

# Análise Experimental de Equidade em Roteador com Diferenciação *Stateless* de Fluxos

Salim S. Mussi e Moisés R. N. Ribeiro

**Resumo**—Propomos uma metodologia experimental para análise de equidade em roteadores com política de diferenciação de fluxos longos e curtos. A segregação de fluxos analisada (RuN2C) encaminha gradativamente fluxos que tornam-se volumosos para uma segunda fila, proporcionando uma distribuição mais equânime de banda via técnicas de escalonamento. Sob tráfego real, num ambiente controlado, resultados indicam o uso do escalonamento *Round Robin* pela redução em até 34% do tempo médio de transferência de arquivos pequenos além da diminuição de sua variabilidade em 87,94%. Não há prejuízos dignos de nota às longas conexões, uma vez que os resultados se aproximam do referencial analítico de equidade obtido pelo *Processor Sharing*.

**Palavras-Chave**—QoS, equidade, diferenciação de fluxos, escalonamento

**Abstract**—We propose an experimental methodology to analyse bandwidth fairness in routers with long and short flow differentiation policy. The segregation of flows, in our case, takes the long-lasting connections to a second queue. By the use of scheduling techniques, bandwidth is distributed among these queues. Under real traffic, using controlled tests, this approach indicates that the use of Round Robin scheduling is able to reduce in 34% the average transfer time and 87.94% on the standard deviation for small files transfers. There are no noticeable side-effects for long-lasting connections and performance meets the analytical fairness benchmark provided by Processor Sharing.

**Keywords**—QoS, fairness, flow differentiation, scheduling

## I. INTRODUÇÃO

A crescente exigência, por parte das aplicações, de qualidade de serviço, contribuem para a complexidade do processamento dos pacotes uma vez que perdas e atrasos devem ser controladas num cenário atual de maior interatividade. Portanto, políticas mais eficientes são necessárias para adequar estas diferentes taxas de crescimento de tráfego e da capacidade de processamento. Neste sentido, esforços estão sendo direcionados para o desenvolvimento de técnicas que possibilitem, a partir de um baixo processamento eletrônico, atender a esta demanda. Uma destas técnicas é a priorização de fluxos curtos sem manutenção de estados em roteadores.

O cenário de tráfego da Internet é difícil de ser bem caracterizado analiticamente. A alta variabilidade do tamanho das transferências de arquivos, o uso de um protocolo extremamente elástico (i.e., *Transmission Control Protocol*, TCP) que se adapta às diversas condições da rede e o surgimento constante de novos serviços são apenas alguns dos fatores complicadores na tarefa de modelagem desta rede. Todavia, um fato evidente é que a maior parte das interações dos

usuários com esta rede é de curta duração, em parte pelo predomínio da interação dos navegadores com *Web pages*. Seguindo a tendência do aumento das conexões curtas, os aplicativos *Peer-to-Peer* subdividem grandes transferências num conjunto de pequenos arquivos os quais se dispersam na rede, em virtude da diversidade de origens. Apesar da grande quantidade, as conexões curtas ainda são responsáveis por uma pequena porção da carga total dos enlaces e ainda disputam, injustamente, recursos com conexões de altíssima duração resultantes da transferência de grandes arquivos de uma única origem. Assim, esforços estão sendo direcionados para que se desenvolvam técnicas de baixa complexidade que tentem diminuir o tempo de transferência de pequenos arquivos e que promovam, portanto, uma alocação mais justa dos recursos disponíveis sem sobrecarregar o processador e memória do roteador. Entende-se como uma alocação justa de recursos quando os atrasos experimentados pelas sessões são apenas proporcionais ao volume de dados a serem transferidos. Em outras palavras, que as sessões com baixo volume de tráfego, que eventualmente não passam da fase de *slow start* do TCP, não sejam influenciadas pelas grandes rajadas características dos fluxos de longa duração já operando na fase de controle de congestionamento.

Neste trabalho apresentamos metodologia experimental baseada em *Processor Sharing*, PS, para análise da equidade entre fluxos no acesso a banda. Implementamos o mecanismo *Running Number 2 Class* (RuN2C) para diferenciação e priorização de fluxos curtos, em roteadores sobre hardware de PC, utilizando a arquitetura modular *CLICK Router* [2] (roteadores em PC) analisada sob tráfego real FTP, *File Transfer Protocol*. O RuN2C foi escolhido devido ao baixo requisito de processamento e memória, além da alta compatibilidade com roteadores que não possuem esta funcionalidade. Comparamos ainda, o desempenho do escalonamento por prioridade estrita, originalmente proposto em [1], e sob *Round Robin* para servir as duas filas utilizadas pelo RuN2C. O restante do trabalho é dividido da seguinte forma. Algumas publicações diretamente relacionadas são brevemente discutidas na próxima Seção. Uma introdução a roteadores em PC é apresentada na Seção III. Os aspectos do RuN2C e o esquema de implementação utilizado são introduzidos na Seção IV. Na Seção V a metodologia de testes e a formulação do padrão de equidade são apresentadas. Os experimentos realizados e resultados obtidos ficam na Seção VI seguido das Conclusões e comentários finais na Seção VII.

## II. TRABALHOS RELACIONADOS E CONTRIBUIÇÕES

Há uma infinidade de métodos de diferenciação e priorização de fluxos na literatura. Em [5] são discutidas duas

Salim S. Mussi e Moisés R. N. Ribeiro Laboratório de Telecomunicações, Departamento de Engenharia Elétrica, Universidade Federal do Espírito Santo, Av. Fernando Ferrari, N° 514, 29075-910, Vitória-ES, Brasil, E-mails: salim-suhet@gmail.com, moises@ele.ufes.br

possíveis soluções. A primeira é baseada na diferenciação de serviços (DiffServ), utilizando *Assured Forwarding* (AF) e *RED with In and Out* (RIO). Esta solução requer a escolha não trivial de inúmeros parâmetros para os mecanismos AF e RIO. Outra solução apontada é um mecanismo com manutenção de estados dos fluxos TCP, que utiliza do tamanho das janelas de cada sessão e conta com a observância dos *hosts* de destino. Desta vez, a solução requer o *tuning* de parâmetros para o *Weighted Round Robin* (WRR) que novamente requer uma escolhas não evidentes e conseqüentemente se trata de outra solução não robusta.

Em [8] é proposto um método de diferenciação de fluxos através da contabilização dos *bytes* transmitidos em cada sessão denominado *Deficit Round Robin with Short Flow First* (DRR-SFF). Para isto, é necessária a manutenção da quintupla de caracterização de cada fluxo (IP e porta de origem e destino, além do protocolo) em uma tabela, além do total de *bytes* transmitidos. Os pacotes de cada fluxo são divididos através de um limiar em duas filas distintas escalonadas via *Weighted Deficit Round Robin* (DRR). Mostram-se resultados positivos, todavia, somente através de simulações numéricas, na redução do tempo médio de transmissão dos fluxos em relação ao SSF, que utiliza prioridade estrita para a fila de fluxos curtos. Porém, tal mecanismo demanda muita memória para armazenamento dos estados e elevado processamento diferenciação dos fluxos sendo passível de implementação apenas em roteadores de borda.

Em [11] é proposto um protocolo de transporte com controle de congestionamento em roteadores, alternativo ao TCP, denominado *Rate Control Protocol* (RCP). Ele visa proporcionar o menor tempo possível de finalização para os fluxos ativos. O RCP é uma emulação da divisão ideal de Processor Sharing (PS), onde o roteador tenta dividir igualmente os recursos entre os fluxos ativos propiciando ao usuário de aplicações mais interativas com trocas de dados de pequenos volumes seja melhor atendido num ambiente com competição com fluxos que intercibiam grandes arquivos. Esta política possui a vantagem de a duração dos fluxos ser invariante à distribuição granulométrica dos fluxos e de ser intrinsecamente justa. Contudo, a sua incompatibilidade com o TCP torna custosa a sua implementação em larga escala.

Em [1] o mecanismo sem manutenção de estados *Running Number 2 Class* (RuN2C) é proposto. Tal mecanismo consiste na divisão entre fluxos curtos e longos em filas distintas utilizando a informação de números de *bytes* transferidos dos números de sequência dos pacotes TCP para comparação com um limiar de decisão. Uma política de prioridade estrita é adotada para o escalonamento destas filas. Nesta política a fila de fluxos longos não é servida até que a de fluxos curtos esteja vazia. Este método pode ser implementado em diferentes roteadores, tanto em redes de acesso como em redes de *backbone*, por não ser necessária a manutenção de estados dos fluxos (i.e., *stateless*). Além disto, a implantação pode se dar de maneira gradual não necessitando de prévio acordo entre as operadoras. Entretanto, somente resultados de simulação (em NS2) e com tráfego sintético são apresentados para demonstrar a validade da técnica.

No presente trabalho é apresentado, no conhecimento dos

autores, a primeira implementação física de um roteador com o mecanismo RuN2C. Trazemos ainda a contribuição da investigação do seu desempenho sob diferentes técnicas de escalonamento. Do ponto de vista de metodologia de teste, a contribuição está nos experimentos em ambiente controlado, para possibilitar comparações entre as técnicas de escalonamento, com tráfego real oriundo de um *backup* de um *hard-disk* via *File Transfer Protocol* (FTP). Ao contrário das conclusões do artigo original [1] onde o escalonamento com prioridade estrita é apontado como o de melhor desempenho, os nossos resultados experimentais apontam o *Round Robin* como a melhor estratégia de equidade no escalonamento. Assim como no RCP, nossa implementação experimental do RuN2C apresentou resultados bastante próximos ao ideal PS, porém sem exigir alterações significativas do TCP requeridas pelo RCP.

### III. ROTEADORES EM PC

Há apenas algumas opções de roteadores baseados em PC, com código aberto, derivados do esquema de encaminhamento por *kernel* do *Berkeley Software Distribution* (BSD). Os mais completos no quesito de protocolos de roteamento são o XORP [3] e o Quagga [4]. Todavia, vamos optar pela tecnologia CLICK (hoje incorporado à plataforma XORP) a qual se baseia em uma arquitetura flexível e modular de software para criar e encaminhar pacotes, facilitando a implementação de novas técnicas de tratamento dos pacotes. Diversos aspectos da arquitetura CLICK foram inspirados diretamente por propriedades dos roteadores [2]. Um exemplo disto é que a passagem de pacotes ao longo de uma conexão pode ser iniciada tanto pela extremidade da fonte (pacote é 'empurrado') quanto pela extremidade de destino (pacote é 'puxado'), modelando a maior parte dos padrões de fluxo de pacotes em roteadores.

Há duas formas de se utilizar o software CLICK: como aplicação a nível de usuário (*driver user-level*), ou como módulo do *kernel* Linux (*driver linux-module*). As duas possibilidades se destinam a funções diferentes. O *driver user-level* é útil para depuração e execução de testes repetitivos; ele pode receber pacotes da rede usando mecanismos do sistema operacional através de bibliotecas originalmente programadas para *sniffers*. Entretanto, o *user-level* não se previne da pilha de rede do sistema operacional para encaminhar um pacote. Já o *driver* em nível de *kernel*, o *linux-module*, substitui completamente a pilha de rede do SO, transformando um computador pessoal convencional num roteador, alcançando alto desempenho [2]. Desta forma só se utilizará o último *driver* neste trabalho.

### IV. DIFERENCIAÇÃO DE FLUXOS RUN2C

As aplicações TCP reagem ao congestionamento e às perdas em uma rede de três formas distintas: através da alteração do tamanho de suas janelas (congestionamento e de recepção), de processos rápidos de retransmissão de pacotes ou da retransmissão destes após o *timeout* [6]. Para conexões curtas, o que implica em pequenas janelas de congestionamento, a perda é normalmente detectada após o *timeout* e, provavelmente, após a finalização da transmissão de todos os dados. Como

resultado disto, temos que a retransmissão após o *timeout*, para conexões curtas, não é muito eficaz na redução do tráfego global de uma rede e em sua estabilização.

O mecanismo de diferenciação RuN2C é um escalonador de tráfego TCP/IP que processa duas classes de prioridades. Nele, os pacotes são classificados como baixa ou alta prioridade de acordo com o número de *bytes* transmitidos numa sessão TCP. Quando uma conexão é estabelecida, um número de sequência inicial, ISN (*Initial Sequence Number*), de 32 bits é gerado de maneira aleatória. A cada pacote transmitido é somado a este ISN o número de Bytes transferidos. Isto possibilita a diferenciação dos fluxos ativos no roteador, através deste campo do cabeçalho TCP, apenas com o controle da geração destes números iniciais nas máquinas fontes de tráfego e da comparação deles com um limiar nos roteadores.

**A. Número de Sequência e Determinação dos Limiares**

Após a transferência do primeiro pacote um número de sequência inicial é incrementado de um número máximo de *bytes* igual ao MSS (*Maximum Segment Size*) da rede. Para um MSS igual a 1460 *bytes*, temos que, aproximadamente os 10 bits menos significativos,  $\log_2(MSS = 1460) \approx 10$ , do número de sequência seriam utilizados logo no primeiro pacote. Assim, estes bits poderiam ser utilizados, sem prejuízos ao limiar, para geração de um ISN. No entanto, 10 bits gerariam apenas 1024 possíveis ISNs, não provendo aleatoriedade suficiente a estes números, possibilitando problemas de segurança, tais como IP *spoofing* e sessão *hijacking* [8].

Desta forma, a divisão do número de sequência é feita de acordo com a Figura 1. Utiliza-se além dos 10 bits menos significativos, os 10 bits mais significativos, campo R, para a geração dos ISN, totalizando em  $2^{20} \approx 1$  milhão de possíveis números iniciais.

O limiar de separação, TH, entre as duas classes de prioridade deve ser escolhido de forma que os fluxos curtos se beneficiem do mecanismo de diferenciação, todavia, mantendo-se baixa a carga de alta prioridade para que fluxos longos não sejam prejudicados demasiadamente.

Para uma solução que satisfaça este compromisso é importante ressaltar que os fluxos TCP curtos são altamente propensos a *timeouts* após a perda de um pacote. O impacto disto é extremamente relevante para o tempo de resposta deste tipo de fluxos, uma vez que o tempo mínimo para *timeout* é de 1 segundo [6]. Para evitar este impacto, deve-se dar prioridade a estes fluxos antes que a janela de congestionamento do TCP atinja 3 ou 4 pacotes, o que ocorre, por exemplo, se são transmitidos 8 pacotes. Isto totaliza um limiar de aproximadamente 12KBytes, para um MSS igual a 1460 KBytes.

Outro ponto é que se os fluxos TCP na Internet são descritos por uma distribuição cauda-pesada [9][10], fluxos abaixo de 8 pacotes representam um significativo número do total dos fluxos TCP, mas uma pequena porção da carga total. Assim, ao se dar prioridade a estes fluxos, atendemos ao requisito de não haver prejuízo digno de nota aos longos fluxos.

A decisão do limiar, TH, ideal é amplamente discutida em [1]. Limitando-o em 12 KB, como neste artigo de referência, são necessários os  $TH = \log_2(12 \times 1024) \approx 14$  bits menos

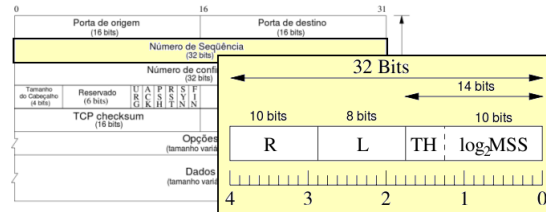


Fig. 1. Estrutura do número de sequência TCP para o RuN2C

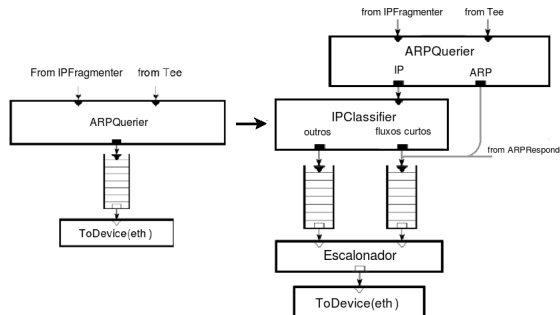


Fig. 2. Alteração implementada no roteador básico

significativos do campo do número de sequência. Sendo R, o seguimento de 10 bits utilizado na geração de um ISN, o segmento L será os 8 bits seguintes ao TH. Em nosso experimentos testamos outros diferentes limiares, 100KB e 1000KB, e os resultados destes também indicam aquela como a escolha mais eficiente.

**B. Implementação de Roteador RuN2C com Escalonador sobre o CLICK**

Para implementação de um roteador que priorize as conexões curtas no CLICK, foi adicionada uma segunda porta de saída ao elemento CLICK *ARPQuerier*, conforme mostrado na Figura 2. Quando esta saída é adicionada, os pacotes IP são encaminhados à primeira saída, e pacotes do protocolo ARP à outra, possibilitando inclusão dos pacotes do último tipo na mesma fila com prioridades dos pacotes TCP de fluxos curtos. Isto proporciona ganhos, uma vez que enquanto não são obtidas as respostas das requisições ARP os pacotes não são encaminhados.

Para a classificação dos pacotes IP incluiu-se o elemento *IPClassifier* que pode ser configurado para separar pacotes do protocolo TCP de acordo com o número de sequência. Este elemento utiliza o mesmo padrão do *software* Linux *TCP-Dump*. Nesta configuração, ao número de sequência, situado nos *bytes* de 5 a 8 do cabeçalho TCP, aplica-se uma máscara ocultando os 10 bits do campo R, mostrado anteriormente na Figura 2. Em seguida compara-se o resultado com o limiar de 14 bits. A string de configuração é “tcp[4:4] & 0xFFC00000 < 0x00004000, -”. Os pacotes que obedecem à configuração são destinados a uma interface e os demais à outra.

As filas são finalmente ligadas a um escalonador de prioridades que é encarregado de encaminhar os pacotes à interface. Serão utilizados neste trabalho dois elementos CLICK de escalonamento: *PrioSched* e *RoundRobinSched*. O *PrioSched* funciona como o esquema de prioridades proposto para o

Run2C, a fila de fluxos longos não é servida até que a de fluxos curtos esteja vazia. Já o *RoundRobinSched* faz o escalonamento do tráfego transmitindo seqüencialmente um pacote de cada entrada. Se uma entrada acabou de ser servida, o escalonador busca o próximo pacote na fila da entrada seguinte. *RoundRobinSched* utiliza-se de um mecanismo de notificação para não servir filas vazias.

### V. METODOLOGIA DE TESTES

Na literatura disponível é corriqueiro o uso de simuladores para a comparação de eficiência de diferentes métodos de tratamento de fluxos. Todavia, o processo de geração de tráfego assim como o processo de tratamento deste se baseiam em modelos muitas vezes demasiadamente simplificados. Portanto, os resultados de simulação frequentemente não representam de maneira fidedigna o comportamento dos sistemas físicos. Desta forma, foi objetivo neste trabalho a submissão dos roteadores implementados a um cenário de tráfego real no qual pudéssemos ter algum grau de controle nos testes. Para isto, optou-se pela transferência simultânea, gerando concorrência entre os fluxos, de uma base real de arquivos via FTP. O objetivo principal é avaliar o efeito sobre sessões TCP, daí para podermos isolar o efeito da política de diferenciação de fluxos devemos trabalhar com conexões com apenas um salto.

No entanto, ao se transmitir múltiplos arquivos simultaneamente a um servidor, é possível que este não consiga tratar todo o tráfego recebido, levando ao enfileiramento destes datagramas recebidos em um *buffer* por cada sessão no servidor. Quando isto ocorre, o servidor diminui o tamanho da janela de recepção anunciada, resultando na redução do tráfego que o emissor irá injetar na conexão. Se este *buffer* se esgota, o tamanho da janela é anunciado como zero, voltando a subir apenas quando o servidor tratar os pacotes recebidos [7].

Para avaliarmos o desempenho de cada política de escalonamento de tráfego temos que garantir, então, que o gargalo seja o roteador e não o servidor FTP. Assim, adicionou-se antes de cada elemento de saída *toDevice* do roteador um elemento limitador de banda chamado *BandwidthShaper*, configurado para taxa de 1Mbps. Além disso, para compararmos os mecanismos com duas filas a outro com uma única fila evitaremos possíveis descartes de pacotes, tornando estas filas grandes o suficiente. O tamanho adotado foi de 10000 pacotes.

Para todos os experimentos foram utilizadas três máquinas conectadas diretamente umas as outras com cabos cruzados, como mostrado na Figura 3, evitando-se assim, a interferência nas medidas por outros equipamentos de rede. As máquinas têm a seguinte configuração de hardware: processador Intel Pentium III, 933MHz, NIC PCI 10/100 com clock 33MHz, e 512MBytes de memória RAM 133MHz. Apenas na máquina roteadora utilizou-se memória de 768MBytes.

#### A. Testes de Vazão do Roteador CLICK

Para aferir o desempenho de um roteador implementado com a tecnologia CLICK, apontado na literatura como possuidor de uma capacidade de encaminhamento de pacotes superior a dos roteadores comerciais Cisco da série 7200 [2], utilizou-se de dois métodos de medidas. Um destes métodos,

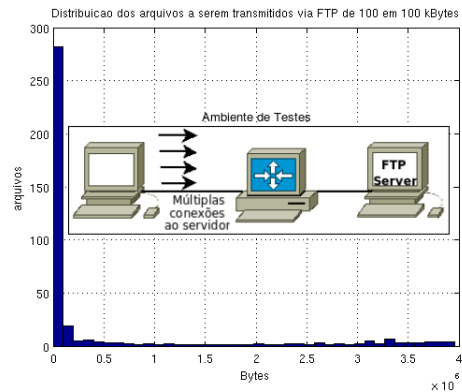


Fig. 3. Ambiente de testes e distribuição dos arquivos transmitido simultaneamente ao servidor FTP

interno ao roteador CLICK, utiliza das informações de *bytes* transmitidos providas pelos *handlers* dos elementos para gerar um *log* em tempo de execução. A partir deste *log* pode-se calcular a taxa média de transferência. O outro método foi a utilização do Iperf, renomado *software* das distribuições Linux para medição da banda disponível em enlaces, que também foi utilizado como fonte de tráfego. Os resultados destes testes mostram a coerência das medidas pelos dois métodos, alcançando taxas máximas de 72.8 Mbps. Também é importante ressaltar o funcionamento adequado do elemento *BandWidthShaper* limitando a banda em 1Mbps como desejado.

#### B. Base de Arquivos Transmitida

Visando obter uma base consistente de arquivos com a qual se pudesse estudar um procedimento de *backup* de disco, realizou-se uma amostragem de 381 arquivos na faixa de 1KB até 4MB oriundos da listagem de todos os arquivos de um computador pessoal, resultando na distribuição mostrada na Figura 3. O tamanho da base é limitado ao número máximo de *threads* concomitantes de transferência FTP conseguido experimentalmente através de um *script* em *python* para dar início às transferências. A limitação em 4MB no tamanho dos arquivos pela capacidade de identificação pelos bits tomados, campo L, no número de seqüência.

#### C. Alteração da Geração dos Números de Seqüência no SO

Para que a política de diferenciação de fluxos e escalonamento de tráfego via Run2C funcione adequadamente é necessário que uma adaptação na geração dos números de seqüência iniciais seja realizada. Esta adaptação deve-se dar pelo menos na máquina que envia os arquivos. Para os experimentos, optamos pela alteração nas duas máquinas. O requisito para que o mecanismo Run2C funcione adequadamente é a geração dos 32 bits dos números de seqüência inicial na forma: 20 bits, os 10 bits mais e os 10 bits menos significativos são gerados de maneira aleatória possibilitando um total de 1 milhão de ISNs e os 12 bits restantes devem ser iniciados com zero quando uma conexão TCP é estabelecida. Para isto, alterou-se as funções do *kernel* Linux responsável pela geração destes ISNs. O arquivo alterado foi o *Random.c*, localizado dentro do *kernel* em *linux<versão>/drivers/char/*.

#### D. Padrão de Equidade

Propomos como padrão de equidade uma modelagem analítica da política PS que distribuirá os recursos igualmente entre todos os fluxos ativos,  $NF_i$ . Na medida que os fluxos são finalizados, a banda,  $B$ , disponível no roteador é redistribuída entre os fluxos que permanecem. Considerando  $A$  um vetor de  $n$  posições contendo em cada uma o número de arquivos de uma faixa de um histograma de granularidade  $G$ , o número de fluxos ativos será  $NF_i = \sum_{j=i}^n A(j)$ .

O tempo médio,  $T_i$  de transmissão dos arquivos da  $i$ -ésima faixa do histograma é calculado com base na média dos tamanhos dos arquivos desta faixa. Também é considerado, para tal cálculo, que a banda igualmente distribuída disponível para cada arquivo é  $B_E(i) = B/NF_i$ . Portanto, o desvio padrão segundo a equidade gerada pelo PS deve ser igual a zero uma vez que todos os fluxos de uma mesma faixa terminariam exatamente no mesmo instante.

$$T_1 = \frac{\frac{G}{2} \cdot 1024 \cdot 8}{B_E(1)} \quad (1)$$

$$T_i = T_{i-1} + \frac{G \cdot 1024 \cdot 8}{B_E(i)} \quad (2)$$

### VI. EXPERIMENTOS E RESULTADOS

Nesta seção apresentaremos os resultados de número de fluxos ativos em relação ao tempo, tempo médio de finalização destes fluxos e seu respectivo desvio padrão, para o roteador limitado à taxa de 1Mbps, prevenindo assim que limitações do servidor FTP afetem nossos resultados, e aumentando os buffers dos roteadores suficientemente para que se evite descartes na entrada destes. Utilizou-se um limiar de 14 bits, totalizando 12KB, para diferenciação entre os fluxos curtos e longos. Outros experimentos, tais como variação deste limiar de decisão e a não limitação da banda nos roteadores são também comentados.

#### A. Análise Temporal dos Fluxos Transmitidos

Na Figura 4 temos o número de fluxos ativos para as políticas de fila única e sem nenhuma diferenciação entre fluxos (conhecida como *DropTail*), RuN2C com escalonamento de prioridade estrita à fila de fluxos curtos (*PrioSched*) e RuN2C com escalonamento *Round Robin*. Apresentamos ainda o padrão de referência analítico,  $NF_i(t)$ , para o caso ideal de divisão equânime de banda entre os fluxos.

Nota-se que em  $t = 0 s$  são iniciadas simultaneamente as 381 sessões FTP. Até  $t = 200 s$  o RuN2C possibilita a finalização de algumas sessões em tempo menor que o indicado pelo padrão de equidade, indicando que algum privilégio foi dado a tais sessões em termos de divisão de banda como esperado. Todavia, contrariando a intuição, de que o escalonamento com prioridade estrita seria a melhor forma de oferecer este privilégio, a política *Round Robin* apresenta mais fluxos finalizados em menos tempo, enquanto não nota-se diferença do *Drop Tail* e *PrioSched* para esta faixa de tempo de transiência. Em um segundo momento, a partir de  $t = 300 s$ , a mecanismo RuN2C se aproxima do número de finalizações de

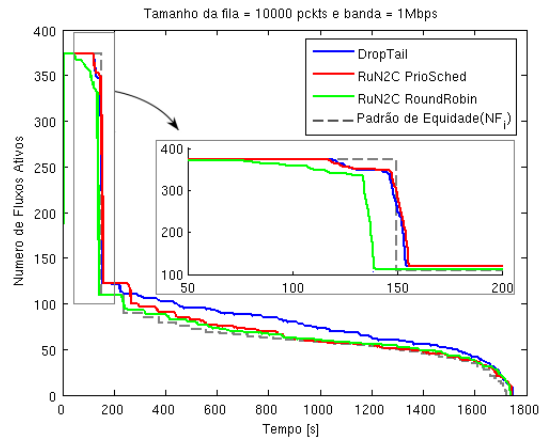


Fig. 4. Comparação do término dos fluxos nos roteadores

fluxos dado pelo padrão. Em um último momento, percebe-se que a finalização global dos fluxos se dá em torno de  $t = 1800 s$  liderado pelo padrão de equidade, uma vez que os privilégios dados no início das sessões pelo RuN2C resultam num pequeno aumento dos tempos de finalização. Apesar dos resultados da Figura 4 já indicarem alguns benefícios do RuN2C em relação ao *Drop Tail*, uma análise mais detalhada é necessária em termos do tratamento dado às sessões em função do tamanho dos arquivos por elas transferidos.

Na Figura 5 apresentamos os tempos médios para finalização dos fluxos contidos em cada faixa do histograma da base de arquivos. Percebe-se que uma alta variabilidade dos tempos médios entre faixas vizinhas para o *Drop Tail*, enquanto um comportamento bastante regular é apresentado pelos esquemas de diferenciação baseados no RuN2C. Mais ainda, quando comparados com o padrão de equidade,  $T_i$ , o RuN2C com escalonamento *Round Robin* destaca-se por aproximar-se, em todas as faixas, do comportamento ideal que um sistema PS propiciaria aos fluxos. A priorização dos fluxos curtos surte efeito principalmente na faixa que contém os arquivos menores que 12KB (i.e., primeira faixa do histograma). Neste caso, devido ao regime de privilégio dos fluxos curtos, os roteadores com RuN2C apresentam tempos médios abaixo do padrão de equidade, destacando-se mais uma vez o escalonamento *Round Robin*. O tempo médio de transferência de arquivos pequenos para a política *Drop Tail*, RuN2C *PrioSched* e RuN2C *Round Robin* são, respectivamente, 201.21, 154.54 e 132.38 segundos, totalizando um ganho em relação ao *Drop Tail* de 23.2% quando se utiliza o *PrioSched* e 34.2% quando o *Round Robin* é empregado.

É importante salientar que o padrão de equidade é atingido pelo RuN2C, o que o aproxima do RCP em termos de desempenho. Todavia, o RCP requer alteração completa do protocolo de transporte, enquanto o RuN2C apenas requer a alteração do processo de geração do número de sequência do TCP na fonte do tráfego. Cabe agora examinar a diferença de tempos de finalização dentro de cada faixa, tomando como referência os tempos médios por faixa mostrados na Figura 5, ou seja, comparar os desvios padrão por faixa. Na Figura 6 percebe-se a maior frequência de altas variabilidades

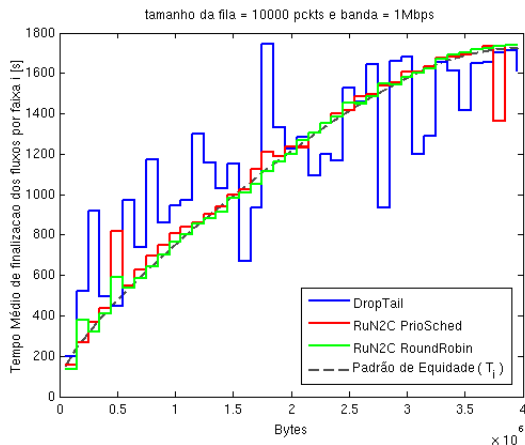


Fig. 5. Comparação dos tempos médios de transferência de arquivos

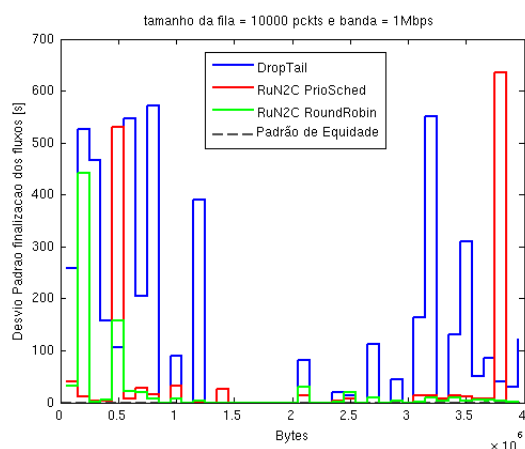


Fig. 6. Desvios padrão dos tempos de transferência de arquivo

para os arquivos atendidos pela política *Drop Tail*. Porém, quando se analisa a primeira faixa, nota-se claramente que a priorização dos fluxos curtos diminui a competitividade destes com os longos, permitindo, inclusive, seus terminos em tempo bastante próximos, levando o desvio padrão a se aproximar de zero, semo como no caso do PS. Quantitativamente, o desvios padrão para esta faixa é de 264.39, 39.74 e 31.89 segundos, para as filas *Drop Tail*, Run2C com *PrioSched* e Run2C com *Round Robin*, respectivamente. Isto representa uma redução de 84.7% entre os dois primeiros mecanismos e de 87.94% do primeiro para o terceiro.

### B. Variações do Experimento

Realizamos também experimentos com os roteadores operando sem limitação de banda. A vazão obtida neste caso foi de aproximadamente 70Mbps. Constatou-se a hipótese que os mecanismos de diferenciação operam bem em regime de recursos escassos. Uma vez que a limitação de banda é removida, verifica-se o distanciamento das métricas realizadas do padrão de equidade. Todavia, um indicativo de melhora do sistema ainda permanece, a diminuição dos tempos médios e desvio padrão de finalização dos fluxos curtos quando os mecanismos de diferenciação e priorização de tráfego

são implementados. Estes ganhos consistem na diminuição dos tempos médios de finalização em 2.27% e 4.73% para Run2C com *PrioSched* e *RoundRobin* respectivamente, além da diminuição do desvio padrão em 26.85% e 37.96%. Sob banda limitada, variações do limiar de divisão entre fluxos longos e curtos, TH, para pacotes de início de sessão, 100KB e 1000KB foram estudadas. Resultados apontam que o valor de TH=12KB utilizado na seção anterior deve estar próximo do ponto ótimo para a distribuição de arquivos empregada, de acordo com os resultados analíticos de segregação ótima apresentados em [10].

## VII. CONCLUSÕES

Neste trabalho foi apresentada uma metodologia experimental que utiliza o conceito de *Processor Sharing* para análise da equidade entre fluxos no acesso à banda. Testamos o mecanismo de diferenciação e priorização de fluxos curtos Run2C, porém em roteadores físicos. Comparamos seu desempenho em relação a uma única fila e ao padrão de equidade proposto num cenário de tráfego real. O mecanismo Run2C faz com que o tempo médio de finalização dos fluxos se aproxime do referencial de equidade provendo justiça na alocação de recursos em regime de banda escassa. Os ganhos giram em torno de 23% quando se utiliza o Run2C com priorização das filas pelo escalonador *PrioSched* e aproximadamente 34% quando o escalonador é o *Round Robin*, quando o limiar é de 12 KBytes. Estes ganhos são resultado da facilitação do início de novas sessões acelerando o *slow start* reduz o atraso de fila para fluxos genuinamente curtos. Os efeitos da incrementação deste o limiar TH faz com que as políticas de escalonamento se aproximem da fila política *Drop Tail* pois as métricas distanciam-se do padrão de equidade. Em trabalhos futuros, testes com tráfego de diferentes características e alternativas à alteração do número de sequência serão estudados.

## REFERÊNCIAS

- [1] K. Avrachenkov, U. Ayesta, P. Brown, E. Nyberg, "Differentiation between short and long TCP flows: Predictability of the response time", in Proceedings of INFOCOM, Hong Kong, 2004.
- [2] E. Kohler, "The click modular router". Ph.D. Thesis, MIT, agosto, 2000.
- [3] M. Handley, O. Hodson, E. Kohler, "XORP: An Open Platform for Network Research" Proc. 1st Wksp. Hot Topics in Networks, Princeton, NJ, 2002.
- [4] Quagga, *Quagga Routing Suite*, <http://www.quagga.net>, 2008.
- [5] W. Noureddine, F. Tobagi, "Improving the performance of interactive TCP applications using service differentiation", in Proceedings of IEEE INFOCOM, New York, NY, 2002.
- [6] V. Paxson, M. Allman, "Computing TCP's Retransmission Time". RFC2988, novembro, 2000.
- [7] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms". RFC2001, janeiro, 1997.
- [8] C. Sun, L. Shi, C. Hu e B. Liu, "DRR-SFF: A Practical Scheduling algorithm to Improve the Performance of Short Flows", Third International Conference on Networking and Service, IEEE, 2007.
- [9] K. Thompson, G. J. Miller, R. Wilder, "ide-area Internet traffic patterns and characteristics", IEEE Network, vol. 11, no. 6, 1997.
- [10] F. L. Trazzi Jr., M. R. N. Ribeiro, H. Waldman, "On the benefits of traffic segregation in WDM networks". In: XXIII Simpósio Brasileiro de Rede de Computadores, 2005, Fortaleza. XXIII Simpósio Brasileiro de Rede de Computadores, 2005. v. 2. p. 869-882.
- [11] N. Dukkipati, "Rate Control Protocol (RCP): Congestion control to make flows complete quickly". Ph.D. Thesis, Stanford University, outubro, 2007.