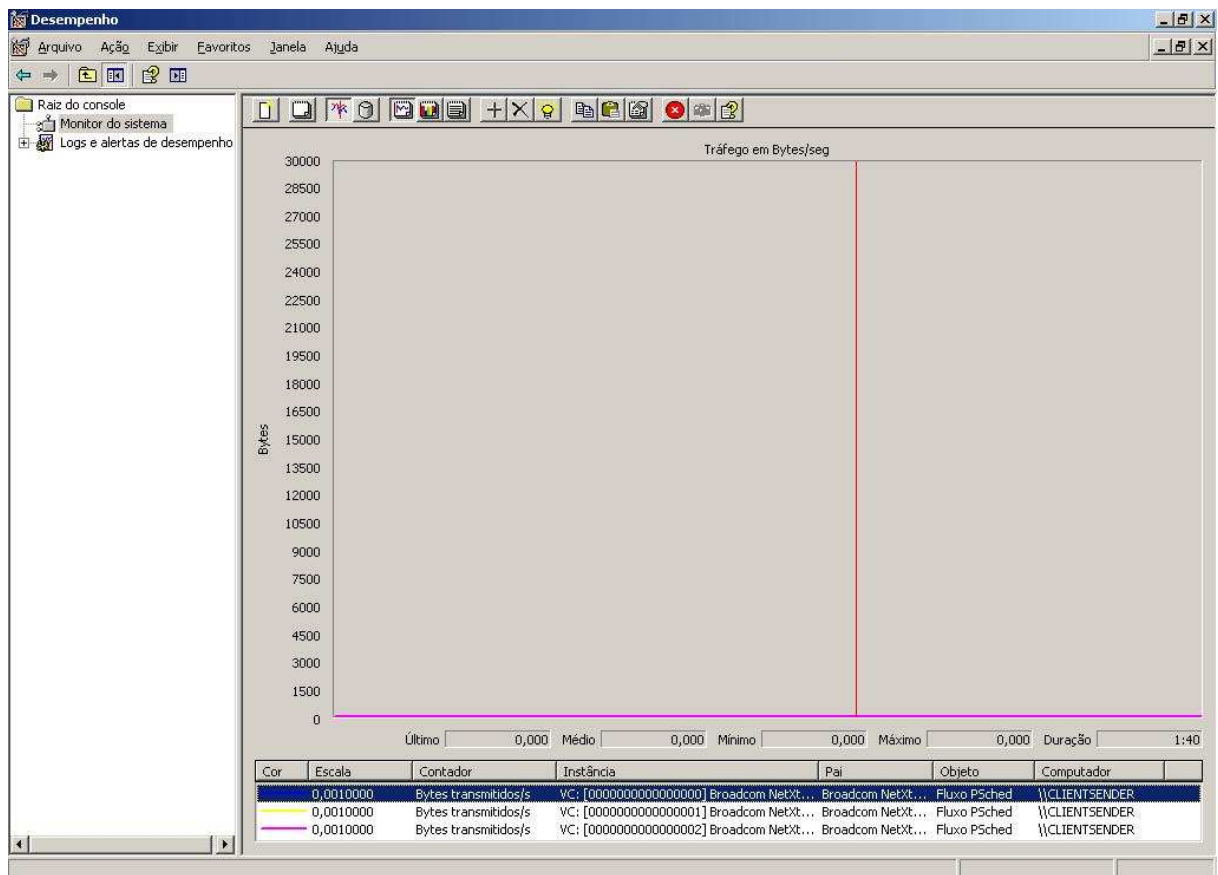


FIGURA 5.9: Perfmon após a adição do contador “Bytes transmitidos/s”.

Para a visualização do conteúdo dos pacotes, foi utilizada ferramenta *Wireshark* na máquina transmissora configurada com o filtro “*udp.dstport == 5001 || udp.dstport == 5002 || udp.dstport == 5003*”, para que os pacotes mostrados fossem somente os que tivessem as portas UDP 5001, 5002 e 5003 como destino.

Após a configuração do TCMon e do *Wireshark* na máquina transmissora, foram executados os arquivos em lote “*servidor.bat*” na máquina receptora e “*cliente.bat*” na máquina transmissora, obtendo os seguintes resultados:

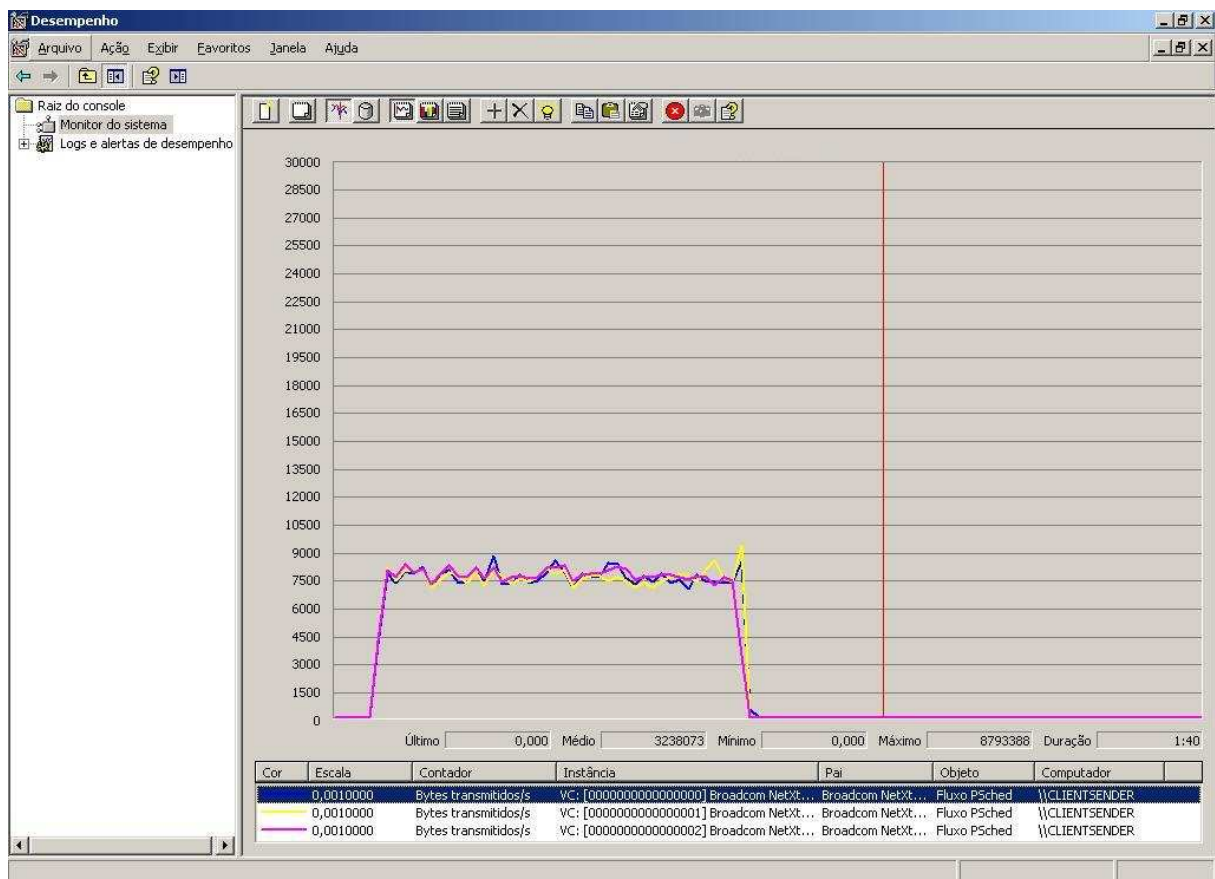


FIGURA 5.10: Perfmon mostrando a taxa de transferência em KB/s dos três fluxos.

Pode-se observar que os três fluxos concorreram de forma igualitária pela prioridade na transmissão. Todos eles tiveram uma taxa de transferência média de 61 Mb/s e tempo de transmissão de 41 segundos.

Com o *Wireshark* foi possível capturar os pacotes e verificar seu conteúdo. Na figura 5.11 podemos visualizar que o campo DSCP está preenchido com o valor 0x18, que corresponde à classe *Controlled load*, exatamente como configurado no TCMon.

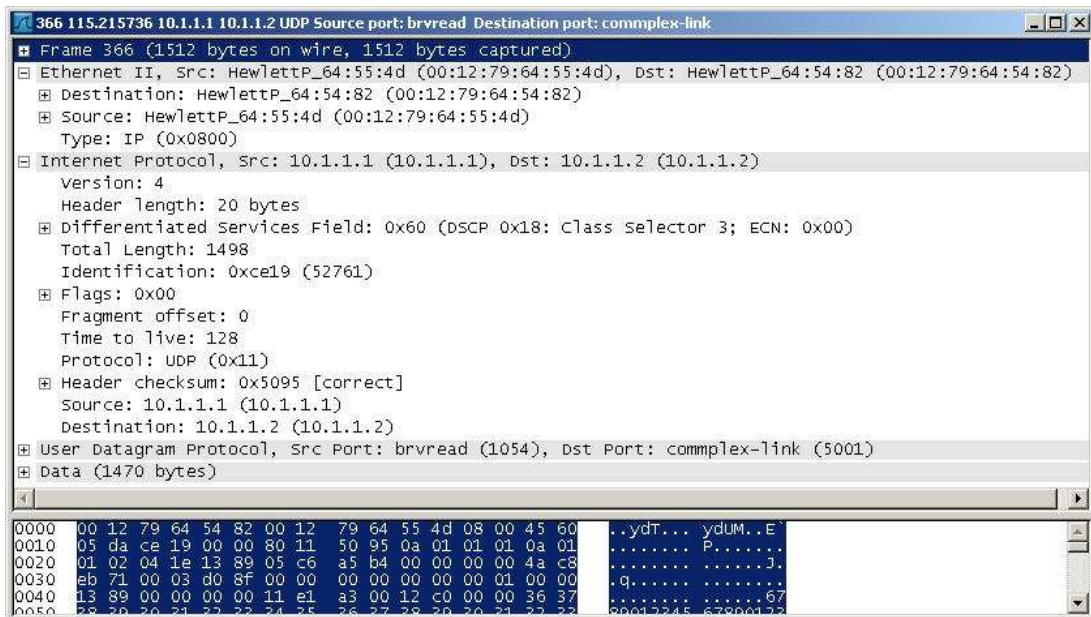


FIGURA 5.11: Conteúdo do pacote capturado com o *Wireshark*.

A visualização do campo *User Priority* não foi possível pois este campo é inserido pelo *driver* da placa de rede, e aplicativos de análise de protocolos de rede como o *Wireshark* se inserem acima do nível do *driver*, ou seja, antes da inclusão do campo.

No segundo teste as métricas foram configuradas de forma diferente para os três fluxos. No fluxo para a porta UDP 5001 foi configurada uma taxa de transferência máxima de 300 Mb/s e classe de serviço *Network Control*. Para o fluxo da porta UDP 5002 foi estabelecida uma taxa máxima de 225 Mb/s e classe de serviço *Controlled load*. Já para o fluxo da porta UDP 5003 configurou-se uma taxa de transferência máxima de 150 Mb/s e classe de serviço *Best-effort*.

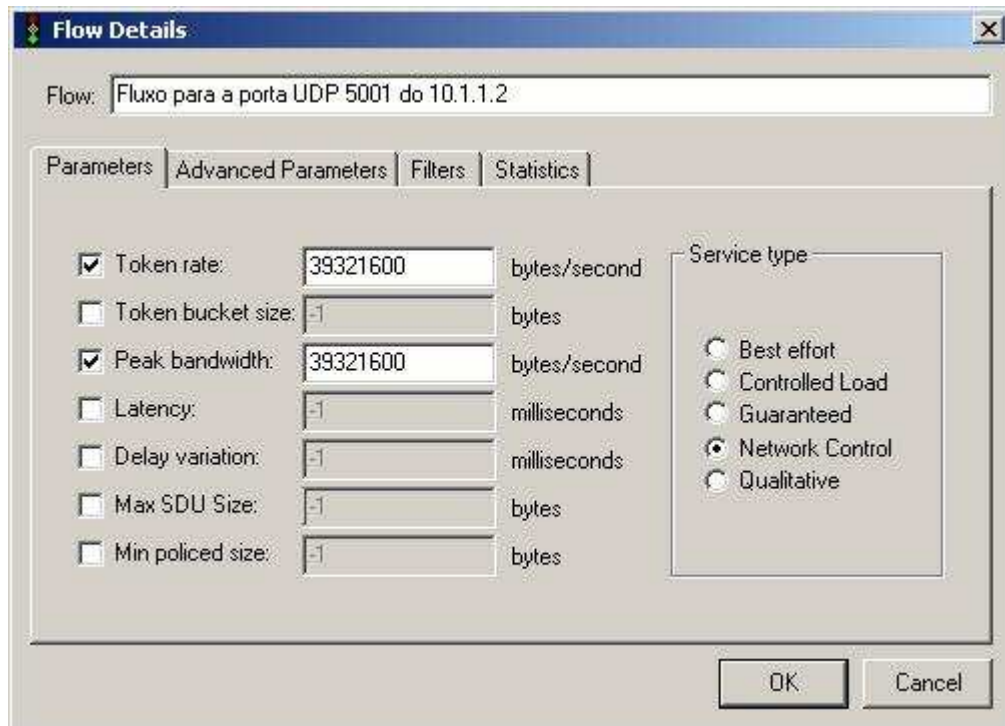


FIGURA 5.12: Métricas do fluxo para a porta UDP 5001.

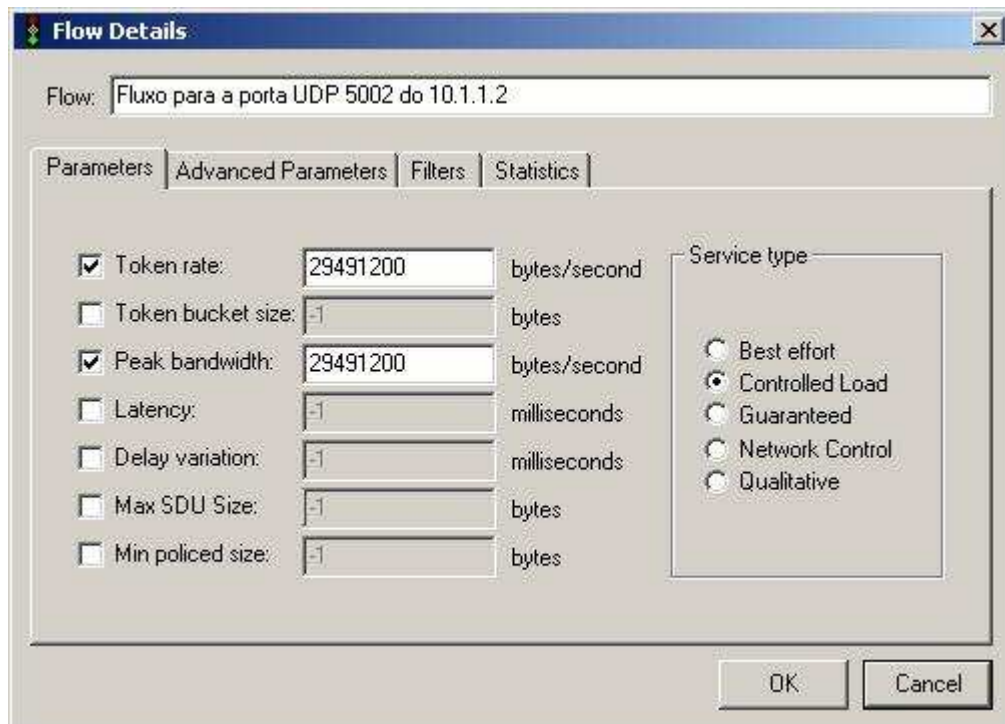


FIGURA 5.13: Métricas do fluxo para a porta UDP 5002.

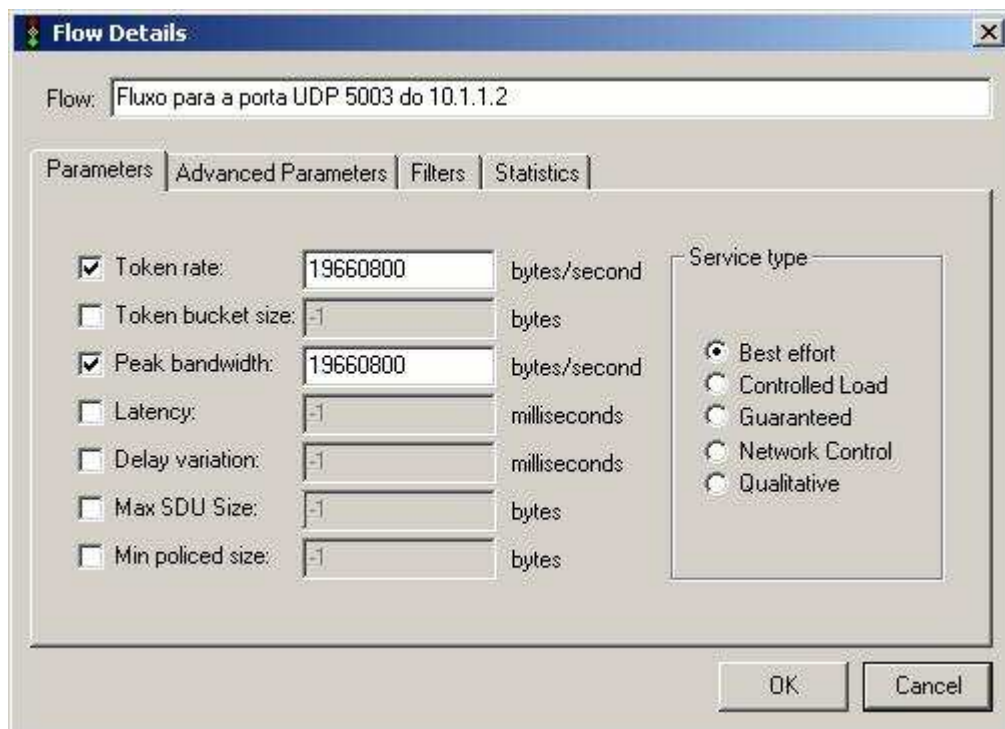


FIGURA 5.14: Métricas do fluxo para a porta UDP 5003.

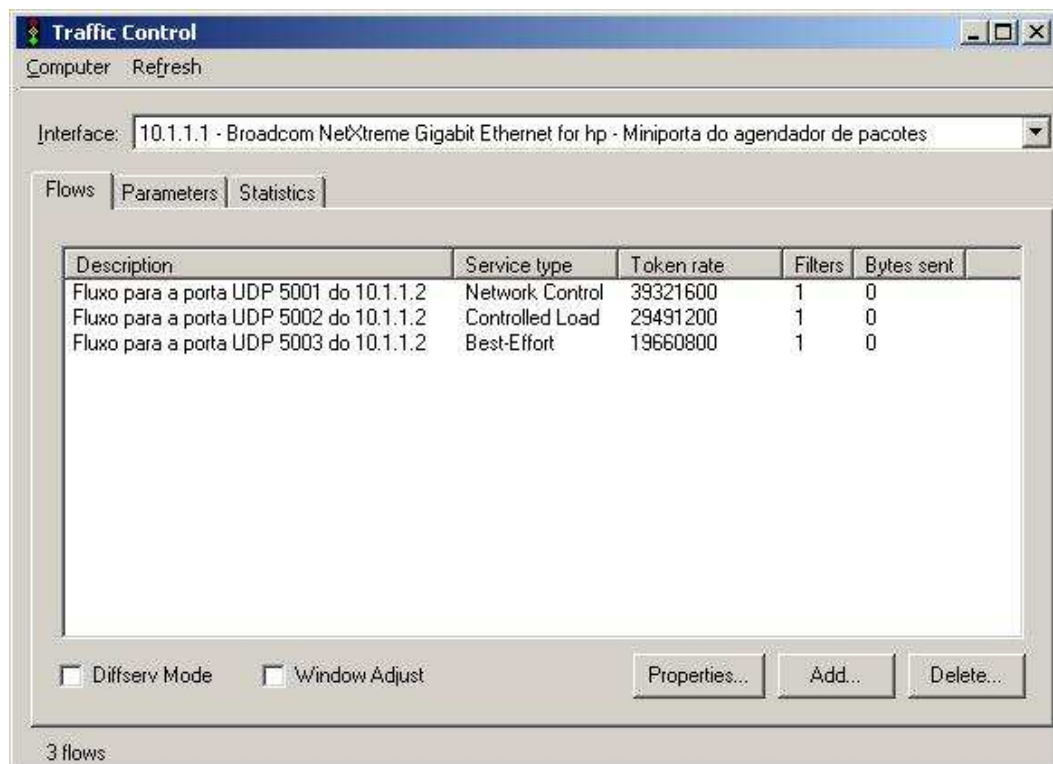


FIGURA 5.15: TCMon com as novas métricas.

Após a configuração do TCMon com as novas métricas, foram executados os arquivos em lote “servidor.bat” na máquina receptora e “cliente.bat” na máquina transmissora, obtendo os seguintes resultados:

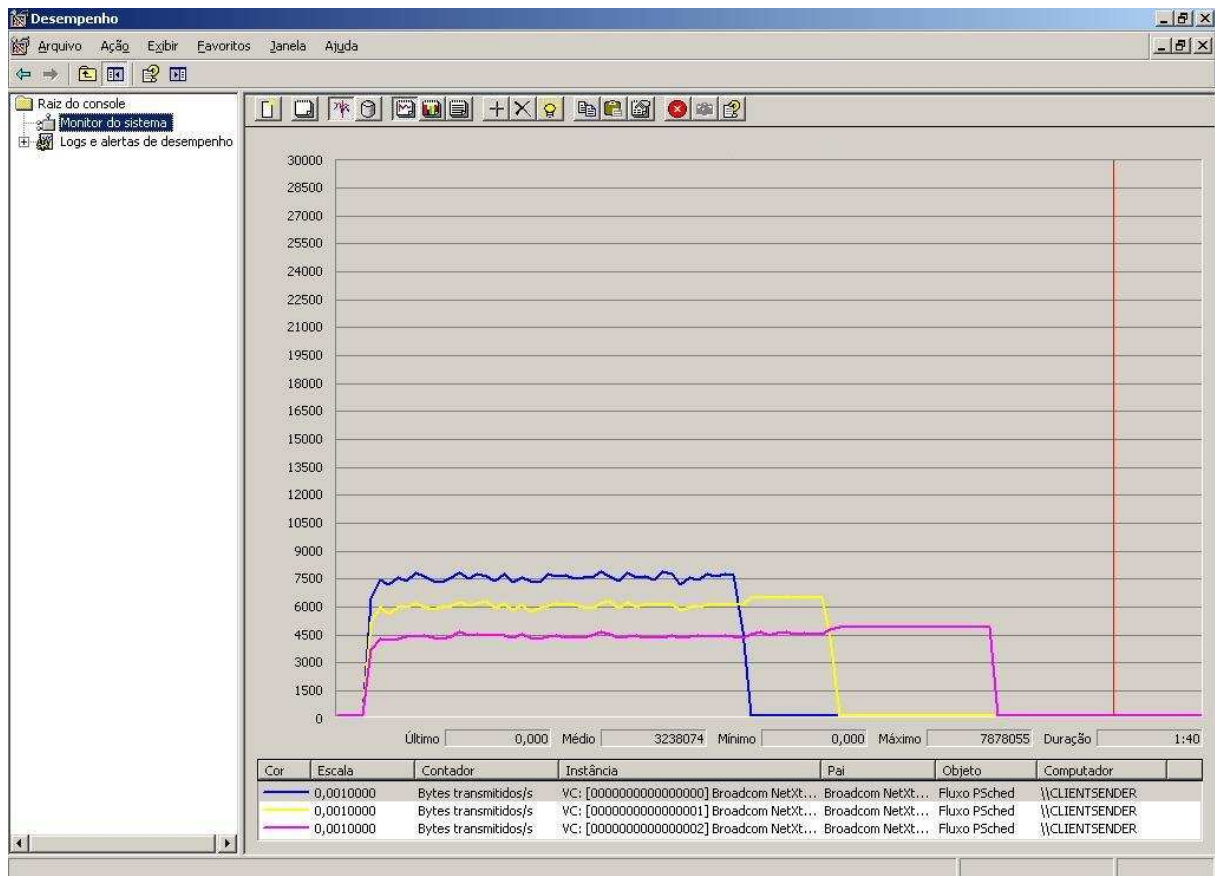


FIGURA 5.16: Perfmon mostrando a taxa de transferência em KB/s dos três fluxos com as novas métricas.

Pode-se observar que desta vez o fluxo para a porta UDP 5001 (gráfico azul) obteve prioridade no acesso a placa de rede, obtendo uma taxa de transferência média de 57,7 Mb/s em 43,6 segundos. Já o fluxo para a porta UDP 5002 (gráfico amarelo) obteve uma taxa de transferência média de 46,7 Mb/s em 59,9 segundos, e o fluxo para a porta UDP 5003 (gráfico vermelho) obteve uma taxa de transferência média de 34,6 Mb/s em 72,7 segundos.

Com o *Wireshark* foi possível capturar os pacotes dos três fluxos e verificar seu conteúdo. Nas figuras 5.17, 5.18 e 5.19 podemos visualizar que o campo DSCP está

preenchido com os valores correspondentes às classes de serviço conforme configurado no TCMon.

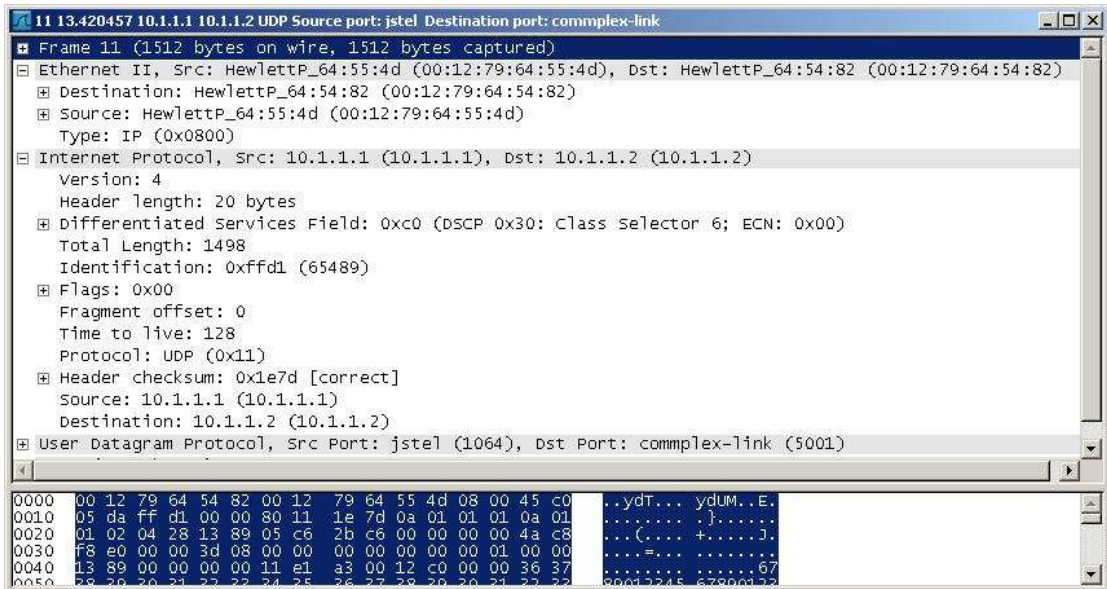


FIGURA 5.17: Conteúdo de um pacote do fluxo para a porta UDP 5001, com o campo DSCP preenchido com o valor 0x30, correspondente a classe de serviço *Network Control*.

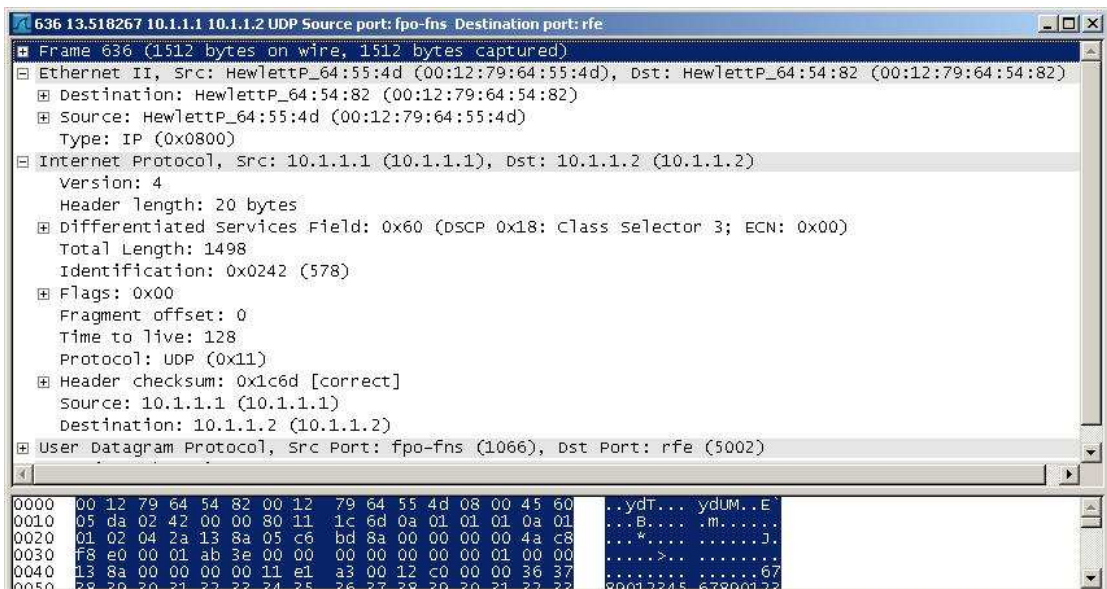


FIGURA 5.18: Conteúdo de um pacote do fluxo para a porta UDP 5002, com o campo DSCP preenchido com o valor 0x18, correspondente a classe de serviço *Controlled load*.

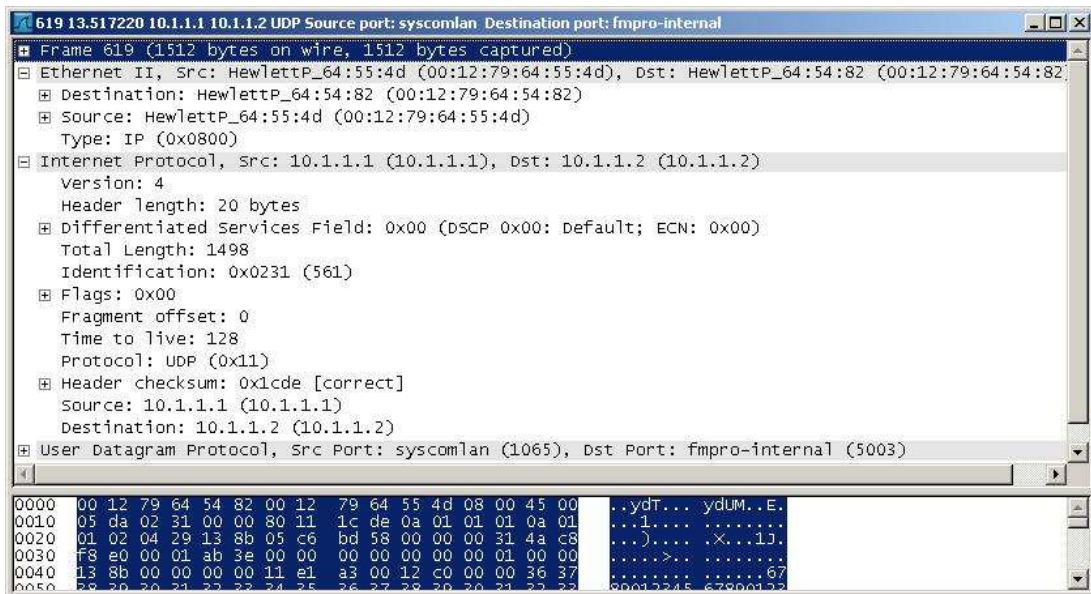


FIGURA 5.19: Conteúdo de um pacote do fluxo para a porta UDP 5003, com o campo DSCP preenchido com o valor 0x00, correspondente a classe de serviço *Best-effort*.

Em um terceiro teste, as taxas de transferência dos três fluxos foram alteradas de forma dinâmica no TCMon, alternando os valores entre os fluxos. O resultado pode ser observado na figura 5.20:

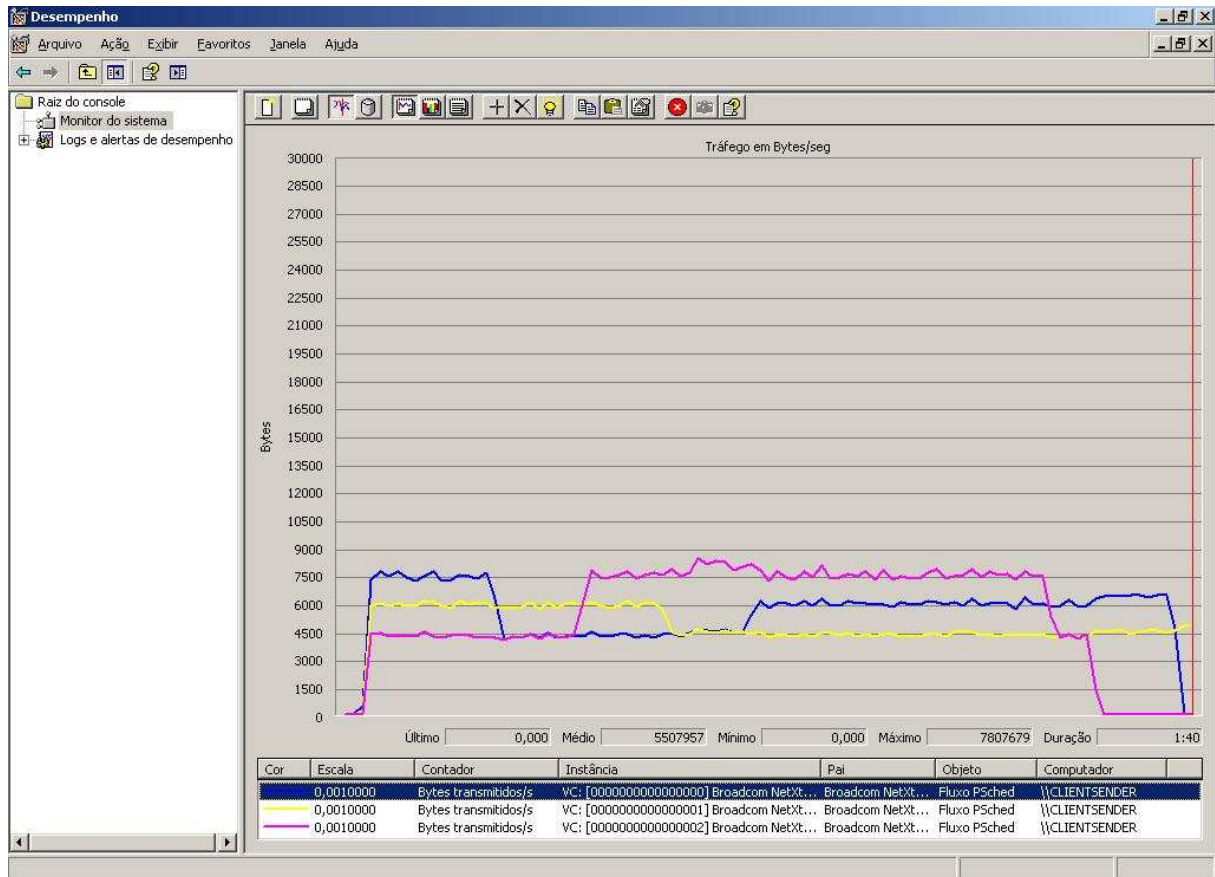


FIGURA 5.20: Taxas de transferência dos fluxos alteradas dinamicamente no TCMon.

6 CONCLUSÃO

Pesquisas apontam que, atualmente, 90% das estações de trabalho do mundo possuem um sistema operacional da *Microsoft* instalado [GMS09], e cada vez mais aplicações multimídia e de tempo real farão parte do tráfego dessas estações. Nesse cenário, onde a visão de convergência de serviços se torna cada dia mais vigente, é imperioso que conheçamos os mecanismos e tecnologias envolvidos na qualidade de serviço, em especial, os que são implementados nos sistemas operacionais da *Microsoft*.

Durante as pesquisas realizadas para este trabalho, constatamos que existe pouco material disponível que trate do assunto qualidade de serviço especificamente nos sistemas operacionais da *Microsoft*. Inclusive, não foram encontrados estudos anteriores realizados nesse sentido.

Com o objetivo de conhecer um pouco mais sobre o assunto, neste trabalho procuramos abordar as formas de se obter qualidade de serviço em ambientes suportados pelas plataformas *Windows XP* e *Windows Server 2003*. Vimos que os desenvolvedores podem fazer uso das APIs *Generic Quality of Service* (GQoS) e *Traffic Control* (TC), as quais integram a API *Windows Sockets*, para tornar suas aplicações capazes de aplicar métricas de QoS ao seu tráfego. Já os administradores de rede podem usar ferramentas específicas (por exemplo, o TCMon - *Traffic Control Monitor*), que interagem com a arquitetura de QoS do *Windows* através da API *Traffic Control*, para aplicar métricas de qualidade de serviço ao tráfego de aplicações que não sejam *QoS-aware*.

Nos testes realizados, os quais focaram na função do administrador de rede, constatamos que, atendidos alguns requisitos, é simples de implementar QoS em estações que utilizam o *Windows XP* (o mesmo serve para o *Windows Server 2003*). Com a utilização de

algumas ferramentas (TCMon, Iperf, *Windows Performance Monitor* e *Wireshark*), conseguimos simular e visualizar os resultados de situações nas quais determinado tráfego deveria ter prioridade em detrimento de outros. Vimos, assim, que é totalmente possível para o administrador de uma rede suportada pelo *Windows XP* eleger quais aplicações devem ter prioridade no seu tráfego para, a partir daí, determinar e aplicar uma política de QoS para esta rede.

Salientamos, entretanto, que para obter QoS fim-a-fim em uma rede é necessário que todos os dispositivos que a compõem (*switches*, roteadores) suportem as tecnologias utilizadas (IEEE 802.1p, DiffServ).

REFERÊNCIAS

- [ARD01] ARDEOLA, F. R. **Qualidade de Serviço em Redes IP: Serviços Diferenciados**. 2001. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.
- [BEN98] BENNETT, G. **Internetworking com TCP/IP: Tecnologia e Infraestrutura**. Rio de Janeiro: Infobook, 1998.
- [BLO09] Blog CCNA – QoS (Qualidade de Serviço). Disponível em: <<http://blog.ccna.com.br/2008/08/07/qos-qualidade-de-servico-parte-iiiv>>. Acesso em: 2009.
- [CHA09] Charles Spurgeon's Ethernet (IEEE 802.3) Site. Disponível em: <<http://www.ethermanage.com/metcalfe-drawing.html>>. Acesso em: 2009.
- [DAV09] DAVIES, J. QoS Support in Windows. **TechNet Magazine**. 2007. Disponível em: <<http://technet.microsoft.com/en-us/magazine/2007.02.cableguy.aspx>>. Acesso em: 2009.
- [DEL09] Dell Support Home Page. Disponível em: <<http://docs.us.dell.com/support/edocs/network/pc5212/bp/UG/pc5212cc.htm>>. Acesso em: 2009.
- [DIF09] An Architecture for Differentiated Services. Disponível em: <<http://www.ietf.org/rfc/rfc2475.txt>>. Acesso em: 2009.
- [FIL02] FILHO, F. Z.; TAKASHI, L. S. **QoS em Redes IP**. 2002. Trabalho de Conclusão de Curso (Especialização em Redes e Sistemas Distribuídos) – Instituto de Informática, UFRGS, Porto Alegre.
- [GMS09] Global Market Share Statistics. Disponível em: <<http://marketshare.hitslink.com/default.aspx>>. Acesso em: 2009.
- [HPT09] HP Technical Documentation. Disponível em: <<http://docs.hp.com/en/5992-0538/ar01s01.html>>. Acesso em: 2009.
- [HPT09a] HP Technical Documentation. Disponível em: <<http://www.docs.hp.com/en/T1453-90001/ch02s10.html?btnPrev=%AB%A>>.

Acesso em: 2009.

- [IEE09] 802.1D - MAC bridges. Disponível em: <<http://www.ieee802.org/1/pages/802.1D.html>>. Acesso em: 2009.
- [INT09] Integrated Services in the Internet Architecture: an Overview. Disponível em: <<http://www.ietf.org/rfc/rfc1633.txt>>. Acesso em: 2009.
- [POZ01] POZZEBON, S. **Análise Comparativa das Arquiteturas IntServ, DiffServ, MPLS e Roteamento Baseado em QoS para o oferecimento de QoS na Internet**. 2001. Trabalho Individual (Pós-Graduação em Computação) – Instituto de Informática, UFRGS, Porto Alegre.
- [QOS09] QOS Changes in Windows Server 2003 and Windows XP. Microsoft. 2003. Disponível em: <<http://support.microsoft.com/kb/318602/en-us/>>. Acesso em: 2009.
- [QOS09a] QOS Technical Reference. Microsoft. 2003. Disponível em: <<http://technet.microsoft.com/en-us/library/cc739780.aspx>>. Acesso em: 2009.
- [SAN09] SANTANA, H. **Qualidade de Serviço (QoS) em Redes IP: Princípios Básicos, Parâmetros e Mecanismos**. Disponível em: <<http://www.inf.ufsc.br/~bertoni/sistemas/aulas/QoS.pdf>>. Acesso em: 2009.
- [SOA95] SOARES, L. F. G.; LEMOS, G.; COLCHER, S. **Redes de Computadores: das LANs, MANs e WANs às Redes ATM**. 2 ed. Rio de Janeiro: Campus, 1995.
- [SOU06] SOUSA, L. B. **TCP/IP Básico & Conectividade em Redes**. 3 ed. São Paulo: Érica, 2006.
- [TAN03] TANENBAUM, A. S. **Redes de Computadores**. 4 ed. Rio de Janeiro: Campus, 2003.
- [TCP09] Microsoft Windows Server 2003 TCP/IP Implementation Details. Disponível em: <<http://www.microsoft.com/downloads/details.aspx?FamilyID=06c60bfe-4d37-4f50-8587-8b68d32fa6ee&displaylang=en>>. Acesso em: 2009.
- [TEC09] Microsoft TechNet – Resources for IT Professionals. Disponível em: <<http://technet.microsoft.com>>. Acesso em: 2009.