RAFLE: Uma Proposta para Diferenciação de Fluxos sem Manutenção de Estados em Roteadores

Salim S. Mussi[1], **Moisés R. N. Ribeiro**[1]

[1]Departamento de Engenharia Elétrica – Universidade Federal do Espírito Santo (UFES)
Av. Fernando Ferrari, 514 – 29075-910 – Vitória – ES – Brasil

salimsuhet@gmail.com, moises@ele.ufes.br

Abstract. *We propose a stateless flow classifier to provide differentiated service to short and long flows in stateless routers without the need to change current network protocols. Long flow detection is achieved by sampling a short-term memory of forwarded packets. A PC hardware implementation is tested with real FTP traffic in two different situations: hundreds of flows starting simultaneously and constant load of active flows. Round robin and strict priority are employed to schedule separate queues holding packets from the two flow classes. For flows starting simultaneously, the average database transmission time has been decreased by 30% while the number of files transferred under constant load has been improved by 14% in relation to conventional routers.*

Resumo. *Propomos um classificador de fluxos que visa a diferenciação no tratamento de fluxos curtos e longos em roteadores sem modificações dos protocolos de rede e sem manutenção de estados. A detecção de fluxos longos é feita por amostragem dos últimos pacotes encaminhados. Uma implementação em hardware de PC foi testada com tráfego real FTP em duas situações distintas: início simultâneo de centenas de sessões e carga constante de sessões ativas. Round robin e prioridade estrita foram empregados para escalonar as duas filas. Sob transmissão simultânea, a redução no tempo médio de transmissão foi superior a 30%, enquanto o número de arquivos servidos sob carga constante cresceu em 14% em relação aos roteadores convencionais.*

1. Introdução

O tráfego da Internet é dominado por fluxos TCP de curta duração [Labovitz et al. 2009]. Fluxos inferiores a dez pacotes representam 95% do tráfego do cliente para o servidor TCP e 70% do tráfego em sentido oposto [Ciullo et al. 2009]. Todavia, apesar da grande quantidade, os fluxos curtos ainda são responsáveis por uma pequena porção da carga total dos enlaces e ainda disputam, injustamente, recursos com conexões que transportam grandes volumes de dados [Matta 2001].O desempenho de sessões TCP operando em fase de *slow-start* ou em regime de pequenas janelas sofre de forma significativa ao compartilhar *buffers* e capacidade dos enlaces com grandes rajadas oriundas de sessões na fase de controle de congestionamento [Anelli et al. 2011]. Uma forma de amenizar essa desigualdade é tratar diferenciadamente fluxos curtos e longos. Isto diminui o tempo de resposta percebido nas interações de usuário com páginas *web* ou na taxa transferência (i.e. arquivos por segundo) em operações *backups*, uma vez que em ambos os casos há predominância de pequenos arquivos a serem transportados por sessões TCP.

Recentemente, tornou-se popular uma tecnologia de identificação e tratamento de fluxos denominada OpenFlow [McKeown et al. 2008]. Baseia-se em um controlador externo aos *switches* Openflow (responsáveis por encaminhar os pacotes na rede) que centraliza a gerência de encaminhamento de pacotes através de uma visão total e manutenção de estados dos fluxos ativos na rede. Embora quando operando com regras fixas não sejam salientes as suas limitações, quando empregado em redes com muitos elementos e sob demandas muito dinâmicas a melhor implementação conhecida pode iniciar apenas poucas centenas de fluxos por segundo. Uma estimativa deste valor no equipamento desenvolvido pela fabricante HP, o ProCurve 5406z, este número é de 275 fluxos por segundo [Curtis et al. 2011]. Assim, o OpenFlow pode encontrar problemas de escalabilidade no futuro em função da necessidade de manutenção completa de estados (*fullstate*) dos fluxos ativos e da necessidade, a cada novo fluxo, de comunicação com o controlador e seu processamento. Soma-se a isto o fato do contínuo crescimento da capacidade de transmissão por enlaces pesar sobre a capacidade de processamento eletrônico dos pacotes para roteamento ou encaminhamento. Desta forma, iniciativas para o desenvolvimento de técnicas sem manutenção de estados (*stateless*) atingindo um bom compromisso entre eficiência e complexidade são necessárias.

No presente trabalho propomos e implementamos experimentalmente um classificador de fluxos *stateless* que mostrou-se bastante eficiente no desempenho global no tratamento de fluxos longos e curtos para uma aplicação real de *backup* de disco. O restante do artigo será dividido da seguinte forma. Na Seção 2, trabalhos relacionados são brevemente discutidos, onde também destacamos as contribuições em relação as propostas anteriores. Na Seção 3 é apresentada a metodologia de testes e de comparação com esquemas alternativos. Os aspectos do esquema RAFLE são abordados na Seção 4. Os experimentos realizados e seus resultados ficam nas Seções 5 e 6, sendo seguidos das conclusões e comentários finais.

2. Trabalhos Relacionados e Contribuições

Em [Avrachenkov et al. 2004] 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 (TH). Uma política de prioridade estrita é adotada para o escalonamento destas filas. Embora exija mudanças menos drásticas como a troca total do protocolo de transporte sugerida pelo RCP (*Rate Control Protocol*) [Dukkipati 2007], há ainda a necessidade de controle na geração dos números de sequência iniciais, reduzindo o número de bits aleatórios, o que além de exigir alteração da implementação do TCP pode facilitar ataques do tipo *session hijack*.

Propostas recentes como o DevoFlow [Curtis et al. 2011] ainda estão sendo avaliadas no objetivo de reduzir o impacto do processo de manutenção de estados da proposta original do OpenFlow. As principais alterações estão no sentido de diminuir as interações *switch*-controlador introduzindo mecanismos que permitam aos *switches* tomarem decisões locais de encaminhamento de fluxos sem que haja habilitação do controlador. No entanto, ainda há a necessidade de que os *switches* se comuniquem eventualmente com o controlador no início de um novo fluxo. Isto adiciona atraso durante o ingresso de um fluxo em uma nuvem OpenFlow [Curtis et al. 2011]. Esta introdução de latência,

mesmo quando da ordem de 1ms, pode ser crítico para aplicações intolerantes a atraso [Alizadeh et al. 2010]. Mais ainda, observando-se o sistema do ponto de vista do plano de dados, à medida que o número de regras cresce, pode haver um grande hiato entre o tempo de instalação ou modificação de regras e de sua efetiva implementação. Por exemplo, inserções ou modificações na faixa de 1000 regras em *switches* OpenFlow comerciais podem demorar entre 1 e 10 segundos para se tornarem válidas [Rotsos et al. 2012]. Cria-se assim um problema na transferência de fluxos curtos, pois, eventualmente, mais tempo será gasto no estabelecimento das regras do que com a transferência de dados.

A motivação do presente trabalho é em contribuir para a redução dos requisitos de memória, processamento e comunicação em roteadores que tenham a capacidade de dar tratamento diferenciado a fluxos longos e curtos. Dessa forma, técnicas como o DevoFlow podem se beneficiar de decisões locais aliviando a carga de comunicação com os controladores. Outro requisito seria desenvolver uma solução que suporte a evolução de taxas dos enlaces sem produzir carga extra para o processamento e memória nos roteadores e *switches*, além de não implicar em modificações na pilha de protocolos TCP/IP. Propomos assim, um método de diferenciação de fluxos denominado RAFLE (*Random Assorter of Flow LEngths*) que infere a classe de fluxo (longos ou curtos) que pertence um pacote através de uma pequena memória recente, com a identificação de fluxo dos últimos pacotes encaminhados. Ele é capaz de regular a taxa de transmissão de cada categoria de fluxos atuando-se sobre o atraso destes no sistema, uma que a taxa de transmissão alcançada por um fluxo TCP é inversamente proporcional ao RTT [Fredj et al. 2001], não prejudicando o funcionamento de aplicações críticas (i.e. sensíveis a perdas) com a utilização de técnicas que atuam através de descarte de pacotes.

Uma vantagem do RAFLE é que os benefícios desta técnica são obtidos salto a salto, possibilitando a implementação gradual em ambientes em produção. Ainda, o RAFLE apresenta a flexibilidade de ser aplicado tanto no cenário de tráfego da Internet, como no interior de *datacenters*, uma vez que o trafego neste último ambiente também é composto em sua maioria por pequenos fluxos, que transportam poucos *KBytes* [Benson et al. 2010], sendo em sua grande parte, compostos de *hellos* e requisições de metadados em sistemas de arquivos distribuídos [Greenberg et al. 2009].

3. Metodologia

A plataforma *Click Router* [Kohle 2000] sobre *hardware* de PC foi usada para a implementação do roteador RAFLE e os demais métodos utilizados para comparação de desempenho. Mais ainda tráfego real foi utilizado, evitando que os resultados fossem limitados a modelos de tráfego e nem a simuladores. Todavia, é necessário que neste ambiente de testes, apresentado na Seção 3.1, haja algum grau de controle sobre a dinâmica de tráfego, nas modalidades descritas na Seção 3.2, permitindo comparações de desempenho entre diferentes formas de tratamento de tráfego.

Avaliamos o efeito do RAFLE sob a transferência de arquivos entre cliente e servidor FTP, comparando os resultados com: 1) roteador convencional com fila única e operando no regime *Drop Tail*; 2) classificador com segregação ideal de fluxos longos e curtos; 3) RuN2C. Uma implementação OpenFlow operando em regime (i.e., ignorando-se o período de instalação de regras) apresentaria desempenho equivalente ao classificador ideal que implementamos. Todavia, para evitarmos os efeitos de diferentes detalhes de implementação na comparação de resultados de desempenho, decidimos pelo desenvolvi-

mento de um sistema simples que distingue com 100% de acerto pacotes pertencentes a fluxos longos e curtos através de marcação no cabeçalho IP. Tal sistema também emularia um classificador com manutenção de estados *fullstate*, e por isso o nome ideal. Utilizamos também, nos casos pertinentes, uma estimativa analítica para o desempenho da transferência de arquivos baseada no conceito de distribuição equânime de banda entre os fluxos ativos [Mussi and Ribeiro 2009].

Os métodos de classificação utilizados encaminham os pacotes pertencentes a fluxos longos e curtos a filas distintas, que serão servidas de acordo com as disciplinas *Round Robin* (RR) e *Priority Schedulling* (PrioSched). Para podermos isolar os efeitos de cada esquema de diferenciação de fluxos devemos trabalhar, e isso justifica também a topologia de rede, com conexões com apenas um salto. Testes realizados em redes com topologias diferentes desta, incluindo inserção e retirada de tráfego em diferentes nós, acarretariam inúmeras combinações. A análise do impacto causado pela aplicação de uma política de tratamento de tráfego estaria sujeita a incontáveis correlações, não deixando claros os ganhos alcançados.

3.1. Ambiente de Testes

O ambiente de testes gera competição entre fluxos que transportam diferentes volumes de dados em duas situações distintas: transmissão simultânea de fluxos e início de novos fluxos quando em regime constante de carga. Optou-se pela utilização do modelo cliente-servidor FTP. Devemos lembrar que tal protocolo apresenta uma característica interessante que é a existência de sessões distintas para o controle e para a transferência de dados. Em geral, as conexões de controle transferem pequenos volumes de dados, mas a transmissão de arquivos é sempre dependente da agilidade no trâmite da sessão de controle. Assim, o FTP propicia um estudo de caso significativo para o processo de diferenciação no tratamento de fluxos longos e curtos. O *setup* básico de testes constituído por um roteador conectando cliente e servidor FTP é mostrado na parte interna da Figura 1. Nessa figura também vemos o histograma da base de arquivos a ser transferida. Ela é representativa de listagens obtidas de histogramas de arquivos em discos rígidos de computadores pessoais. Para os propósitos do experimento, há uma limitação de *threads* simultâneas e a modificação do TCP necessária para o RuN2C permitiria a segregação de fluxos com até 4MB de transferência [Avrachenkov et al. 2004]. Daí o número de 564 arquivos presentes na base e a restrição do tamanho máximo dos arquivos no histograma de 4MB. É evidente que a grande maioria dos arquivos estão concentrados na primeira faixa do histograma (o qual é representado na Figura 1 com granularidade de 100KB) restando poucos arquivos para as faixas superiores. Os fluxos curtos serão considerados pela transferência de arquivos de até 16KB, por compatibilidade com o valor sugerido para o RuN2C [Avrachenkov et al. 2004]. Todavia, se somarmos todos os arquivos considerados curtos, o total de *bytes* acumulados é praticamente equivalente a apenas um arquivo da última faixa do histograma; fato característico de distribuições caudas-pesadas.

Embora as nossas implementações de roteadores sobre PC tenham apresentado vazões superiores a 70Mbps utilizando-se placas de rede *FastEthernet*, os roteadores aqui estudados limitam banda em apenas 1Mbps por descarte de pacotes. Tal medida tem por objetivos: 1) evitar que a janela do receptor seja limitante durante as transferências das sessões TCP; 2) evitar qualquer hipótese de pequenas diferenças entre as implementações dos roteadores estudados no processamento de pacotes afetassem o desempenho; 3) inse-

rir aleatoriedade nos experimentos entre as diversas realizações emulando melhor redes reais com mais que um salto, passíveis de congestionamentos e perdas de pacotes. Entretanto, para evitarmos que descartes locais afetassem a análise de medida dos classificadores/escalonadores as filas de saída do roteador foram configuradas grandes o suficiente para não haver perda de pacotes.

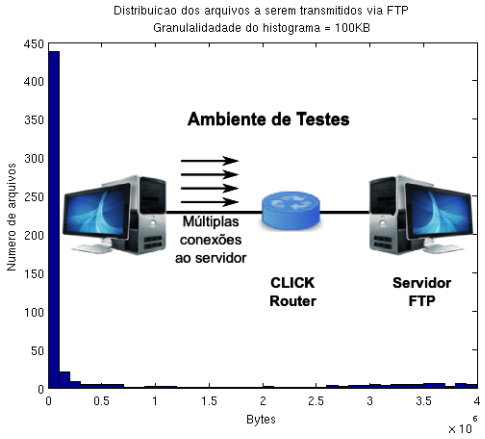


Figura 1. Ambiente de testes e base de arquivos transmitidos ao servidor FTP

3.2. Dinâmica de Tráfego e Métricas Adotadas

Os roteadores serão avaliados sob duas circunstâncias distintas. Primeiramente, avaliaremos o comportamento para o caso extremo de concorrência de recursos no qual se inicia a transmissão de toda a base de arquivos simultaneamente. Este experimento visa reproduzir situações reais como a execução de *backups*, de buscas distribuídas em *datacenters*, distribuição de tarefas em *clusters* de servidores, entre outros. Posteriormente, um segundo regime mais relaxado no qual uma carga constante de fluxos sendo transferidos é mantida. Este experimento visa estudar um ambiente de tráfego real no qual os usuários disputam a utilização de um servidor o submetendo ao limite capacidade de atendimento de requisições simultâneas.

Em nossos experimentos utilizaremos como tarefa padrão a transmissão de um arquivo dentro de uma base definida na Seção 3.1. A influência da sessão de controle do FTP é desprezível quando as tarefas têm início simultâneo, porém o trâmite de controle necessário no FTP, e em outros protocolos que necessitam de autenticação, influi no tempo total de realização da tarefa encontrando uma rede com um dado regime de carga. Em função das diferentes dinâmicas de tráfego, métricas distintas serão adotadas para cada um dos regimes citados. Todos os resultados apresentados serão médias de três realizações dos experimentos.

A análise do número de fluxos ativos ao longo do tempo será estudada. A dinâmica de finalização das sessões iniciadas será comparada com o valor esperado pelo padrão de equidade, onde os recursos são compartilhados igualmente entre os fluxos ativos. As políticas mais justas na alocação de recursos são as que mais se aproximam do padrão de equidade. Outra métrica adotada será a tempo médio de finalização dos fluxos iniciados para transmissão dos arquivos da base. A média e desvio padrão serão computados para os arquivos dentro de cada faixa de um histograma com 100KB de granularidade.

Para uma análise de desempenho em regime de carga constante, os roteadores ficam submetidos a uma carga de fluxos praticamente invariante. A cada tarefa finalizada, uma nova aleatoriamente selecionada na base de arquivos é iniciada, entrando na disputa pelos recursos de rede em desigualdade com sessões de longa duração, que alcançaram grandes janelas de transmissão. Neste caso, o tempo médio de finalização das sessões não é uma boa métrica, já que elas não são submetidas sempre às mesmas condições de ocupação dos enlaces. Analisaremos, então, o número de requisições atendidas pelo servidor ao longo do tempo, pois se as conexões de curta duração, como a transmissão de um arquivo pequeno, forem priorizadas, menor será o tempo de duração desta, abrindo a possibilidade de que uma nova requisição seja atendida. Porém, para que a análise seja justa, devemos constatar que o regime de requisições e serviço está em estado estacionário.

4. RAFLE: *Random Assorter of Flow LEngths*

Conexões TCP realizam a transmissão de informação através de rajadas de pacotes. O número de pacotes de uma rajada é determinado pela janela de transmissão da conexão. Devido a esta característica de transmissão é provável que se receba vários pacotes em sequência pertencentes a um mesmo fluxo. O RAFLE baseia-se a inferência de pacotes pertencentes a fluxos longos e curtos em uma pequena memória da identificação de fluxo dos últimos pacotes encaminhados por uma interface do roteador. Seus algoritmos de classificação são inspirados na técnica CHOKe, utilizada para oferecer proteção de fluxos TCP contra tráfego UDP que tentem ocupar completamente a capacidade de um enlace [Wang et al. 2003]. Entretanto, ele visa simplesmente o descarte de pacotes UDP que disputam espaço no *buffer* com os pacotes TCP. Na nossa proposta implementamos um sistema baseado no armazenamento do ID do fluxo dos últimos pacotes encaminhados em uma memória é possível tentar inferir se um fluxo é longo ou curto. Como ilustrado na Figura 2, a cada pacote pertencente a uma rajada de um fluxo que é processado pelo roteador, maior a probabilidade do *flowID* de um novo pacote que ingressa neste estar armazenado na memória. Desta forma, a estratégia utilizada pelo RAFLE é sortear uma posição da memória (P_a), segundo uma distribuição uniforme, em seguida, efetua-se a comparação do *flowID* existente nesta posição da memória ($FlowID\_memoria(P_a)$) com o do novo pacote ($FlowID\_atual$) que chega ao roteador. Caso os *flowIDs* sejam idênticos, o novo pacote é encaminhado para a fila de fluxos longos. Caso contrário ele ingressa na fila de fluxos curtos. Ressalta-se que o desempenho do classificador depende do ajuste do parâmetro N, que indica o tamanho da memória, em número de linhas, que o classificador irá utilizar.

Durante iniciação do mecanismo, quando a memória ainda não se encontra completa, caso não haja *FlowID* na posição da memória sorteada, o pacote é enviado para a fila de fluxos curtos e seu *FlowID* é inserido na tabela normalmente. A memória do classificador RAFLE é utilizada de forma rotativa, para que não seja necessário deslocar todos os *FlowIDs* nela inseridos para as posições posteriores a cada novo pacote. A última posição onde foi inserido um FlowID é marcada através de ponteiro incremental (P_u).

Os algoritmos do RAFLE possibilitam adaptações para funcionamento com diferentes definições de fluxo. Todavia, a identificação dos fluxos TCP utilizada neste trabalho é composta da tupla (IP de Origem, Porta de Origem, IP de Destino, Porta de Destino) totalizando apenas o armazenamento de 12 *bytes* por posição da memória. O RAFLE segrega pacotes de fluxos que ocupam muitos recursos da rede, isto é, possuam alta taxa

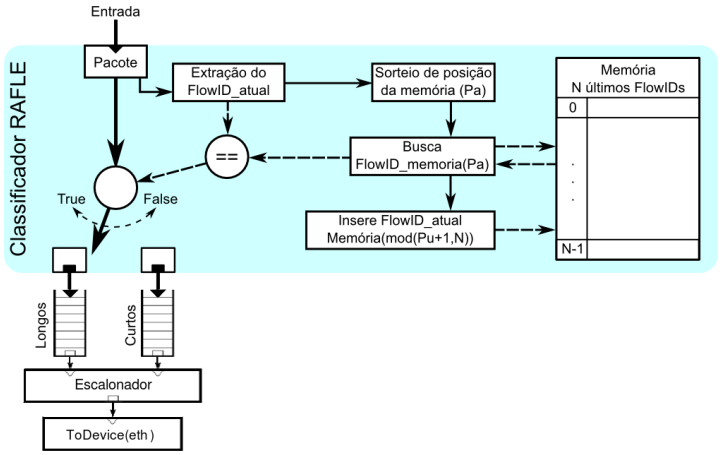


Figura 2. Diagrama de fluxo do algoritmo RAFLE

de transmissão de pacotes. No RuN2C a identificação de fluxos longos é realizada pelo volume total de dados transmitidos. Esta característica do RuN2C penaliza fluxos pouco agressivos em termos de uso de banda, mas que se alongam por tempo suficiente a ultrapassar o limiar de 16KB. Porém, para que a classificação do RAFLE seja eficiente é necessário determinar o tamanho ideal de memória recente a ser utilizada.

4.1. Determinação do Parâmetro de Memória

A determinação do tamanho da memória recente utilizada no RAFLE (parâmetro N) deve resultar na priorização de fluxos curtos. É importante perceber que quanto menor for esta memória mais chance existe de um pacote pertencente a um fluxo curto ser direcionado para a fila de fluxos longos. Por outro lado, quanto maior esta memória for, menor se torna a probabilidade do ID de fluxo de um pacote pertencente a uma rajada de um fluxo longo ser sorteado dentro das N posições desta memória. Os índices de acertos para diferentes valores de N obtidos experimentalmente são apresentados na Tabela 1. Calcularam-se as probabilidades PA_c, PE_c, PA_l, PE_l e o índice produto de acerto (i.e., $PA_c \cdot PA_l$), para diferentes valores de N. Quanto maior for o índice de acerto, mais adequada é a identificação. Obtém-se, assim, que uma memória com os IDs dos últimos 40 pacotes é a melhor escolha para a transferência da base de dados definida. No entanto, para $N = 10$ observa-se o índice de acerto próximo ao índice de $N = 40$. O principal problema em utilizar este valor baixo de N é que a aumentar a probabilidade de deslocar um dos últimos pacotes de um fluxo curto para a fila de longos, fazendo com que haja um atraso que pode ser responsável pelo não atendimento dos requisitos de latência de uma aplicação. Ressalta-se que experimentos empíricos realizados com a variação do parâmetro N, para o cenário de rede estudado, corroboram com as indicações apontadas anteriormente de que o parâmetro $N = 40$ é o mais adequado.

5. Resultados para Tarefas Concorrentes com Início Simultâneo

A finalização de fluxos ao longo do tempo efetuada pelo classificador RAFLE é apresentada na Figura 4. As curvas são apresentadas juntamente com o padrão de equidade analítico e os resultados obtidos para o classificador ideal. Inicialmente, observa-se a grande semelhança entre o comportamento, ao longo de todo experimento, das finalizações alcançadas pelo RAFLE e pelo escalonador ideal, quando são utilizadas as

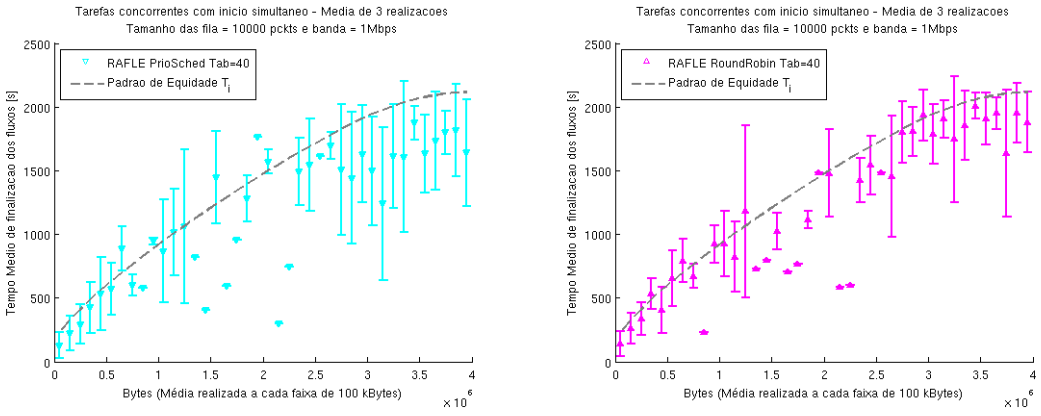
Tabela 1. Definição experimental do tamanho da memória recente (N) para o RAFLE

Limiar	Fluxos Curtos		Fluxos Longos		Índice
N	PA_c	PE_c	PA_l	PE_l	$PA_c \cdot PA_l$
10	0,937	0,063	0,140	0,860	0,131
30	0,976	0,024	0,096	0,904	0,094
40	0,979	0,021	0,139	0,861	0,136
50	0,983	0,017	0,094	0,906	0,092
60	0,986	0,014	0,079	0,921	0,078
100	0,989	0,011	0,087	0,913	0,086

$PA_c \rightarrow$ Probabilidade de acerto para pacotes de fluxos curtos
$PE_c \rightarrow$ Probabilidade de erro para pacotes de fluxos curtos
$PA_l \rightarrow$ Probabilidade de acerto para pacotes de fluxos longos
$PE_l \rightarrow$ Probabilidade de erro para pacotes de fluxos longos

mesmas disciplinas de serviço para as filas. Na faixa entre $t = 200\ s$ e $t = 500\ s$ ocorre o espelhamento da divergência entre o RAFLE PrioSched e o RAFLE RR quando comparado com os seus respectivos resultados para o classificador ideal. A partir de $t = 1900\ s$ a finalização dos fluxos do RAFLE PrioSched se distancia do ideal PrioSched e vai ao encontro do tempo de finalização global dos fluxos para os escalonadores RAFLE e ideal com disciplina RR. Quando analisamos as curvas em relação ao padrão de equidade, os roteadores implementados utilizando escalonamento RR alcançam melhor desempenho na distribuição equânime de banda. Apenas para o intervalo entre $t = 225\ s$ e $t = 410\ s$ a utilização da disciplina de serviço PrioSched alcança maior proximidade ao padrão de equidade. Para melhor aferir a proximidade do padrão de equidade passa-se a análise do tempo médio de finalização dos fluxos e seus respectivos desvios padrão.

São apresentados nas Figuras 3(a) e 3(b) os tempos médios de finalização de fluxos e desvios padrão efetuadas com o RAFLE junto ao padrão de equidade analítico. Observa-se que quando se utiliza a disciplina RR os tempos médios aproximam-se de forma mais justa ao padrão analítico. Em oposição, é notado maior dispersão para os tempos médios para as faixas de arquivos acima de 1MB no escalonador RAFLE PrioSched. Os tempos médios estão em geral abaixo da linha do padrão de equidade no RAFLE PrioSched. Todavia, a penalização dos fluxos deste intervalo ocorre no sentido do desvio padrão, indicados por barras mais longas associadas aos pontos médios para o RAFLE PrioSched em relação ao *Round Robin*.



(a) Utilizando o escalonador de prioridade estrita (b) Utilizando o escalonador *round robin*

Figura 3. Tempos médios de finalização de fluxos e desvio padrão para o RAFLE

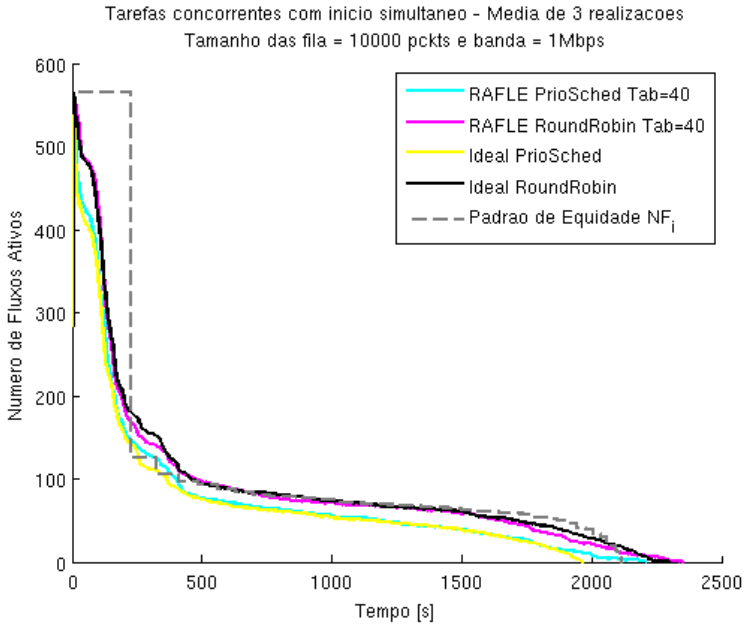


Figura 4. Diagrama comparativo de fluxos ativos por tempo para o classificador RAFLE

Tabela 2. Tempos médios e desvio padrão para o RAFLE (N=40) e RuN2C (TH=16KB)

Faixa	Padrão (T_i)		*Drop Tail*		RAFLE PrioSched		RAFLE RR		RuN2C PrioSched		RuN2C RR	
[KB]	$\bar{t}$ [s]	σ [s]	$\bar{t}$ [s]	σ [s]	$\bar{t}$ [s]	σ [s]	$\bar{t}$[s]	σ [s]	$\bar{t}$ [s]	σ [s]	$\bar{t}$[s]	σ [s]
0 à 100	225,600	0	168,267	304,951	128,028	101,626	137,564	98,583	157,072	107,247	146,344	107,868
100 à 200	326,400	0	242,231	117,352	226,004	135,493	258,138	120,287	281,258	127,090	277,436	130,317
200 à 300	410,400	0	320,642	126,085	293,833	154,693	335,265	127,435	411,895	139,599	370,537	149,912
300 à 400	488,000	0	453,563	169,347	423,693	201,833	530,535	121,696	509,720	145,537	487,665	133,801
400 à 500	562,400	0	450,703	182,594	532,777	288,520	404,530	182,580	500,218	192,643	440,305	226,393

Na Figura 5 é apresentado o diagrama de finalização de fluxos ativos ao longo do tempo, normalizado em relação a curva obtida pelo *Drop Tail*. Nos segundos iniciais o RAFLE PrioSched serve quase três vezes o número de arquivos se comparado ao roteador convencional sem priorização de fluxos. Este resultado é seguido pelo RAFLE RR e se aproxima do classificador ideal. Em $t = 400s$ os ganhos se juntam em duas faixas. O RAFLE PrioSched segue a tendência do classificador Ideal PrioSched, finalizando as conexões mais rapidamente, o que era desejado. As demais curvas seguem a tendência o classificador Ideal RR. Essa aproximação das curvas aponta que a banda está praticamente dividida entre os fluxos longos. A maior finalização de tarefas no período inicial do experimento propiciado pela diferenciação de fluxos longos e curtos faz com que as curvas decaiam mais rapidamente quando se aproxima do final do experimento. A Tabela 2 apresenta um detalhamento dos tempos médios e desvios padrão alcançados na transmissão dos arquivos das primeiras faixas do histograma da base definida. Nota-se que, assim como na análise temporal, o comportamento dos tempos médios obtidos pelo RAFLE se aproxima aos resultados obtidos pelo classificador ideal quando as mesmas disciplinas de escalonamento são utilizadas.

Apresentam-se na Tabela 3 os ganhos obtidos pelos fluxos curtos, faixa de 0 à 100KB, sobre o *Drop Tail*. Quando o classificador RuN2C é utilizado, os tempos médios corresponderam a 2/3 dos obtidos com o classificador ideal, com a disciplina RR e a menos que 1/5 quando comparados à disciplina PrioSched. Com a utilização do RAFLE, obteve-se ganho equivalente a 2/3 do valor alcançado pelo classificador ideal, para

a disciplina PrioSched. Enquanto o RR é estatisticamente equivalente ao classificador de segregação ideal. Maiores ganhos são obtidos em relação ao desvio padrão, indicando o maior grau de confiança que a transmissão dos fluxos curtos ocorre sempre em menor tempo. Todos os classificadores empregados alcançaram ganhos superiores a 180%, aproximando-se estatisticamente do classificador ideal. Isto, porém, não traz prejuízos dignos de nota à finalização dos fluxos longos, uma vez que o tempo de finalização de todas as tarefas se aproxima do valor apontado pelo roteador *Drop Tail*, como mostrado anteriormente na Figura 4.

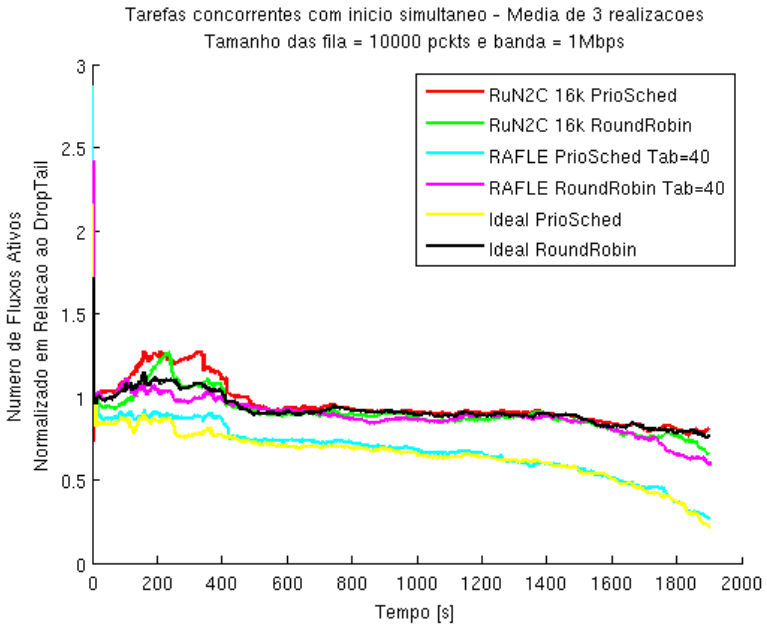


Figura 5. Diagrama de finalização de fluxos pelo tempo normalizado

Tabela 3. Ganhos percentuais para os arquivos da primeira faixa em relação ao *Drop Tail*

Escalonador	Disciplina de Serviço das Filas			
	Priority Scheduling		*Round Robin*	
	$\bar{t}$ [%]	σ [%]	$\bar{t}$ [%]	σ [%]
Ideal	44, 69	198, 44	21, 87	233, 27
RAFLE N=40	31, 43	209, 33	22, 32	200, 07
RuN2C 16k	7, 13	184, 34	14, 98	182, 70

6. Resultados para Tarefas Concorrentes com Início não Simultâneo

Neste experimento, o servidor pode atender a até 300 requisições simultâneas. A cada tarefa cumprida, a máquina cliente inicia uma nova conexão FTP de controle e, logo após, a conexão para transferência dos dados. O processo de abertura das conexões pode demorar algum tempo em uma rede congestionada, no entanto, isto deve ocorrer de forma menos demorada quando se atribui prioridade a conexões de curta duração. No classificador de segregação ideal toda a tarefa, tanto a conexão de controle, como a de dados, de um fluxo longo é tratada como tal. No RuN2C e no RAFLE isto não ocorre, as conexões de controle por transferirem pequeno volume de dados são tratadas isoladamente como fluxos

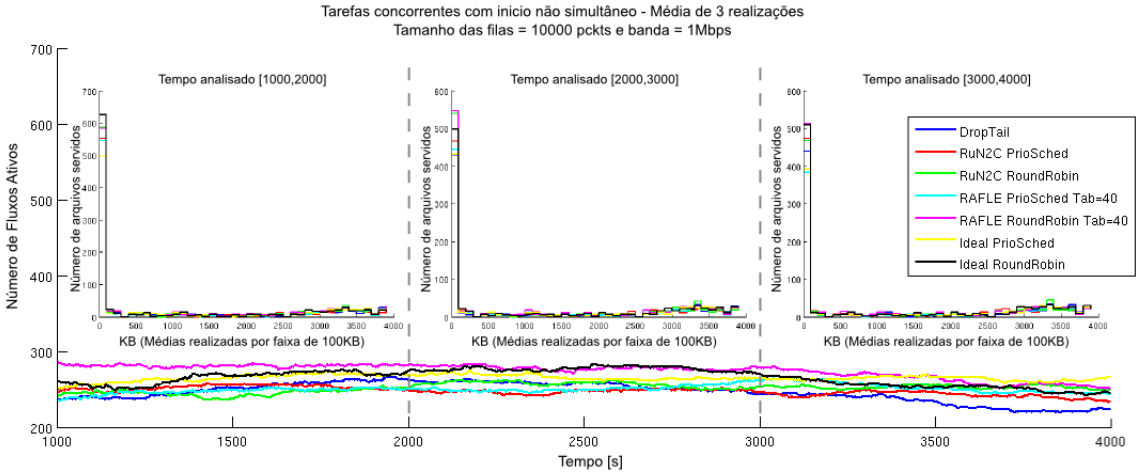


Figura 6. Diagrama comparativo de fluxos ativos com início não simultâneo ao longo do tempo

curtos, podendo agilizar, assim, o início da transferência de uma tarefa que caracterizaria um fluxo longo.

Na Figura 6 são apresentados os diagramas de fluxos ativos ao longo do tempo para os diversos classificadores implementados no intervalo de $[1000, 4000]$ segundos dos experimentos, desprezando-se, assim, a fase transitória do início e do fim do experimento. Nota-se, antecipadamente, que o classificador RAFLE com *Round Robin* consegue manter um número mais constante de tarefas ativas, demonstrando algum benefício no tratamento da conexão de controle que faz parte das tarefas. Para possibilitar uma análise visual de estacionariedade e de semelhança com a base original de arquivos, a cada 1000 segundos foi contabilizado o histograma dos arquivos servidos para cada classificador. Optou-se por traçar apenas os contornos dos histogramas obtidos, para que se pudesse comparar o número tarefas cumpridas a cada faixa. Comparando-se os resultados das três janelas de tempo, para propósitos práticos assumimos o sistema como estacionário. Uma pequena diferença ocorre entre os arquivos da primeira faixa dos histogramas (fluxos curtos) e um pequeno acúmulo ao longo do tempo de tarefas nas últimas faixas dos histogramas. Este acumulo acontece porque quando uma conexão curta é finalizada, e sequencialmente uma de longa duração assume seu lugar, esta última permanece no sistema por mais tempo. Ainda, de acordo com o classificador utilizado, a priorização de fluxos curtos irá favorecer de forma distinta o cumprimento das tarefas.

Apresenta-se na Figura 7 o histograma totalizado das médias de arquivos servidos (tarefas cumpridas) em todo o intervalo analisado, [1000s, 4000s]. Observa-se que, para as faixas seguintes à primeira, o número de arquivos servidos é praticamente o mesmo para todos os classificadores implementados. Já na primeira faixa, onde se localizam os fluxos curtos, podemos visualizar o ganho no número de arquivos servidos para cada classificador testado. Primeiramente, é importante salientar o melhor desempenho na finalização de fluxos curtos dos classificadores que utilizam a disciplina *Round Robin* para serviços das filas. Este resultado é diferente do obtido nos experimentos com tarefas concorrentes iniciadas simultaneamente. Um fator que influi para este resultado é que se um fluxo curto possuir poucos pacotes que ultrapassem o limiar determinado com o RuN2C, sendo classificados para a fila de fluxos longos. Em um sistema com carga

alta previamente instalada, tais pacotes levarão mais tempo para serem servidos se uma política de prioridade estrita for utilizada. Mais influente ainda é o efeito da política de escalonamento sobre os pacotes da sessão de controle. No classificador ideal os pacotes de tal sessão são sempre encaminhados para as respectivas filas dos fluxos que serão transportados. Assim, embora sendo uma sessão curta, o início de um fluxo longo é sempre encaminhado para a fila das conexões que transportam grandes volumes de dados, atrasando seu início.

Quando comparamos os roteadores com a disciplina RR, o pior desempenho é do RuN2C. Isso é decorrente da classificação de todos os fluxos que iniciam no sistema como curtos, misturando o início das sessões de transferência de dados de fluxos curtos e longos. Ainda, nota-se o desempenho do RAFLE ligeiramente superior ao do classificador ideal. Além da incerteza estatística, atribuímos tal fato a eventuais erros de classificação que o RAFLE pode cometer ao longo de toda a transmissão de um fluxo. Ele conseguiria aumentar a equidade de taxas de transmissão alcançadas diminuindo o tempo para cumprimento de uma tarefa entendida como um fluxo curto. Desta forma, ele penaliza com o atraso de pacotes os fluxos que já obtiveram maiores janelas transmissão, o que não ocorre com o classificador ideal.

Com a disciplina de prioridade estrita obtém-se, neste experimento, desempenho melhor que o *Drop Tail* somente quando o classificador RuN2C é empregado. O classificador denominado ideal alcança os piores resultados, já que a prioridade estrita para fluxos curtos dificulta o término das conexões longas que ocupam o sistema por mais tempo. Somado a este motivo, este classificador prejudica o início das tarefas consideradas como fluxos longos, já que suas conexões de controle também não recebem priorização aumentando o tempo necessário para cumprimento da tarefa. Assim, neste regime de funcionamento, a priorização estrita dos fluxos curtos diminui o desempenho total do sistema.

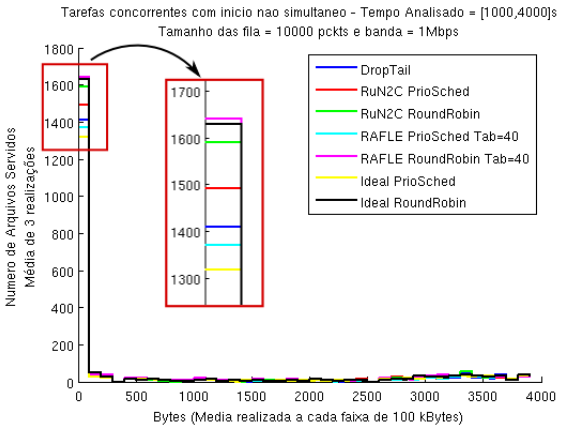


Figura 7. Histograma dos arquivos servidos no experimento de tarefas não simultâneas

Tabela 4. Ganhos em relação ao número de arquivos servidos na primeira faixa do histograma em regime estacionário

Escalonador	Ganhos percentuais em relação ao *Drop Tail* para as disciplina de serviço das filas	
	Priority Scheduling	*Round Robin*
Ideal	$-6,83\%$	$13,57\%$
RuN2C 16k	$5,57\%$	$11,45\%$
RAFLE N=40	$-2,27\%$	$14,15\%$

Na Tabela 4 são mostrados os ganhos percentuais em relação do número de arquivos servidos, tarefas cumpridas, no experimento de tarefas concorrentes com início não simultâneo. Nota-se a grande aproximação do RAFLE ao classificador ideal, como era desejado. Quando a prioridade estrita é utilizada, apenas o classificador RuN2C obteve ganhos positivos. Isto se deve a priorização estrita dificulta o termino das conexões lon-

gas, ocupando recursos do sistema por mais tempo sob a restrição do número máximo de sessões simultâneas. É importante salientar que a hipótese do classificador ideal ter um *overhead* de processamento para classificação é descartada em função de estarmos usando o roteador para tratar 1Mbps, bem abaixo de sua vazão máxima.

7. Conclusão

Um novo esquema de diferenciação de fluxos sem manutenção de estados foi proposto e implementado sobre PCs, utilizando a arquitetura CLICK, submetidos a tráfego real. Desta forma, estes não estão atrelados à qualidade de modelos de simuladores, frequentemente muito simples para traduzir o comportamento real das redes de computadores. A metodologia foi cuidadosamente pensada para que outros fatores fossem eliminados durante os experimentos.

Mostrou-se que a divisão dos recursos pode ser realizada de forma mais equânime através da segregação do tráfego em duas classes distintas, fluxos longos e curtos, e atuando no atraso relativo ao enfileiramento dos datagramas. Com estas medidas atrasos no tempo de finalização das sessões tornam-se mais proporcionais a duração e ao volume de dados trafegado individualmente por cada uma das sessões.

Dois mecanismos, o RuN2C e o RAFLE, foram investigados neste trabalho, utilizando-se variações no serviço das classes de tráfego, fluxos longos e curtos, classificadas por estes mecanismos. Dois tipos de disciplinas de serviço de filas, escalonamento de prioridade estrita (PrioSched) e *Round Robin* (RR) foram testados. Os resultados foram ainda comparados aos obtidos por um classificador de fluxos ideal, por um roteador sem a capacidade de diferenciação de tráfego (*Drop Tail*).

Notou-se que em todos os casos não são observados prejuízos relevantes aos fluxos longos, uma vez que as curvas de tempo de finalização de fluxos e de tempos médios seguem próximas ou abaixo das definidas pelo padrão analítico de equidade, evidenciando o provimento de justiça na alocação de recursos em regime de banda escassa.

Evidencia-se que o classificador RAFLE obteve ganhos significativos em relação ao RuN2C com a grande vantagem de que ele só necessita de implementação nos roteadores da rede, não necessitando de alteração nos protocolos e funcionalidades dos sistemas operacionais hoje existentes. Portanto a implantação de equipamentos operando com RAFLE pode ser gradativa, trazendo benefícios a cada salto da rede onde estiverem implementados.

Trabalhos futuros avaliarão o RAFLE em cenários de tráfego para diferentes aplicações como HTTP e protocolos específicos de *datacenters* e em topologias com vários saltos de forma a verificar o impacto de tais fatores sobre os ganhos obtidos. A interação do RAFLE com o OpenFlow será também investigada de forma a melhorar a escalabilidade, em especial no tratamento de fluxos curtos em cenários dinâmicos de tráfego.

8. Agradecimentos

A FAPES, Fundação de Amparo à Pesquisa do Espírito Santo, pelo suporte financeiro parcial no projeto 45445648 e aos revisores pelas sugestões para aperfeiçoamento do artigo.

Referências

Alizadeh, M., Greenberg, A., Maltz, D. A., Padhye, J., Patel, P., and Prabhakar, B. (2010). DCTCP : Efficient Packet Transport for the Commoditized Data Center. *SIGCOMM*.

Anelli, P., Lochin, E., and Diana, R. (2011). FavourQueue: a stateless active queue management to speed up short TCP flows (and others too!). *Cornell University Library*.

Avrachenkov, K., Brown, P., and Nyberg, E. (2004). Differentiation between short and long TCP flows: Predictability of the response time. In *In Proc IEEE INFOCOM*.

Benson, T., Akella, A., and Maltz, D. A. (2010). Network Traffic Characteristics of Data Centers in the Wild. *Traffic*, pages 267–280.

Ciullo, D., Mellia, M., and Meo, M. (2009). Two schemes to reduce latency in short lived TCP flows. *IEEE Communications Letters*, 13(10):806–808.

Curtis, A. R., Mogul, J. C., Tourrilhes, J., Yalagandula, P., Sharma, P., and Banerjee, S. (2011). DevoFlow : Scaling Flow Management for High-Performance Networks. *ACM SIGCOMM*, pages 254–265.

Dukkipati, N. (2007). *Rate Control Protocol (RCP): Congestion control to make flows complete quickly*. Phd, Stanford University.

Fredj, S. B., Bonald, T., Proutiere, A., R, G., and Moulineaux, I. (2001). Statistical Bandwidth Sharing : A Study of Congestion at Flow Level. *Simulation*, 31(4):111–122.

Greenberg, A., Hamilton, J. R., Jain, N., Kandula, S., Kim, C., Lahiri, P., Maltz, D. A., Patel, P., and Sengupta, S. (2009). VL2 : A Scalable and Flexible Data Center Network. *In Proceedings of the ACM SIGCOMM 2009*, 39(4):51–62.

Kohle, E. (2000). *The click modular router*. Phd, MIT.

Labovitz, C., Iekel-Johnson, S., McPherson, D., Oberheide, J., Jahanian, F., and Karir, M. (2009). Atlas internet observatory 2009 annual report. *47th NANOG*.

Matta, I. (2001). The war between mice and elephants. *Proceedings Ninth International Conference on Network Protocols ICNP 2001*, pages 180–188.

McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74.

Mussi, S. S. and Ribeiro, M. R. N. (2009). Análise experimental de equidade em roteador com diferenciação stateless de fluxos. *XXVII Simpósio Brasileiro de Telecomunicações - SBrT*.

Rotsos, C., Sarrar, N., Uhlig, S., Sherwood, R., and Moore, A. W. (2012). OFLOPS: An Open Framework for OpenFlow Switch Evaluation. In *Proc of PAM*.

Wang, J., Tang, A., and Low, S. H. (2003). Maximum and asymptotic UDP troughput under CHOKe. *ACM Sigmetrics*.