

5288865

Boletim Técnico da Escola Politécnica da USP
Departamento de Engenharia de Computação e
Sistemas Digitais

ISSN 1413-215X

BT/PCS/0222

Modelamento de Roteadores IP
para Análise de Atraso

Marcelo Blanes
Graça Bressan

São Paulo - 2002

de ok

O presente trabalho é um resumo da dissertação de mestrado apresentada por Marcelo Blanes, sob orientação da Profa. Dra. Graça Bressan: "Modelamento de Roteadores IP para Análise de Atraso", defendida em 06/09/2002, na EPUSP.

A íntegra da dissertação encontra-se à disposição com o autor e na biblioteca de Engenharia de Eletricidade da Escola Politécnica da USP.

FICHA CATALOGRÁFICA

Blanes, Marcelo

Modelamento de roteadores IP para análise de atraso / M. Blanes, G. Bressan. – São Paulo : EPUSP, 2002.

p. – (Boletim Técnico da Escola Politécnica da USP, Departamento de Engenharia de Computação e Sistemas Digitais, BT/PCS/0222)

1. Protocolos de comunicação 2. Simulação de sistemas I. Bressan, Graça II. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais III. Título IV. Série
ISSN 1413-215X

CDD 004.62
003.3

MODELAMENTO DE ROTEADORES IP PARA ANÁLISE DE ATRASO

Graça Bressan, Marcelo Blanes
LARC - Escola Politécnica
Universidade de São Paulo
05508-900 São Paulo - SP
{gbressan, mblanes}@larc.usp.br

RESUMO

O protocolo IP ("Internet Protocol") é amplamente utilizado, seja para fins de transferência de dados, gerenciamento ou outras aplicações de rede específicas. Devido a sua flexibilidade é um dos protocolos mais utilizados para desenvolvimento de recentes tecnologias como MPLS ("Multiprotocol Label Switching"), VoIP ("Voice over IP") entre outras.

Com o crescimento de utilização de redes IP, tanto para fins de interconexão de redes públicas (Internet) como para montagem de redes privadas novas exigências foram lançadas ao IP, como QoS ("Quality of Service"). Uma das maiores dificuldades do IP é, no entanto, se obter alguma garantia de qualidade bem como efetuar medições em uma rede real.

Essa dissertação apresenta a arquitetura, o projeto, a implementação e a validação de um simulador de roteadores IP, utilizando simulação por eventos discretos, ferramenta que permite criar vários roteadores IP e interligá-los, gerar e aplicar diversos tipos de tráfegos e coletar informações como atrasos, variação de atraso, picos de utilização das filas de entrada e saída e utilização de CPU.

O simulador foi concebido também para permitir a utilização de outras estratégias de filas além do FIFO ("First In First Out") e se realizar experimentos e coletar resultados sem ser necessário utilizar uma rede real e equipamentos de testes.

ABSTRACT

The IP protocol, also called as the Internet Protocol, is widely used for data transfer, management systems or other specific network applications. Due to this flexibility, it is a protocol with constant development of new technologies like MPLS ("Multiprotocol Label Switching"), VoIP ("Voice over IP"), and others.

Due to increasing demand for IP networks, either for providing connectivity for public network as Internet, or to connect private networks, new requirements were demanded to the IP, like QoS ("Quality of Service"). But one of the most difficult things in the IP protocol is to offer some type of guarantee and also to perform measures in a real network.

This dissertation presents the architecture, the project, the implementation and the validation of an IP router simulator using discrete event simulation, this tool allows the creation of many IP routers and connect them each other or with traffic generator from where many types of traffics can be generated, and collect information like delay, jitter, input and output queue utilization as well CPU process utilization.

The simulator was also implemented to allow another types of queueing strategies besides FIFO (First In First Out), to perform experiments and to collect results without using a real test bed or certification equipments

1 INTRODUÇÃO

Com o crescimento das redes IP, inúmeras aplicações passaram a utilizá-la como base de comunicação devido a sua grande flexibilidade. Para o correto funcionamento das aplicações, novas exigências tem sido requeridas do IP. O motivo principal é o fato de muitas instituições terem migrado as suas aplicações que utilizavam circuitos dedicados e se adaptado a essas redes estatísticas. Muitas aplicações como voz e vídeo requerem algumas garantias formalizadas como aplicações que requerem QoS ("Quality of service").

Parâmetros de QoS são não facilmente mensuráveis principalmente quando não se dispõe de equipamentos específico de medição e também quando não se possui o ambiente correto para realização de provas. Nem sempre é possível também definir quais são as melhores técnicas para se realizar QoS e avaliar a sua real eficiência.

Esta dissertação apresenta uma arquitetura, o projeto, a implementação e validação de simulador de roteadores IP, permitindo se definir as características de cada roteador, interligá-los entre si e a geradores de

tráfego. Os geradores de tráfego podem ser criados a partir de tráfegos reais de rede LAN ou WAN dando uma representação mais realística da topologia a ser testada. Usando o simulador é possível verificar o comportamento do roteador assim como do tráfego de saída, medindo-se o tempo de saída, atraso, atraso máximo e variação de atraso.

O objetivo do trabalho é modelar um roteador, comparar com um real e validar o modelo. O critério de validação serão os tempos de atraso e suas variações. Após a validação será verificada a influência do tipo de estratégia de fila para tráfegos de dados, voz e vídeo.

Algoritmos de roteamento não serão simulados e implementados pelo fato de eles não serem representativos na análise do atraso.

Uma vez validado o modelo, o simulador será capaz que reproduzir os resultados de roteador real e poderá ser usado como ferramenta para simular várias topologias de rede IP, permitindo se estudar o comportamento desse ambiente quanto ao atraso sem a necessidade de nenhum recurso físico.

O artigo está organizado da seguinte forma: a seção 2 trata de alguns conceitos básicos sobre o protocolos IP e tipos de tráfegos, a seção 3 trata da

arquitetura de roteadores IP e suas principais características para análise de atraso, a seção 4 mostra as principais estratégias de filas, a seção 5 apresenta o modelo do roteador implementado, validação e resultados e a seção 6, as principais conclusões do trabalho.

2 CONCEITOS BÁSICOS DE IP

Breve definição do protocolo IP:

“Camada de rede da pilha TCP/IP que oferece serviços não orientados a conexão. O protocolo IP possui características como endereçamento, especificação do tipo de serviço, fragmentação e remontagem de pacotes além de opções de segurança documentado na RFC 791”.

Atualmente existem duas versões do IP em operação, a versão 4 e a versão 6. A de número 5 foi utilizado por um outro protocolo da Internet, o “Stream Protocol”, versão 2. Com a versão 4 até 232 combinações de endereços são possíveis. Com a versão 6, 264 combinações de endereços IP são possíveis. A versão 6 foi desenvolvida para endereçar uma futura escassez de endereços IP.

O presente trabalho se baseia apenas na versão 4, por ser mais comumente utilizada.

2.1 PACOTE IP E SEUS CAMPOS

Uma vez que o objetivo é construir um simulador de roteadores IP, o pacote IP deve ser precisamente definido. Da RFC 791 (1981) verificamos que o pacote IP é formado pelos seguintes campos:

0	4	8	16	19	24	31
VERS	HLN	SERVICE TYPE	TOTAL LENGTH			
IDENTIFICATION			FLAGS	FRAGMENT OFFSET		
TIME TO LIVE		PROTOCOL	HEADER CHECKSUM			
SOURCE IP ADDRESS						
DESTINATION IP ADDRESS						
IP OPTIONS (IF ANY)					PACKING	
DATA						

O pacote IP é formado basicamente de um cabeçalho de no mínimo de 20 bytes adicionado pelo campo de dados. Dentre os diversos campos do pacote IP, os mais importantes para o projeto e implementação do roteador serão os campos de “Service Type” ou “Type of Service”, conhecido como ToS, “Total Length”, “Time to Live”, “Source address” e “Destination address”.

O campo de endereço destino (“Destination address”) de 32 bits é um parâmetro essencial para permitir o roteamento em conjunto com o campo “Time to Live” (8 bits). Através do par endereço origem (“Source address”) também de 32 bits e destino é possível se definir um fluxo de dados. O campo de tamanho total (“Total Length”) de 16 bits será importante para o

cálculo de atraso. No valor de ToS, estaremos apenas interessados nos 3 primeiros bits que são os valores de “IP precedence”.

2.2 TIPOS DE TRÁFEGOS QUANTO À APLICAÇÃO

Podemos classificar os tipos de tráfego quanto à aplicação nos seguintes tipos:

- Tráfego de voz
- Tráfego de vídeo
- Tráfego de dados

Cada tipo de tráfego tem suas características assim definidas bem como seus requisitos de QoS. Veremos a seguir em detalhes cada tipo de tráfego citado.

2.2.1 TRÁFEGO DE VOZ E SEUS REQUISITOS

O tráfego de voz é representado por qualquer pacote IP que contenha informações de voz, seja ela a voz codificada em si ou qualquer mensagem pertinente aos protocolos de sinalização de voz. Isso significa que se uma chamada telefônica for estabelecida dentro de uma rede IP usando a tecnologia de voz sobre IP (VoIP), será considerado tráfego de voz todos os pacotes de sinalização para estabelecer a chamada, a voz digitalizada e codificada em pacotes IP, sinalização para manutenção da chamada, assim com a sinalização para término de chamada. Segundo o ITU-T G.114 as seguintes recomendações são feitas para voz.

Atraso entre fala e escuta unidirecional	Descrição
0-150 ms	Adequado à maioria das aplicações sensível a atraso como voz
150-400 ms	Ainda aceitável, mas começa a influenciar na qualidade da voz.
400 ms ou mais	Não aceitável para maioria das aplicações sensíveis a atraso.

Tabela 1 - Recomendações para voz

Outra recomendação para voz é que a variação de atraso não seja maior que 10 ms para garantir a boa qualidade da voz codificada.

A banda exigida pela voz pode ficar entre 64 kbps e 8 kbps dependendo da técnica de codificação utilizada bem como as tecnologias de transporte envolvidas. A voz sobre IP compactada utilizando o G729a utiliza 8 kbps sendo que quando encapsulada no pacote IP mais o cabeçalho UDP pode variar entre 24 kbps e 12,5 kbps.

2.2.2 TRÁFEGO DE VÍDEO E SEUS REQUISITOS

Um dos tipos de tráfegos mais complexos e mais exigentes são os tráfegos de vídeo. São tráfegos sensíveis a atraso e que geralmente requerem um alta desempenho da rede em termos de atraso.

Existem diversas técnicas para codificação de vídeo, mas iremos nos restringir ao MPEG por ser uma plataforma aberta e muito utilizada. Conforme referenciado em Garrett; Willinger (1994) o tráfego de vídeo tem um comportamento de tráfego VBR, e com as seguintes características:

Codificação	DCT ("Discrete Cosine Transformation")
Duração	2 horas
Número de quadros	171 000
Dimensão do quadro	480 linhas por 504 colunas
Taxa de quadros	24 por segundos
Banda Média consumida	5,34 Mbps

Tabela 2 - Estatísticas de um tráfego de vídeo VBR

O vídeo utilizado para extrair essas estatísticas foi o filme "Stars Wars".

Outro requisito importante do vídeo é o atraso e máxima variação do atraso. Segundo Baldi; Ofek (2000), o atraso máximo fim-a-fim aceitável para um vídeo é de 100ms. Uma vez que vídeo e áudio precisam estar sincronizados, a variação máxima deve ser em torno de 10 ms.

2.2.3 TRÁFEGO DE DADOS E SEUS REQUISITOS

Os tráfegos de dados são em geral tráfegos pouco modeláveis compostos por rajadas e picos de utilização de banda. Leland, et al (1993) se preocuparam em definir exatamente como os tráfegos de rede local, que utilizam tecnologia Ethernet como acesso, se comportam. Uma série de amostras foram realizadas em diferentes redes com tipos de aplicações e períodos diversos. A tabela abaixo mostra as estatísticas de um dos meses:

Mês	Período	% da utilização da rede Ethernet
Outubro 1989	Total	15,7%
	Baixa utilização(06:25am-7:35am)	10,4%
	Média utilização(2:25pm-3:25pm)	18,4%
	Alta Utilização(4:25pm-5:25pm)	30,7%

Tabela 3- Estatísticas de utilização de rede LAN

Depois de muitos estudos e levantamentos, definiu-se que o tráfego local pode ser considerado tráfego "auto-similar" (de comportamento fractal), ou seja,

tráfegos que não podem ser modelados através de expressões matemáticas conhecidas. Tráfegos mais específicos podem ter alguns dos seus parâmetros aproximados por distribuições estatísticas matemáticas como mostra a tabela abaixo, citada em Sikdar. et al. (1997) e em Floyd; Paxson (2001):

Aplicação	Distribuição	Duração	Dados
Telnet	Exponencial	Log-normal	Pareto
WWW	Exponencial	Log-normal	Auto-similar
FTP	Exponencial	Log-normal	Auto-similar
SMTP	Exponencial	Log-normal	Auto-similar

Tabela 4 - Distribuições para os parâmetros de várias aplicações

2.3 CARACTERIZAÇÃO DE ATRASO EM REDES IP

Atraso de pacote é um dos principais parâmetros a serem considerados na análise de uma rede quanto a QoS. O atraso ou tempo que um pacote leva para atravessar uma rede desde da sua geração até a sua recepção pelo destinatário é resultado dos seguintes atrasos, classificados em:

- Atraso de serialização
- Atraso de propagação
- Atraso de chaveamento

2.3.1 ATRASO DE SERIALIZAÇÃO

É o tempo que um dispositivo gasta para transmitir um pacote em uma dada velocidade. O atraso de serialização depende da banda do "link" assim como do tamanho do pacote a ser transmitido. Um pacote de 64 bytes na velocidade de 3Mbps, por exemplo, leva 171ms (micro segundos) para ser transmitido. Percebe-se que o atraso de serialização depende da banda, por exemplo, o mesmo pacote de 64 bytes na velocidade de 19.2 kbps leva 26ms (mili segundos). O atraso por serialização também é referido como atraso de transmissão.

2.3.2 ATRASO DE PROPAGAÇÃO

É o tempo que se leva para o bit transmitido ir do transmissor até o receptor dentro do enlace ("link"). Ele é significativo porque, na melhor das hipóteses, é uma fração da velocidade da luz. Esse tipo de atraso é proporcional à distância e do tipo de mídia utilizado no enlace, mas não ao tamanho da banda. Para enlaces WAN, o atraso de propagação é da ordem de milissegundos. Se uma fibra ótica for utilizada e se a distância a ser percorrida for de 1000 milhas, o atraso de propagação será aproximadamente 10 ms (milissegundos).

2.3.3 ATRASO DE CHAVEAMENTO

O atraso por chaveamento é o tempo que um dispositivo leva para comutar um pacote da interface receptora para a interface transmissora. Esse tempo pode variar em função do equipamento utilizado, mas geralmente é menor que 10ms (micro segundos).

Supondo que uma rede não esteja congestionada e que as filas não se formaram nos roteadores, o atraso total será uma soma dos atrasos de serialização, propagação e chaveamento. Note que os atrasos de serialização se tornam menos significativos quando a velocidade de transmissão aumenta e os atrasos de chaveamento e propagação menos significativos quanto se diminui a velocidade das interfaces.

Admitindo que os atrasos por propagação e chaveamento são pequenos se tratando de distâncias pequenas e equipamentos com grande capacidade de processamento, os atrasos de transmissão serão os atrasos mais significativos.

3 ARQUITETURA DE ROTEADORES IP

3.1 MODELO GENÉRICO

Podemos definir que um roteador IP possui os seguintes componentes básicos:

- Subsistema de entrada e saída
- Máquina de encaminhamento de pacotes
- Subsistema de seleção de pacotes

A figura a seguir ilustra esse modelo que foi baseado em McDysan (2000):

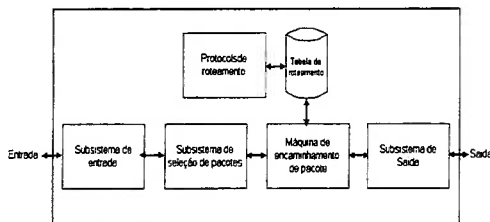


Figura 1- Modelo de um roteador IP

O subsistema de entrada e saída é composto por controladores de porta que efetuam as seguintes funções:

- “Bufferização” dos pacotes de entrada e saída
- Requisição para tratamento de pacotes
- Sincronização com o meio físico
- Checagem de erros

O subsistema de seleção de pacotes é a lógica capaz de selecionar qual pacote deve ser tratado e em qual instante.

A máquina de encaminhamento de pacote ou “Forwarding Engine” é unidade que decide qual

interface ou quais interfaces (no caso de “Multicast”) o pacote deve sair. Eventualmente, pelo fato de que um roteador IP não ser orientado à conexão, esse pacote pode ser descartado ou “dropped”. Para que o pacote possa sair pela porta destino correta, a máquina se baseia em uma base de dados alimentada pelos protocolos de roteamento.

Esse mesmo modelo pode ser estendido para um modelo distribuído se multiplicando os sub-sistemas de entrada, saída, seleção e encaminhamento de pacotes.

3.2 ARQUITETURA DE ROTEADORES DA JUNIPER

Nesse item vamos descrever o funcionamento de roteadores comerciais da Juniper Networks como descrito em Semeria (2002).

A arquitetura desses roteadores é composta por dois componentes:

• Máquina de encaminhamento de pacotes (“Packet Forwarding Engine”). Capaz de encaminhar até 40 milhões de pacotes por segundo qualquer que seja o tamanho do pacote

• Máquina de roteamento (“Routing Engine”). Realiza atualizações na tabela de roteamento e no sistema de gerenciamento. Essa máquina consiste em vários processos de protocolos de roteamento que rodam em modo protegido. A máquina de roteamento tem uma conexão direta de 100 Mbps com a máquina de encaminhamento de pacotes.

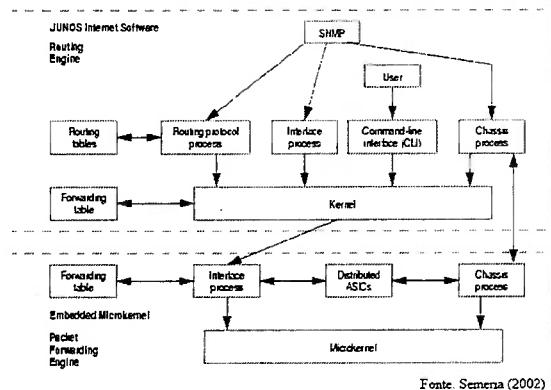


Figura 2- Arquitetura de roteadores da Juniper

3.2.1 MÁQUINA DE ENCAMINHAMENTO DE PACOTES

A máquina de encaminhamento (“Packet Forwarding Engine”) de pacotes é responsável por encaminhar os pacotes da interface de entrada até saída.

3.2.2 FLUXO DE UM PACOTE ATRAVÉS DO ROTEADOR

Basicamente, um pacote passa pela interface de recepção, pelo “switching fabric” que representa o

BUS do equipamento, e finalmente pela interface de transmissão conforme representado na figura a seguir. Quando um pacote chega na interface de entrada, a controladora também chamada de PIC ("Physical interface cards") realiza todo tratamento específico da mídia como checagem de erros e sincronização de quadros.

A controladora passa uma sequência serial de bits para FPC ("Flexible PIC Concentrator"). O FPC também quebra o pacote em blocos de 64 bytes e passa para esses blocos para o gerenciador de "buffer" distribuído da ASIC ("Arbiter Application-Specific Integrated Circuit"). O gerenciador de buffer distribuído da ASIC escreve então os blocos no buffer de memória de pacotes, que é distribuído eventualmente para todos os FPCs instalados no roteador.

Em paralelo com a bufferização, o gerenciador de buffer distribuído da ASIC extrai a informação necessária do pacote para que o roteamento seja realizado e passa a informação para o processador de Internet que realiza uma varredura completa na tabela de roteamento encontrando a interface de saída. A tabela de roteamento pode encaminhar todos os pacotes de "unicast" que não tem campo de opções e pacotes "multicast" que tenham sido recentemente mantidos em "caches". Pacote com campos de opções ou pacotes de "multicast" não mantidos em "caches" são encaminhados para máquina de roteamento para resolução.

Nesse ponto o ponteiro do pacote é enfileirado e não mais o pacote em si. Cada porta de saída tem quatro filas, cada uma delas podendo ser configurada para dividir a banda do enlace. Vários fatores podem contribuir para ordenação da fila como, por exemplo, o bits de "IP precedence" o endereço destino ou o algoritmos de RED e WRED, que serão descritos em na seção 3.7. Quando o ponteiro do pacote chega na frente da fila e ela esta pronta para transmissão, os blocos de memória são lidos do buffer de memória de pacotes de forma que o pacote é remontado e passado para mídia específica da PIC para transmissão na linha.

3.2.3 MÁQUINA DE ROTEAMENTO

A máquina de roteamento lida com todos os protocolos de roteamento e outros processos de software que controlam as interfaces do roteador, alguns componentes de chassi, sistema de gerenciamento e o acesso do usuário ao roteador. Esses processos rodam no topo do núcleo do sistema operacional e interage com a máquina de encaminhamento de pacotes.

O núcleo do sistema operacional foi desenvolvido a partir do sistema operacional "FreeBSD".

3.3 ARQUITETURA DE ROTEADORES DA CISCO

A Cisco System possui uma grande variedade de equipamentos com capacidade de roteamento. Esses

equipamentos são destinados desde aplicações SOHO como para backbone de ISPs. A arquitetura é bem diversificada em função da aplicação do roteador. Iremos estudar um roteador da família 7500 que um roteador de grande porte.

O roteador 7500 é constituído basicamente por um processador de roteamento e switching chamado de RSP ("Route and Switching Processor") e suas interfaces de entrada e saída.

A figura abaixo ilustra a arquitetura básica do 7500:

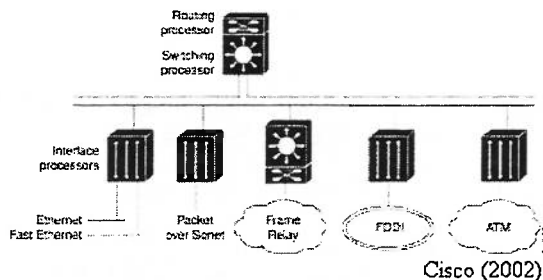


Figura 3- Arquitetura do roteador 7500 da Cisco

O roteador 7500 também utiliza processadores dedicados por interface chamado de VIP ("Versatile Interface Processors") baseados na arquitetura RISC. As VIPs mantêm um "cache" da tabela de roteamento chamada também de FIB ("Forwarding Information Base"). O Bus que interliga a RSP às VIPs tem velocidade 1 Gbps.

O roteamento, ou encaminhamento de pacotes, é composto basicamente de dois processos:

- Fazer a decisão de encaminhamento por roteamento
- Mover os pacotes para a interface de saída por comutação

O processo de roteamento é um processo mais intensivo e de maior latência que o processo de comutação. O primeiro pacote é processado pelo processo de roteamento onde o melhor caminho será escolhido. Uma vez que o primeiro pacote é processado, os demais pacotes, que possuem o mesmo destino, serão processados pelo processo de switching uma vez que uma tabela de caching se forma dentro de cada interface de entrada. Esse caching é distribuído por todas as interfaces do roteador e o processador central da RSP apenas administra o cache. Uma vez que a informação estiver armazenada em todas as VIPs o processador principal não é mais consultado. Isso faz com que todo processamento seja realizado pela VIP de forma distribuída. Esse mecanismo ou técnica é chamado "dCEF" ou "distributed Cisco Express Forwarding".

Cada interface contém um buffer de I/O dedicado associado ao CPU da mesma. Esse buffer fica localizado na SRAM ("Shared Packet Memory").

Uma vez que a VIP decidiu o destino do pacote baseado nas tabelas de "cache" o roteador manipula apenas o ponteiro do pacote.

Além da SRAM, o roteador possui ainda a RAM onde roda o sistema operacional e onde a memória de pacotes ("Packet Memory") reside. A memória de pacotes é utilizada pelo "Public Buffer pools" e são áreas temporárias para armazenamento de pacotes utilizados quando o buffer I/O não é suficiente. Diferente dos buffers de I/O os "Public Buffer pools" são alocados por máquina e são separados em função do valor de MTU de cada tipo de tecnologia de interface.

4 TIPOS DE ESTRATÉGIA DE FILAS

Outro aspecto relevante nos roteadores é o tipo de estratégia de fila que pode ser adotada para selecionar os pacotes tanto nas filas de entrada como de saída. Nos itens a seguir vamos abordar alguns tipos de estratégia de filas, seu funcionamento, características e benefícios.

4.1 "RANDOM EARLY DETECTION"

O "Random Early Detection" conhecido como RED é uma técnica baseada em descarte de pacotes cuja probabilidade aumenta quando os recursos da fila diminuem.

O "Random Early Detection" (RED) conforme descrito em Floyd; Jacobson (1993) é um mecanismo para evitar congestionamento que tira vantagens do mecanismo de controle do TCP. Descartando randomicamente os pacotes antes de períodos de alto congestionamento, RED avisa a fonte emissora do pacote para diminuir a sua taxa de transmissão. Assumindo que a fonte utiliza TCP na sua comunicação, ela irá diminuir a sua transmissão até que todos os pacotes cheguem no destino, indicando que o congestionamento foi solucionado.

4.1.1 BENEFÍCIOS

Quando uma fila não está com o RED configurado, os buffers de saída ficam cheios durante os períodos de maior tráfego e congestionamento. Nessa situação ocorre descarte de pacotes ou chamado também de "tail drops" pelo fato de todos os recursos do buffer de saída estarem esgotados. Se existirem várias conexões TCP ativas nesse instante, atravessando a mesma interface congestionada, vários descartes acontecerão devido ao funcionamento do controle de transmissão do TCP. Isso irá fazer com que cada fonte transmissora reduza a sua taxa de transmissão ao mesmo tempo. Assim que as transmissões diminuírem o TCP irá notificar que o "canal" está menos congestionado e todas as transmissões TCP irão aumentar as suas taxas causando um efeito de ida-e-volta.

Utilizando o RED, esse efeito é reduzido uma vez que alguns pacotes são selecionados para serem descartados de uma forma aleatória, quando sinais de congestionamento são percebidos na interface de saída com RED habilitado.

O RED possui diferentes limites e pesos para cada "IP precedence", permitindo que se controle diversos tipos de tráfegos marcados com diferentes valores de "IP precedence". O "IP precedence" são os três primeiros bits do campo de ToS ("Type of Service") do pacote IP.

Existem ainda outros estudos como o de Christiansen et al. (2001) onde vários ambientes e tipos de tráfegos foram simulados para se avaliar o ganho de performance para tráfego Web utilizando RED. Esse estudo mostra que os parâmetros RED devem ser cuidadosamente configurados para que a performance do RED seja melhor que um tráfego FIFO.

4.1.2 RESTRIÇÕES

O RED possui as seguintes restrições:

- Apenas útil para aplicações TCP/IP uma vez que o TCP possui um mecanismo de controle de congestionamento, outras aplicações podem não reagir ao descarte por congestionamento e até eventualmente retransmitir os pacotes descartados e mandá-los na mesma taxa de transmissão.

- É possível distinguir diferentes tipos de tráfego para o IP. Tráfego não IP serão considerados como "IP Precedence" 0 que são pacotes de menor prioridade.

4.1.3 FUNCIONAMENTO

O seguinte algoritmo é utilizado toda vez que um pacote chega a uma determinada fila de saída:

O tamanho médio da fila é calculado através da seguinte equação:

$$Tamanho_Médio = Tamanho_Médio_Antigo * \left(1 - \frac{1}{2^n}\right) + Tamanho_da_fila_corrente * \frac{1}{2^n}$$

Onde n é um fator exponencial que varia em função da velocidade de transmissão da interface.

O funcionamento do RED ocorre da seguinte forma:

- Se o tamanho médio da fila é menor que o limite mínimo o pacote é enfileirado
- Se o tamanho médio da fila está entre o limite mínimo e máximo, o pacote pode ser descartado ou não, dependendo da probabilidade de descarte do tipo do pacote.
- Se a fila é maior que o limite máximo o pacote é automaticamente descartado

A probabilidade de um pacote ser descartado utiliza como base o limite mínimo, máximo e um denominador. Quando o tamanho da fila chega ao limite mínimo, a probabilidade de um pacote ser descartado é igual ao tamanho da fila dividido por um denominador. Por exemplo, se o denominador é igual a 100, a cada 100 pacotes 1 pacote é descartado quando a fila atinge o tamanho do limite máximo, se o tamanho médio da fila estiver entre o limite máximo e

mínimo, em média 1 a cada 200 pacotes serão descartados. Quando a fila ultrapassa o limite máximo a probabilidade de descarte passar a ser igual a 1 ou seja todo pacote é descartado. Os valores de limite máximo, mínimo podem ser definidos por "IP precedence".

4.2 "WEIGHTED FAIR QUEUEING"

Como o próprio nome sugere, o "Weighted Fair Queueing" (WFQ) tem como objetivo dividir os recursos de uma fila de forma distribuída e equitativa. O WFQ também é conhecido como uma técnica de balanceamento "bit-bit".

É um algoritmo baseado em enfileiramento por fluxo de tráfego (porta TCP/UDP e IP origem destino). Basicamente ele realiza duas funções:

- Seleciona os tráfegos interativos, colocando na frente da fila e diminuindo o tempo que o mesmo demora a ser transmitido.
- Balanceia os demais tráfegos de acordo com a capacidade da fila

4.2.1 BENEFÍCIOS

- Permite que uma aplicação não ocupe todo os recursos de uma fila de saída
- Melhora o tempo de resposta de aplicações interativas
- Funciona melhor que uma divisão por TDM uma vez que os recursos são alocados por tráfego dinamicamente.

4.2.2 RESTRIÇÕES

Algoritmo complexo e dispendioso para CPU processar ($O(\log(n))$ por pacote, segundo Shreedhar; Varghese (1995))

4.2.3 FUNCIONAMENTO

Basicamente o tráfego é classificado segundo os endereços IP origem e destino, protocolo e identificador de sessão. O número de filas representa o número de tipos de fluxos identificados. A seleção se dá pelo tipo da necessidade de cada fluxo, priorizando pacote com "IP precedence" mais alto ou tráfegos sensíveis a atraso.

4.3 DEFICIT ROUND ROBIN

O "Deficit Round Robin" ou DRR é uma aproximação do WFQ. Conforme citado em Shreed; Varghese (1995), esse algoritmo foi desenvolvido com objetivo de tornar a implementação de um balanceamento de fila com melhor performance. Conforme citado no item anterior, o WFQ tem um custo de processamento muito alto quando utilizado. Esse custo faz com que seja impraticável o seu uso em filas de interfaces de alta velocidade. O algoritmo do DRR leva $O(1)$ para

processar cada pacote. O DRR seleciona o pacote baseado no campo de "IP precedence" do pacote IP criando diversas filas para cada valor (0-7). Através de contador chamado de "deficit" ele balanceia as diversas filas alterando a fila que é atendida e que cujos pacotes serão efetivamente transmitidos.

4.3.1 BENEFÍCIOS

Em seguida apresentamos alguns benefícios do DRR.

- O DRR é um algoritmo que pode ser implementado para qualquer velocidade de transmissão
- Pode ser utilizado para qualquer tráfego IP
- Permite se atribuir pesos diferentes para cada fila simulando uma garantia de banda.

4.3.2 RESTRIÇÕES

O DRR possui algumas restrições como, por exemplo, quando a classificação do tráfego precisa ser realizada de uma forma que não seja utilizar o campo "IP Precedence" torna impraticável o seu uso.

4.3.3 FUNCIONAMENTO

O algoritmo do DRR se utiliza os seguintes componentes e variáveis.

- Valor de "Quantum"
- Valor de "Deficit"
- Fila de baixa latência
- Filas

Os valores de "Quantum" e "Deficit" são alocados por fila. O valor de "quantum" representa a quantidade de bytes que podem ser enviados cada vez que a fila é atendida. Esse valor é único para cada fila do DRR. O valor de "deficit" é o número de bytes remanescentes que podem ser atendidos pelo DRR em uma determinada fila. Por exemplo, a cada pacote de 100 bytes transmitidos em uma fila o valor de "deficit" é decrementado por 100. O valor de "Quantum" não deve ser inferior ao MTU do pacote da interface de saída que utiliza DRR. Se essa regra não for seguida uma fila pode ficar "presa" por causa de um pacote. Quando o valor de "deficit" fica negativo uma outra fila é atendida. Toda vez que o algoritmo muda de fila e começa a atender uma outra os valores de "deficit" da fila que começa a ser servida são incrementados do valor de "quantum". Se uma fila ficar vazia o valor de "deficit" é zerado e ela não será atendida até que algum pacote seja enfileirado nela.

A fila de baixa latência é a fila que é atendida sempre que existir alguma informação nela contida ou é a fila é atendida sempre que uma fila "normal" acabou de ser servida. A esses dois modos que a fila de baixa latência pode operar são dados os nomes de "Strict Priority Mode" e "Alternate Mode" respectivamente. O modo "Strict Priority Mode" vai garantir total prioridade para os pacotes que pertencem a essa fila,

por outro lado, essa fila pode ocupar todo o recurso da interface de saída para um único tipo de tráfego. O modo alternado permite ainda uma boa garantia de baixa latência, mas pode eventualmente causar uma variação de atraso ao tráfego que a fila comporta. O MDRR ("Modified Deficit Round Robin") é uma implementação do DRR cuja restrição é suportar no máximo até 6 tipos de filas e uma fila de baixa latência.

5 PROJETO DO ROTEADOR, SIMULAÇÃO E VALIDAÇÃO

5.1 Projeto do roteador

Vários modelos foram apresentados e discutidos, entretanto, com o objetivo de viabilizar a validação do modelo, a arquitetura da Cisco foi adotada com algumas modificações para se aplicar a modelos de roteadores de pequeno porte.

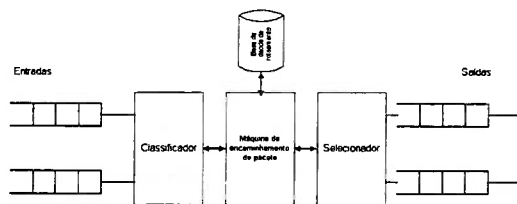


Figura 4- Modelo do roteador a ser simulado

Esse modelo é bastante próximo das arquiteturas vistas nos itens anteriores. A principal diferença é que ela não visa performance ou escalabilidade.

O módulo classificador será responsável em controlar as filas de entrada de tal forma que quando vários pacotes chegarem na fila apenas um será tratado por vez na respectiva ordem de chegada.

A máquina de encaminhamento de pacote recebe o pacote de entrada e com base nas informações da tabela de roteamento que decide qual deverá ser o próximo pulo e por consequência a interface de saída.

O seccionador é capaz de decidir qual pacote deverá ser enfileirado na interface de saída para ser transmitido. O seccionador irá se basear na estratégia da fila e no tipo de pacote para tomar a decisão de qual pacote será enfileirado para transmissão dentro dos pacotes armazenados no buffer de saída de cada fila.

A seguir apresentamos a arquitetura física do roteador:

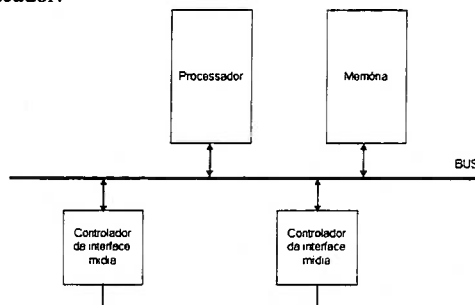


Figura 5- Arquitetura física do modelo de roteador a ser simulado

Basicamente temos três tipos de componentes:

- Memória
- Processador
- Controladores de mídia

Os controladores de mídia são responsáveis por todo o sincronismo e adaptação ao meio físico e checagem de erros. Os controladores deverão também avisar o processador quando um pacote deverá ser tratado através do BUS. A controladora que receber o primeiro pacote será atendida, as demais controladoras ou os pacotes seguintes irão ser enfileirados no buffer de entrada.

O processador é responsável em tratar cada pacote e encaminhar para fila de saída. Apenas um pacote pode ser tratado por vez e armazenado no buffer de saída.

A memória é dividida em vários segmentos:

- Memória para buffers de entrada
- Memória para buffers de saída
- Memória para tabela de roteamento
- Memória para buffer público

A memória de buffers de entrada e saída são regiões de tamanho fixo e independente para cada controladora. O tamanho em bytes é calculado em função da estratégia de fila adotada, tipo de fila (entrada ou saída) e valor de MTU da interface.

A memória de buffers públicos é uma região da memória que pode ser utilizada por qualquer interface. Os buffers públicos somente são utilizados quando o buffer de I/O da interface é totalmente utilizado e quando a estratégia da fila é FIFO.

Aspectos como uso da memória pelo sistema operacional, alocações dinâmicas da memória pelos processos não serão tratadas aqui por não ser relevantes para a análise de atraso em um roteador IP.

5.2 Simulação

Para simulação do modelo de roteador proposto foi utilizada a técnica de simulação de sistemas por eventos discretos. A simulação será orientada por eventos que são iniciados pela entrada de pacotes no sistema e pela comunicação entre cada componente do mesmo. A ferramenta utilizada para a simulação foi a linguagem C ANSI. O principal motivo de se utilizar uma linguagem de simulação foi não só pela grande flexibilidade como pelo baixo custo. Outra vantagem de se utilizar essa ferramenta é portabilidade do código gerado para qualquer plataforma e possibilidade de integração com outros simuladores.

5.3 Validação

Nessa sessão iremos criar vários ambientes de simulação para validar o modelo do roteador escolhido em relação a um modelo real, bem como a

implementação do simulador. Para isto utilizou-se um instrumento de geração de tráfegos e medições de atrasos com precisão de 0,000001 segundos.

A seguinte topologia foi utilizada como base para simulação e coleta de informações:

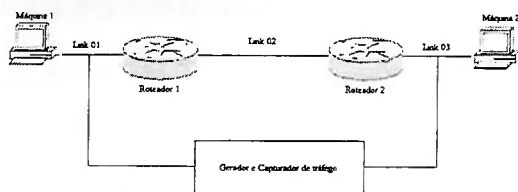


Figura 6 – Topologia física tomada como base para validação

As máquinas 1 e 2 foram utilizadas para simular um transmissor e receptor real respectivamente. A máquina 2 é responsável em absorver o tráfego e gerar uma tabela ARP válida para o roteador 2. Os link 01 e 03 são enlaces da tecnologia Ethernet e o link 02 é um enlace serial usando PPP como protocolo de nível 2. Os roteadores 1 e 2 são dois Cisco 3640 e o gerador e capturador de tráfego é um Interwatch 95000 da Net Test.

5.3.1 VALIDAÇÃO 1, LINK CENTRAL DE 64 KBPS, PACOTES DE 64 BYTES, TRÁFEGO RAJADA

Nessa validação os links 01 e 03 da Figura 22 são ambos de 10Mbytes Ethernet e o link 02 é de 64Kbps. A estratégia de fila utilizada é o FIFO para todas as interfaces. É gerado um tráfego do roteador 1 interface do link 01 com destino o roteador 2 interface do link 03. O intervalo entre cada quadro é de aproximadamente 81ns.

O seguinte tráfego foi gerado na interface do roteador 1 que se liga ao link 1:

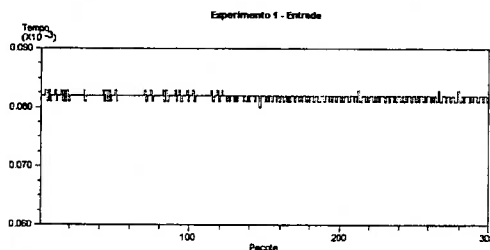


Figura 6 – Gráfico da tráfego de entrada do experimento 1

O eixo X mostra o número de pacote e o eixo Y o seu atraso com relação ao pacote anterior. No gráfico da figura acima o intervalo entre cada pacote é praticamente constante com variação de 2ns. O tamanho do pacote IP foi de 64 bytes. No “burst” gerado, 300 pacotes foram injetados no link 1 e apenas

173 no ambiente real e 172 no simulador chegaram no link 3, conforme mostra a tabela a seguir.

A gráfico a seguir mostra duas saídas obtidas do experimento e do simulador.

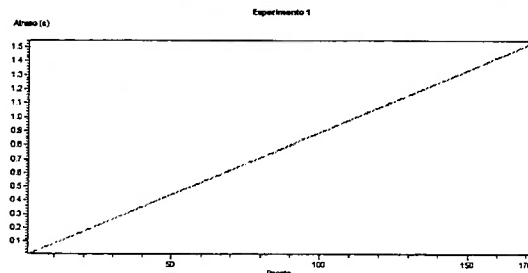


Figura 7 – Gráficos das saídas do experimento 1

	Atraso médio dos pacotes (s)	Variação do atraso	Número de pacotes transitados
Experimento	0,778319	0,447417	173
Simulador	0,771765	0,444142	172
Erros (%)	-0,842137	-0,731968	

Tabela 4 – Resultados do experimento 1

A linha em verde (de cor mais clara) e a linha em vermelho (de cor mais escura) da Figura 24 representam o resultado do simulador e experimento respectivamente. Elas mostram qual foi o atraso (eixo X) de cada pacote (eixo Y). Os pacotes foram capturados no link 3 que liga a interface do roteador 2. Pelo gráfico, observamos que cada pacote deve um aumento linear de atraso devido ao congestionamento ocorrido na fila de saída. Podemos notar que as linhas são praticamente coincidentes mostrando que o simulador calcula com precisão o atraso gerado pela limitação da taxa de saída da interface do link 2. Outro valor importante é o número de pacotes que são perdidos no sistema. A tabela acima mostra que o erro do simulador foi de apenas 1 pacote, mostrando que o simulação dos buffer públicos também é funcional.

5.3.2 VALIDAÇÃO 2, LINK CENTRAL DE 64 KBPS, PACOTES DE 1500 BYTES, TRÁFEGO RAJADA

O experimento 2 é bastante semelhante ao 1 com a principal diferença do tamanho de pacote, que foi alterado para 1500 bytes. No gráfico abaixo podemos verificar qual foi o atraso entre quadros para cada pacote que entrou no ambiente através do link 1.

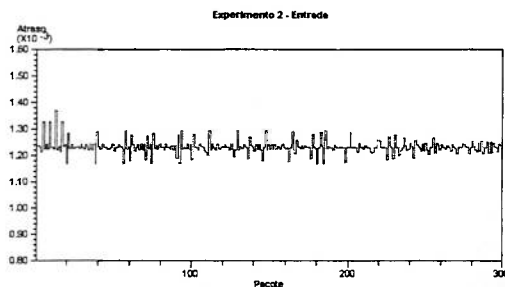


Figura 8 – Gráfico de Tráfego de entrada para o experimento 2

Através da análise do gráfico verificamos que o atraso médio entre os quadros ficou em torno de 1,2ms. O aumento do atraso com relação ao item anterior se deve ao fato que o tamanho do quadro é maior.

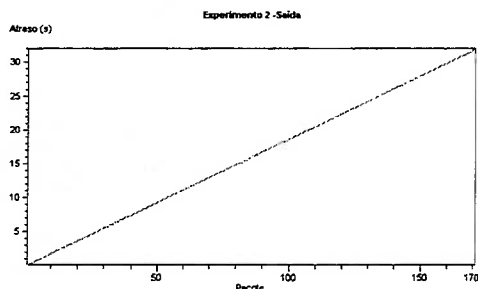


Figura 9 – Gráficos de saída do experimento 2

A linha em verde (de cor mais clara) e a linha em vermelho (de cor mais escura) acima representam o resultado do simulador e experimento respectivamente.

	Atraso médio dos pacotes (s)	Variação do atraso	Número de pacotes transitados
Experimento	16,11272	9,273531	171
Simulador	16,11103	9,273128	171
Erros (%)	-0,010435	-0,004347	

Tabela 5 – Resultados do experimento 2

Analisando tanto o gráfico com a tabela com os resultados do experimento 2 verificamos que o simulador apresentou um resultado satisfatório na medição dos tempos de atraso e variação.

5.3.3 VALIDAÇÃO 3, LINK CENTRAL DE 2MBPS, TRÁFEGO DE VÍDEO

Nesse teste iremos transmitir um vídeo da máquina 1 codificado em MPEG-1 para a máquina 2. O mesmo ambiente montado para a validação 6 foi utilizado com a diferença que a transmissão é “unicast” ou seja existe um fluxo de dados apenas na direção da máquina 1 para a máquina 2. O vídeo transmitido tem duração média de 3 minutos e 20 segundos com taxa de transmissão média de 1,5Mbps por segundo conforme a codificação MPEG-1 padrão.

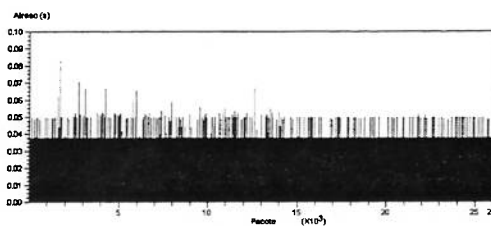


Figura 9 – Atraso entre pacote da fonte transmissora de vídeo

Na figura acima verificamos a diferença entre pacotes com relação ao atraso mostrando que o tráfego de vídeo é bastante variável. A tabela abaixo mostra algumas estatísticas sobre o vídeo capturado:

Número de pacotes total	26404
Tamanho médio do pacote	1457,34 bytes
Banda média calculada	1501,16K
Atraso mínimo entre pacotes	0,000156 segundos
Atraso máximo entre pacotes	0,313539 segundos

Tabela 6 – Estatísticas da entrada de vídeo capturada

Para o tráfego capturado foi gerado um arquivo que contém exatamente o perfil do vídeo capturado. Esse arquivo foi utilizado no simulador para se obter os seguintes gráficos:

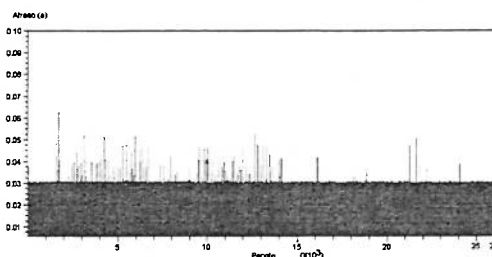


Figura 10 – Gráficos dos atrasos do tráfego de vídeo na rede simulada e real

O gráfico de cor verde (de cor clara) e o de cor vermelha (de cor escura) representam o tempo gasto para cada pacote atravessar a rede simulada e a rede real, respectivamente.

A tabela a seguir mostra algumas estatísticas obtidas para o ambiente real e o simulado.

	Atraso médio (s)	Variação do atraso	Número de pacotes transitados
Experimento	0,018512	0,008583	26404
Simulador	0,017991	0,008622	26404
Erros (%)	-2,822084	0,445027	-

Tabela 7 – Resultado do experimento 3 para o fluxo de dados de vídeo

Seguindo as recomendações para se obter uma boa qualidade de vídeo, a média do atraso ficou menor que 100ms e a variação de atraso ficou menor que 10ms. Comparando-se os gráficos e resultados da tabela, verifica-se que por volta do pacote 14000 o simulador apresentou picos em instantes diferentes do real voltando a normalizar apenas nos últimos pacotes (a partir do pacote número 26000). Isso pode ter ocorrido pela perda de algum pacote na captura do vídeo de entrada ou algum erro de transmissão no enlace central que liga os outros dois roteadores. Para uma melhor análise uma tabela com os 10 maiores valores de atraso foi construída.

Maior valor de k-ésima ordem, k=	Valores de atraso Ambiente Real	Pacote	Valores de atraso Ambiente Simulador	Pacote
1	0,062802	1730	0,061468	1730
2	0,060218	1725	0,058769	1725
3	0,056878	1729	0,055483	1729
4	0,055634	1735	0,05449	1735
5	0,054296	1724	0,05279	1724
6	0,05358	10	0,051858	3135
7	0,05229	12645	0,051856	12645
8	0,052089	3135	0,050941	5960
9	0,051266	5960	0,050915	4265
10	0,051131	4265	0,049507	1728

Tabela 8 – Maiores valores de atraso fim a fim real e simulados

A tabela acima nos mostra que, com exceção do 6º maior valor, que deslocou os valores do ambiente real para baixo, os pacotes que experimentaram um maior atraso no real foram os mesmos praticados no ambiente simulado.

5.3.4 TESTE MDRR, TRÁFEGO DE VÍDEO E CBR

Nesse teste, utilizando FIFO e MDDR como estratégia de fila da interface do roteador 1 que está conectado ao roteador 2, iremos aplicar um tráfego de vídeo pelo link 1 e um tráfego CBR pelo link 2. O tráfego CBR contém as seguintes características, pacotes de 64 bytes, 1 pacote a cada 0,001 segundos e intervalo de duração e início igual ao da transmissão de vídeo. Os pacotes de vídeo serão marcados com "IP Precedence" igual a 4 e os pacotes de dados com valor igual a 0. A programação do MDDR fica:

- Fila de baixa latência (número 0) para pacotes com "IP Precedence" igual à 5, valor de "Quantum" igual a 1500 (configuração não relevante)
- Fila número 1 para pacotes com "IP Precedence" igual à 0, igual a 1500
- Fila número 2 para pacotes com "IP Precedence" igual à 4, igual a 6000

Os gráficos e tabela mostram os resultados do teste:

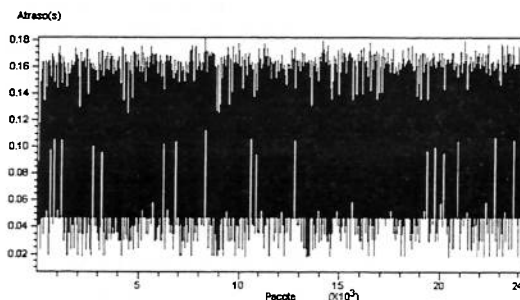


Figura 11 – Gráfico do atraso dos pacotes de vídeo usando FIFO

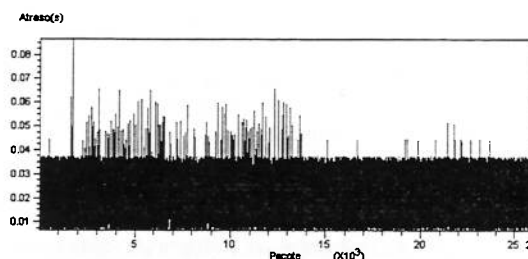


Figura 12 – Gráfico do atraso dos pacotes de vídeo usando MDRR

	Tipo do tráfego	Atraso médio (s)	Variação do atraso	Número de pacotes transitados
FIFO	CBR	0,124100	0,024775	189364
	Vídeo	0,125414	0,026909	24354
MDRR	CBR	0,038159	0,16996	152747
	Vídeo	0,022538	0,009363	25962

Tabela 9 – Resultados do teste MDRR para fila FIFO e MDRR

Utilizando a técnica de FIFO verificamos que o tempo e a variação de atraso ficam fora do especificado por Baldi; Ofek (2000), que é de 100ms para o atraso máximo e 10 ms para a variação do atraso. Utilizando o MDRR e programando-se os valores corretamente foi possível reduzir o atraso médio para 22,538 ms e a variação para 9,363 ms ficando dentro do aceitável. Isso mostra que a fila MDRR foi bastante eficiente para chegarmos nos requisitos necessários para o perfil de tráfego de vídeo.

6 CONCLUSÕES

Este artigo procurou identificar os aspectos relevantes para a modelagem de um roteador IP cujo estudo seria baseado no atraso e nas variações de atraso que a arquitetura modelada pode causar, por ser uma métrica importante na integração de dados, vídeo e voz. A partir do entendimento da forma operação de

um roteador IP e da sua arquitetura, um simulador de roteador IP foi implementado.

Como principal contribuição, descrevemos a arquitetura, o projeto, a implementação e a validação da ferramenta de simulação, detalhando o seu funcionamento, as estruturas de dados envolvidas, e os conceitos de simulação por eventos discretos implementados utilizados pelo simulador.

O simulador foi concebido de forma a tornar possível uma fácil integração com outros simuladores e também para adição de novas funcionalidades como novos tipos de estratégia de filas, ou com as quais se deseja obter uma comparação de funcionamento e eficácia.

Um aspecto importante a ser destacado neste trabalho foi a técnica utilizada para validação do modelo em relação ao sistema real, utilizando ferramentas e equipamentos que fornecem valores de atraso com grande precisão. Essa técnica permitiu verificarmos o correto funcionamento do simulador através de várias validações e testes com diversos tipos de tráfegos, comparando sempre com um ambiente real e o simulado, utilizando os equipamentos de captura e logs do simulador.

Verificamos também a eficiência da estratégia de fila MDRR em relação ao FIFO que permite diferenciar tráfegos de dados, voz e vídeo e permitir que os requisitos de QoS podem ser alcançados utilizando uma programação correta dos parâmetros do seu algoritmo.

Dessa forma, demonstramos que o simulador é uma ferramenta bastante interessante para se avaliar a performance de roteador com relação ao atraso de pacotes IP.

GARRETT, M. W.; WILLINGER, W. Analysis, Modeling and Generation of Self-Similar VBR Video Traffic. ACM/ SIGCOMM, 1994.

LELAND, W. E. et al. On Self-Similar Nature of Ethernet Traffic. ACM 0-89791-619, 1993, 183-193.

SEMERIA, C. Supporting Differentiated Service Classes: Queue Scheduling Disciplines. Juniper Networks, Inc, disponível em: <http://www.juniper.net/techcenter/techpapers/200020-01.html>.

SHREEDHAR, M.; VARGHESE, G. Efficient Fair Queuing using Deficit Round Robin. ACM 0-89791, 1995, 231-242.

RFC 791, Information Sciences Institute University of Southern California., IETF, Setembro de 1981.

REFERÊNCIAS BIBLIOGRÁFICAS

BALDI, M.; OFEK, Y. End-to-End Delay Analysis of Videoconferencing over Packet-Switched Networks. IEEE/ACM Transactions on Networking, Vol 8, No 4, Agosto 2000, 479-492.

CHRISTIANSEN, M. et al. Tuning RED for Web Traffic. IEEE/ACM Transactions on Networking, Vol 9, No 3, Junho 2001, 249-264.

FALOYD, S.; JACABSON, V. Random Early Detection Gateway for Congestion Avoidance. IEEE/ACM Transactions on Networking, Agosto de 1993.

FELDMANN, A. et al. Deriving Traffic Demands for Operational IP Networks: Methodology and Experience. IEEE/ACM Transactions on Networking, Vol 9, No 3, Junho 2001, 265-279.

BOLETINS TÉCNICOS - TEXTOS PUBLICADOS

- BT/PCS/9301 - Interligação de Processadores através de Chaves Ômicron - GERALDO LINO DE CAMPOS, DEMI GETSCHKO
- BT/PCS/9302 - Implementação de Transparência em Sistema Distribuído - LUÍSA YUMIKO AKAO, JOÃO JOSÉ NETO
- BT/PCS/9303 - Desenvolvimento de Sistemas Especificados em SDL - SIDNEI H. TANO, SELMA S. S. MELNIKOFF
- BT/PCS/9304 - Um Modelo Formal para Sistemas Digitais à Nível de Transferência de Registradores - JOSÉ EDUARDO MOREIRA, WILSON VICENTE RUGGIERO
- BT/PCS/9305 - Uma Ferramenta para o Desenvolvimento de Protótipos de Programas Concorrentes - JORGE KINOSHITA, JOÃO JOSÉ NETO
- BT/PCS/9306 - Uma Ferramenta de Monitoração para um Núcleo de Resolução Distribuída de Problemas Orientado a Objetos - JAIME SIMÃO SICHMAN, ELERI CARDOSO
- BT/PCS/9307 - Uma Análise das Técnicas Reversíveis de Compressão de Dados - MÁRIO CESAR GOMES SEGURA, EDIT GRASSIANI LINO DE CAMPOS
- BT/PCS/9308 - Proposta de Rede Digital de Sistemas Integrados para Navio - CESAR DE ALVARENGA JACOBY, MOACYR MARTUCCI JR.
- BT/PCS/9309 - Sistemas UNIX para Tempo Real - PAULO CESAR CORIGLIANO, JOÃO JOSÉ NETO
- BT/PCS/9310 - Projeto de uma Unidade de Matching Store baseada em Memória Paginada para uma Máquina Fluxo de Dados Distribuído - EDUARDO MARQUES, CLAUDIO KIRNER
- BT/PCS/9401 - Implementação de Arquiteturas Abertas: Uma Aplicação na Automação da Manufatura - JORGE LUIS RISCO BECERRA, MOACYR MARTUCCI JR.
- BT/PCS/9402 - Modelamento Geométrico usando do Operadores Topológicos de Euler - GERALDO MACIEL DA FONSECA, MARIA ALICE GRIGAS VARELLA FERREIRA
- BT/PCS/9403 - Segmentação de Imagens aplicada a Reconhecimento Automático de Alvos - LEONCIO CLARO DE BARROS NETO, ANTONIO MARCOS DE AGUIRRA MASSOLA
- BT/PCS/9404 - Metodologia e Ambiente para Reutilização de Software Baseado em Composição - LEONARDO PUJATTI, MARIA ALICE GRIGAS VARELLA FERREIRA
- BT/PCS/9405 - Desenvolvimento de uma Solução para a Supervisão e Integração de Células de Manufatura Discreta - JOSÉ BENEDITO DE ALMEIDA, JOSÉ SIDNEI COLOMBO MARTINI
- BT/PCS/9406 - Método de Teste de Sincronização para Programas em ADA - EDUARDO T. MATSUDA, SELMA SHIN SHIMIZU MELNIKOFF
- BT/PCS/9407 - Um Compilador Paralelizante com Detecção de Paralelismo na Linguagem Intermediária - HSUEH TSUNG HSIANG, LÍRIA MATSUMOTO SAITO
- BT/PCS/9408 - Modelamento de Sistemas com Redes de Petri Interpretadas - CARLOS ALBERTO SANGIORGIO, WILSON V. RUGGIERO
- BT/PCS/9501 - Síntese de Voz com Qualidade - EVANDRO BACCI GOUVÊA, GERALDO LINO DE CAMPOS
- BT/PCS/9502 - Um Simulador de Arquiteturas de Computadores "A Computer Architecture Simulator" - CLAUDIO A. PRADO, WILSON V. RUGGIERO
- BT/PCS/9503 - Simulador para Avaliação da Confiabilidade de Sistemas Redundantes com Reparo - ANDRÉA LUCIA BRAGA, FRANCISCO JOSÉ DE OLIVEIRA DIAS
- BT/PCS/9504 - Projeto Conceitual e Projeto Básico do Nível de Coordenação de um Sistema Aberto de Automação, Utilizando Conceitos de Orientação a Objetos - NELSON TANOMARU, MOACYR MARTUCCI JUNIOR
- BT/PCS/9505 - Uma Experiência no Gerenciamento da Produção de Software - RICARDO LUIS DE AZEVEDO DA ROCHA, JOÃO JOSÉ NETO
- BT/PCS/9506 - MétodoOO - Método de Desenvolvimento de Sistemas Orientado a Objetos: Uma Abordagem Integrada à Análise Estruturada e Redes de Petri - KECHI HIRAMA, SELMA SHIN SHIMIZU MELNIKOFF
- BT/PCS/9601 - MOOPP: Uma Metodologia Orientada a Objetos para Desenvolvimento de Software para Processamento Paralelo - ELISA HATSUE MORIYA HUZITA, LÍRIA MATSUMOTO SATO
- BT/PCS/9602 - Estudo do Espalhamento Brillouin Estimulado em Fibras Ópticas Monomodo - LUIS MEREGE SANCHES, CHARLES ARTUR SANTOS DE OLIVEIRA
- BT/PCS/9603 - Programação Paralela com Variáveis Compartilhadas para Sistemas Distribuídos - LUCIANA BEZERRA ARANTES, LÍRIA MATSUMOTO SATO
- BT/PCS/9604 - Uma Metodologia de Projeto de Redes Locais - TEREZA CRISTINA MELO DE BRITO CARVALHO, WILSON VICENTE RUGGIERO

- BT/PCS/9605 - Desenvolvimento de Sistema para Conversão de Textos em Fonemas no Idioma Português - DIMAS TREVIZAN CHBANE, GERALDO LINO DE CAMPOS
- BT/PCS/9606 - Sincronização de Fluxos Multimídia em um Sistema de Videoconferência - EDUARDO S. C. TAKAHASHI, STEFANIA STIUBIENER
- BT/PCS/9607 - A importância da Completeza na Especificação de Sistemas de Segurança - JOÃO BATISTA CAMARGO JÚNIOR, BENÍCIO JOSÉ DE SOUZA
- BT/PCS/9608 - Uma Abordagem Paraconsistente Baseada em Lógica Evidencial para Tratar Exceções em Sistemas de Frames com Múltipla Herança - BRÁULIO COELHO ÁVILA, MÁRCIO RILLO
- BT/PCS/9609 - Implementação de Engenharia Simultânea - MARCIO MOREIRA DA SILVA, MOACYR MARTUCCI JÚNIOR
- BT/PCS/9610 - Statecharts Adaptativos - Um Exemplo de Aplicação do STAD - JORGE RADY DE ALMEIDA JUNIOR, JOÃO JOSÉ NETO
- BT/PCS/9611 - Um Meta-Editor Dirigido por Sintaxe - MARGARETE KEIKO IWAI, JOÃO JOSÉ NETO
- BT/PCS/9612 - Reutilização em Software Orientado a Objetos: Um Estudo Empírico para Analisar a Dificuldade de Localização e Entendimento de Classes - SELMA SHIN SHIMIZU MELNIKOFF, PEDRO ALEXANDRE DE OLIVEIRA GIOVANI
- BT/PCS/9613 - Representação de Estruturas de Conhecimento em Sistemas de Banco de Dados - JUDITH PAVÓN MENDONZA, EDIT GRASSIANI LINO DE CAMPOS
- BT/PCS/9701 - Uma Experiência na Construção de um Tradutor Inglês - Português - JORGE KINOSHITA, JOÃO JOSÉ NETO
- BT/PCS/9702 - Combinando Análise de "Wavelet" e Análise Entrópica para Avaliar os Fenômenos de Difusão e Correlação - RUI CHUO HUEI CHIOU, MARIA ALICE G. V. FERREIRA
- BT/PCS/9703 - Um Método para Desenvolvimento de Sistemas de Computacionais de Apoio a Projetos de Engenharia - JOSÉ EDUARDO ZINDEL DEBONI, JOSÉ SIDNEI COLOMBO MARTINI
- BT/PCS/9704 - O Sistema de Posicionamento Global (GPS) e suas Aplicações - SÉRGIO MIRANDA PAZ, CARLOS EDUARDO CUGNASCA
- BT/PCS/9705 - METAMBI-OO - Um Ambiente de Apoio ao Aprendizado da Técnica Orientada a Objetos - JOÃO UMBERTO FURQUIM DE SOUZA, SELMA S. S. MELNIKOFF
- BT/PCS/9706 - Um Ambiente Interativo para Visualização do Comportamento Dinâmico de Algoritmos - IZAURA CRISTINA ARAÚJO, JOÃO JOSÉ NETO
- BT/PCS/9707 - Metodologia Orientada a Objetos e sua Aplicação em Sistemas de CAD Baseado em "Features" - CARLOS CÉSAR TANAKA, MARIA ALICE GRIGAS VARELLA FERREIRA
- BT/PCS/9708 - Um Tutor Inteligente para Análise Orientada a Objetos - MARIA EMÍLIA GOMES SOBRAL, MARIA ALICE GRIGAS VARELLA FERREIRA
- BT/PCS/9709 - Metodologia para Seleção de Solução de Sistema de Aquisição de Dados para Aplicações de Pequeno Porte - MARCELO FINGUERMAN, JOSÉ SIDNEI COLOMBO MARTINI
- BT/PCS/9801 - Conexões Virtuais em Redes ATM e Escalabilidade de Sistemas de Transmissão de Dados sem Conexão - WAGNER LUIZ ZUCCHI, WILSON VICENTE RUGGIERO
- BT/PCS/9802 - Estudo Comparativo dos Sistemas da Qualidade - EDISON SPINA, MOACYR MARTUCCI JR.
- BT/PCS/9803 - The VIBRA Multi-Agent Architecture: Integrating Purposive Vision With Deliberative and Reactive Planning - REINALDO A. C. BIANCHI, ANNA H. REALI C. RILLO, LELIANE N. BARROS
- BT/PCS/9901 - Metodologia ODP para o Desenvolvimento de Sistemas Abertos de Automação - JORGE LUIS RISCO BECCERRA, MOACYR MARTUCCI JUNIOR
- BT/PCS/9902 - Especificação de Um Modelo de Dados Bitemporal Orientado a Objetos - SOLANGE NICE ALVES DE SOUZA, EDIT GRASSIANI LINO DE CAMPOS
- BT/PCS/9903 - Implementação Paralela Distribuída da Dissecção Cartesiana Aninhada - HILTON GARCIA FERNANDES, LIRIA MATSUMOTO SATO
- BT/PCS/9904 - Metodologia para Especificação e Implementação de Solução de Gerenciamento - SERGIO CLEMENTE, TEREZA CRISTINA MELO DE BRITO CARVALHO
- BT/PCS/9905 - Modelagem de Ferramenta Hipermídia Aberta para a Produção de Tutoriais Interativos - LEILA HYODO, ROMERO TORI
- BT/PCS/9906 - Métodos de Aplicações da Lógica Paraconsistente Anotada de Anotação com Dois Valores-LPA2v com Construção de Algoritmo e Implementação de Circuitos Eletrônicos - JOÃO I. DA SILVA FILHO, JAIR MINORO ABE
- BT/PCS/9907 - Modelo Nebuloso de Confiabilidade Baseado no Modelo de Markov - PAULO SÉRGIO CUGNASCA, MARCO TÚLIO CARVALHO DE ANDRADE
- BT/PCS/9908 - Uma Análise Comparativa do Fluxo de Mensagens entre os Modelos da Rede Contractual (RC) e Colisões Baseada em Dependências (CBD) - MÁRCIA ITO, JAIME SIMÃO SICHMAN

- BT/PCS/9909 – Otimização de Processo de Inserção Automática de Componentes Eletrônicos Empregando a Técnica de Times Assíncronos – CESAR SCARPINI RABAK, JAIME SIMÃO SICHMAN
- BT/PCS/9910 – MIISA – Uma Metodologia para Integração da Informação em Sistemas Abertos – HILDA CARVALHO DE OLIVEIRA, SELMA S. S. MELNIKOFF
- BT/PCS/9911 – Metodologia para Utilização de Componentes de Software: um estudo de Caso – KAZUTOSI TAKATA, SELMA S. S. MELNIKOFF
- BT/PCS/0001 – Método para Engenharia de Requisitos Norteados por Necessidades de Informação – ARISTIDES NOVELLI FILHO, MARIA ALICE GRIGAS VARELLA FERREIRA
- BT/PCS/0002 – Um Método de Escolha Automática de Soluções Usando Tecnologia Adaptativa – RICARDO LUIS DE AZEVEDO DA ROCHA, JOÃO JOSÉ NETO
- BT/PCS/0101 – Gerenciamento Hierárquico de Falhas – JAMIL KALIL NAUFAL JR., JOÃO BATISTA CAMARGO JR.
- BT/PCS/0102 – Um Método para a Construção de Analisadores Morfológicos, Aplicado à Língua Portuguesa, Baseado em Autômatos Adaptativos – CARLOS EDUARDO DANTAS DE MENEZES, JOÃO JOSÉ NETO
- BT/PCS/0103 – Educação pela Web: Metodologia e Ferramenta de Elaboração de Cursos com Navegação Dinâmica – LUISA ALEYDA GARCIA GONZÁLEZ, WILSON VICENTE RUGGIERO
- BT/PCS/0104 – O Desenvolvimento de Sistemas Baseados em Componentes a Partir da Visão de Objetos – RENATA EVANGELISTA ROMARIZ RECCO, JOÃO BATISTA CAMARGO JÚNIOR
- BT/PCS/0105 – Introdução às Gramáticas Adaptativas – MARGARETE KEIKO IWAI, JOÃO JOSÉ NETO
- BT/PCS/0106 – Automação dos Processos de Controle de Qualidade da Água e Esgoto em Laboratório de Controle Sanitário – JOSÉ BENEDITO DE ALMEIDA, JOSÉ SIDNEI COLOMBO MARTINI
- BT/PCS/0107 – Um Mecanismo para Distribuição Segura de Vídeo MPEG – CÍNTIA BORGES MARGI, GRAÇA BESSAN, WILSON VICENTE RUGGIERO
- BT/PCS/0108 – A Dependence-Based Model for Social Reasoning in Multi-Agent Systems – JAIME SIMÃO SICHMAN
- BT/PCS/0109 – Ambiente Multilíngua de Programação – Aspectos do Projeto e Implementação – APARECIDO VALDEMIR DE FREITAS, JOÃO JOSÉ NETO
- BT/PCS/0110 – LETAC: Técnica para Análise de Tarefas e Especificação de Fluxo de Trabalho Cooperativo – MARCOS ROBERTO GREINER, LUCIA VILELA LEITE FILGUEIRAS
- BT/PCS/0111 – Modelagem ODP para o Planejamento de Sistemas de Potência – ANIRIO SALLES FILHO, JOSÉ SIDNEI COLOMBO MARTINI
- BT/PCS/0112 – Técnica para Ajuste dos Coeficientes de Quantização do Padrão MPEG em Tempo Real – REGINA M. SILVEIRA, WILSON V. RUGGIERO
- BT/PCS/0113 – Segmentação de Imagens por Classificação de Cores: Uma Abordagem Neural – ALEXANDRE S. SIMÕES, ANNA REALI COSTA
- BT/PCS/0114 – Uma Avaliação do Sistema DSM Nautilus – MARIO DONATO MARINO, GERALDO LINO DE CAMPOS
- BT/PCS/0115 – Utilização de Redes Neurais Artificiais para Construção de Imagem em Câmara de Cintilação – LUIZ SÉRGIO DE SOUZA, EDITH RANZINI
- BT/PCS/0116 – Simulação de Redes ATM – HSU CHIH WANG CHANG, WILSON VICENTE RUGGIERO
- BT/PCS/0117 – Application of Monoprocessed Architecture for Safety Critical Control Systems – JOSÉ ANTONIO FONSECA, JORGE RADY DE ALMEIDA JR.
- BT/PCS/0118 – WebBee – Um Sistema de Informação via WEB para Pesquisa de Abelhas sem Ferrão – RENATO SOUSA DA CUNHA, ANTONIO MOURA SARAIVA
- BT/PCS/0119 – Parallel Processing Applied to Robot Manipulator Trajectory Planning – DENIS HAMILTON NOMIYAMA, LÍRIA MATSUMOTO SATO, ANDRÉ RIYUITI HIRAKAWA
- BT/PCS/0120 – Utilização de Padrão de Arquitetura de Software para a Fase de Projeto Orientado a Objetos – CRISITINA MARIA FERREIRA DA SILVA, SELMA SHIN SHIMIZU MELNIKOFF
- BT/PCS/0121 – Agilizando Aprendizagem por Reforço Através do uso de Conhecimento sobre o Domínio – RENÉ PEGORARO, ANNA H. REALI COSTA
- BT/PCS/0122 – Modelo de Segurança da Linguagem Java Problemas e Soluções – CLAUDIO MASSANORI MATAYOSHI, WILSON VICENTE RUGGIERO
- BT/PCS/0123 – Proposta de um Agente CNM para o Gerenciamento Web de um Backbone ATM – FERNANDO FROTA REDÍGOLO, TEREZA CRISTINA MELO DE BRITO CARVALHO
- BT/PCS/0124 – Um Método de Teste de software Baseado em Casos Teste – SÉRGIO RICARDO ROTTA, KECHI HIRAMA
- BT/PCS/0201 – A Teoria Nebulosa Aplicada a uma Bicicleta Ergométrica para Fisioterapia – MARCO ANTONIO GARMS, MARCO TÚLIO CARVALHO DE ANDRADE
- BT/PCS/0202 – Synchronization Constraints in a Concurrent Object Oriented Programming Model – LAÍS DO NASCIMENTO SALVADOR, LÍRIA MATSUMOTO SATO

- BT/PCS/0203 – Construção de um Ambiente de Dados sobre um Sistema de Arquivos Paralelos – JOSÉ CRAVEIRO DA COSTA NETO, LIRIA MATSUMOTO SATO
- BT/PCS/0204 – Maestro: Um Middleware para Suporte a Aplicações Distribuídas Baseadas em Componentes de Software – CLÁUDIO LUÍS PEREIRA FERREIRA, JORGE LUÍS RISCO BECERRA
- BT/PCS/0205 – Sistemas de Automação dos Transportes (ITS) Descritos Através das Técnicas de Modelagem RM-OPD (ITU-T) e UML (OMG) – CLÁUDIO LUIZ MARTE, JORGE LUÍS RISCO BECERRA, JOSÉ SIDNEI COLOMBO
- BT/PCS/0206 – Comparação de Perfis de Usuários Coletados Através do Agente de Interface PersonalSearcher – GUSTAVO A. GIMÉNEZ LUGO, ANALÍA AMANDI, JAIME SIMÃO SICHMAN
- BT/PCS/0207 – Arquitetura Reutilizáveis para a Criação de Sistemas de Tutorização Inteligentes – MARCO ANTONIO FURLAN DE SOUZA, MARIA ALICE GRIGAS VARELLA FERREIRA
- BT/PCS/0208 – Análise e Predição de Desempenho de Programas Paralelos em Redes de Estações de Trabalho – LIN KUAN CHING, LIRIA MATSUMOTO SATO
- BT/PCS/0209 – Previsões Financeiras Através de Sistemas Neuronebulosos – DANIEL DE SOUZA GOMES, MARCO TÚLIO CARVALHO DE ANDRADE
- BT/PCS/0210 – Proposta de Arquitetura Aberta de Central de Atendimento – ANA PAULA GONÇALVES SERRA, MOACYR MARTUCCI JÚNIOR
- BT/PCS/0211 – Alternativas de Implementação de Sistemas Nebulosos em Hardware – MARCOS ALVES PREDEBON, MARCO TÚLIO CARVALHO DE ANDRADE
- BT/PCS/0212 – Registro de Imagens de Documentos Antigos – VALGUIMA VICTORIA VIANA ODAKURA MARTINEZ, GERALDO LINO DE CAMPOS
- BT/PCS/0213 – Um Modelo de Dados Multidimensional – PEDRO WILLEMSSENS, JORGE RADY DE ALMEIDA JUNIOR
- BT/PCS/0214 – Autômatos Adaptativos no Tratamento Sintático de Linguagem Natural – CÉLIA YUMI OKANO TANIWAKI, JOÃO JOSÉ NETO
- BT/PCS/0215 – Fatores e Subfatores para Avaliação da Segurança em Software de Sistemas Críticos – JOÃO EDUARDO PROENÇA PÁSCOA, JOÃO BATISTA CAMARGO JÚNIOR
- BT/PCS/0216 – Derivando um Modelo de Projeto a Partir de um Modelo de Análise, com Base em Design Patterns J2EE – SERGIO MARTINS FERNANDES, SELMA SHIN SHIMIZU MELNIKOFF
- BT/PCS/0217 – Domínios Virtuais para Redes Móveis Ad Hoc: Um Mecanismo de Segurança – LEONARDO AUGUSTO MARTUCCI, TEREZA CRISTINA DE MELO BRITO CARVALHO
- BT/PCS/0218 – Uma Ferramenta para a Formulação de Consultas Baseadas em Entidades e Papéis – ANDRÉ ROBERTO DORETO SANTOS, EDIT GRASSIANI LINO CAMPOS
- BT/PCS/0219 – Avaliação de Performance de Arquiteturas para Computação de Alto Desempenho – KARIN STRAUSS, WILSON VICENTE RUGGIERO
- BT/PCS/0220 – BGLsim: Simulador de Sistema Completo para o Blue Gene/L – LUÍS HENRIQUE DE BARROS CEZE, WILSON VICENTE RUGGIERO
- BT/PCS/0221 – μ P: Uma Solução de Micropagamentos – PEDRO ANCONA LOPEZ MINDLIN, TEREZA CRISTINA MELO DE BRITO CARVALHO