Avaliando a Justiça na Marcação de Tráfegos Agregados *

Igor Briglia Habib de Almeida Alves
José F. de Rezende, Luis Felipe M. de Moraes
igor@ravel.ufrj.br, rezende@gta.ufrj.br, moraes@cos.ufrj.br
Grupo de Teleinformática e Automação (GTA) - PEE/COPPE/UFRJ
Lab. de Redes de Alta Velocidade (RAVEL) - PESC/COPPE/UFRJ
http://www.gta.ufrj.br, http://www.ravel.ufrj.br

Resumo

Este artigo estuda o comportamento e avalia o desempenho de uma implementação do marcador justo (*fair-marker*) para tráfegos agregados definido em [1]. Nesta implementação, este marcador faz uso do algoritmo FRED (*Flow Random Early Drop*) de gerenciamento ativo de filas, explorando a dualidade entre o enfileiramento de pacotes e o consumo de fichas em um balde. O objetivo deste marcador é garantir a justiça entre fluxos diferentes de um mesmo assinante em um domínio de Serviços Diferenciados. Os resultados obtidos mostram que esta implementação proporciona justiça no compartilhamento da largura de faixa assegurada entre fluxos de tráfegos de diferentes características, desde que seja feito um ajuste correto dos parâmetros do algoritmo FRED. Além disso, são estabelecidos critérios bem definidos para orientar este ajuste. Finalmente, é mostrado que este marcador é ineficiente para a obtenção de justiça no compartilhamento da largura de faixa excedente.

Abstract

This article analyses the behavior and evaluate the performance of an implementation of a fair traffic marker proposed in [1]. In this implementation, this marker makes use of the FRED (*Flow Random Early Drop*) fair buffer allocation algorithm. It exploits the duality between buffer allocation and token consumption in a token bucket marker in order to enforce fairness among different flows originated from the same subscriber network in a Differentiated Services domain. The results obtained show that fairness among flows from different traffic sources can be achieved by using this marker if FRED parameters are set correctly. Well-defined guidelines are established to help configure these parameters. Finally, it is shown that the fair marker cannot provide fairness in allocation of excess bandwidth.

*Este trabalho foi realizado com recursos da FUJB, CNPq, CAPES, COFECUB, REENGE e FAPERJ.

1 Introdução

A necessidade de se oferecer diferentes níveis de serviço na Internet tem impulsionado de forma surpreendente a pesquisa em Serviços Diferenciados (DS ou *DiffServ*) [2, 3]. Esta proposta baseia-se num conjunto de mecanismos simples que tratam pacotes de forma diferenciada, em função da marcação dos bits do campo DS do cabeçalho IP. Antes da entrada num domínio de Serviços Diferenciados, este campo é marcado com um determinado valor, que no interior do domínio especificará o tipo de tratamento fornecido no encaminhamento deste pacote. A complexidade ligada à classificação e ao mapeamento de fluxos (ou microfluxos) em agregações de fluxos é deslocada do núcleo para as bordas do domínio DS. No interior do domínio, diferentes níveis de serviço são oferecidos ao tráfego agregado ao invés de para cada fluxo, o que torna esta proposta escalável.

Para fins de padronização, diferentes tratamentos (ou PHBs) estão sendo especificados, juntamente com os valores dos bits do campo DS (ou *codepoint*) associados. Dois PHBs, atualmente em ampla discussão, são o Encaminhamento Expresso (PHB EF) [4] e o Encaminhamento Assegurado (PHB AF) [5]. O PHB AF especifica quatro classes de serviço e três níveis de preferência de descarte por classe. Para fornecer um serviço fim-a-fim utilizando-se deste PHB, condicionadores de tráfego são usados na entrada do domínio para marcar os pacotes como pertencentes a uma das classes, e com um determinado nível de preferência quanto ao descarte. No interior do domínio, roteadores implementam filas com gerenciamento ativo que permitem, no caso de congestionamento, o descarte de pacotes menos prioritários (de maior preferência de descarte) dentro de uma classe.

Várias propostas foram feitas para condicionadores de tráfego, mais especificamente marcadores, para fluxos TCP [1, 6, 7, 8, 9, 10]. A função destes condicionadores é marcar o tráfego com diferentes valores de *codepoint*, conforme o perfil de serviço contratado pelo usuário. O comportamento destes marcadores tem uma grande influência no nível de serviço obtido pelos fluxos TCP que atravessam um domínio DS em termos de banda passante. Este comportamento tem sido estudado em diversos cenários: fluxos TCP com diferentes RTTs, com diferentes expectativas de serviço (taxas contratadas), em presença de fluxos insensíveis ao congestionamento (não-adaptativos). Neste último caso, foram observados problemas de justiça no compartilhamento da banda passante excedente (não-contratada) entre fluxos TCP e os não-adaptativos, tais como fluxos UDP. Porém, um problema ainda pouco explorado é o da justiça na divisão de banda passante entre fluxos de uma agregação, quando a marcação é feita sobre o tráfego agregado.

Neste trabalho, estabelecemos uma classificação dos marcadores de tráfego existentes segundo dois critérios distintos, discutimos a necessidade de marcação de tráfegos agregados, evidenciando o problema de justiça que este tipo de marcação pode trazer. Em seguida, uma proposta para solucionar este problema, chamada *fair-marker* (FM) [1], é apresentada juntamente com uma possível implementação fazendo uso do algoritmo justo de gerenciamento ativo de filas FRED [11]. Com o objetivo de entender o comportamento deste marcador em função do ajuste de seus parâmetros e avaliar seu desempenho em comparação a um balde de fichas clássico, ele foi implementado no simulador ns-2 [12]. De acordo com os resultados obtidos nas simulações, o marcador avaliado garante um alto grau de justiça no compartilhamento da largura de faixa assegurada entre os fluxos de diferentes tráfegos que compõem um agregado. Porém, a obtenção de justiça depende de um ajuste correto de seus parâmetros. Por último, mostramos a ineficiência do marcador FM no compartilhamento da largura de faixa excedente.

Este artigo está organizado da seguinte maneira. A seção 2 especifica os conceitos fundamentais envolvidos com o serviço AF e o gerenciamento ativo de filas associado. Na seção 3, várias propostas de marcadores de tráfego são discutidas e classificadas. Na seção 4, é descrito o *fair-marker* em termos de objetivos, funcionamento e implementação utilizada. Na seção 5 são apresentados os resultados de simulação obtidos, assim como uma análise destes resultados. Finalmente, na seção 6 são apresentadas as conclusões e as perspectivas deste trabalho.

2 Conceitos Fundamentais

No serviço de Encaminhamento Assegurado existe uma garantia de que os pacotes marcados com um menor valor de preferência de descarte serão encaminhados com uma alta probabilidade, contanto que o tráfego agregado não exceda a taxa contratada, definida pelo perfil de serviço. No entanto, o tráfego excedente, embora permitido, não será entregue com a mesma probabilidade. O tratamento em cada nó é definido através do PHB AF especificado em [5]. Este PHB define a maneira pela qual um provedor de um domínio DS oferece diferentes níveis de garantias de encaminhamento de pacotes IP de um de seus usuários. O PHB AF provê a entrega destes pacotes em quatro classes independentes, chamadas classes AFx (x = 1, 2, 3 ou 4), onde cada classe tem reservada em cada nó do domínio uma certa quantidade de recursos (*buffer* e largura de faixa). Os pacotes IP que desejam utilizar os serviços providos pelo PHB AF devem pertencer à uma destas classes de acordo com o contrato estabelecido.

Em cada classe AF, um pacote IP pode ser marcado (pelo usuário ou pelo domínio DS) com um dentre três diferentes níveis de preferência de descarte (*codepoint* = $AFx1$, $AFx2$ ou $AFx3$). No caso de congestionamento na rede, um nó DS deve priorizar o descarte de pacotes com um maior valor de preferência de descarte. Quando apenas dois níveis de preferência de descarte são utilizados, os pacotes de um fluxo que obedecem o perfil de serviço contratado são marcados como *IN* (*in-profile*) e os pacotes que estão além do perfil de serviço são marcados como *OUT* (*out-of-profile*).

Uma implementação do PHB AF deve tentar minimizar congestionamentos duradouros, mas permitindo-os curtos, resultantes de rajadas. Isto requer um algoritmo de gerenciamento ativo de filas. Para que um mecanismo deste tipo atenda aos requisitos do PHB AF, ele deve conter uma função que monitore o nível de congestionamento instantâneo e compute um nível de congestionamento suavizado. O algoritmo de descarte deve utilizar apenas o nível de congestionamento suavizado para determinar quando pacotes deverão ser descartados. Ainda, este algoritmo deve ser insensível às características do tráfego dos fluxos num curto espaço de tempo. Isto pode ser conseguido imprimindo um caráter aleatório à função de descarte. Um exemplo de tal algoritmo é o RED (*Random Early Detection*) [13]. O gerenciamento das filas nos roteadores internos ao domínio DS é normalmente realizado pela adoção de algoritmos RED: um para cada nível de preferência de descarte. Cada algoritmo RED é configurado de maneira mais agressiva que aquele que trata dos pacotes do nível de preferência exatamente acima, visando o descarte prioritário de pacotes com um maior valor de preferência de descarte. O objetivo é reduzir os efeitos do congestionamento antes da necessidade de descarte de pacotes de menor preferência.

Em [14], os autores apresentam quatro categorias gerais para políticas RED quando múltiplos níveis de preferência de descarte, ou múltiplas cores, são utilizados na marcação de pacotes. Estas categorias surgem da maneira pela qual podemos calcular o tamanho médio da fila e estabelecer os limiares de descarte do algoritmo RED. Neste trabalho, utilizamos a categoria Múltiplas Médias/Múltiplos Limiares. Neste tipo de política, um tamanho médio da fila é calculado para cada nível de preferência, onde o número de pacotes na fila de um determinado nível é igual à soma dos pacotes deste nível e, se existirem, dos níveis inferiores. Além disso, para cada nível de preferência de descarte temos diferentes limiares de descarte associados. Por exemplo, a fila RIO (RED com *IN* e *OUT*) [6] pertence a esta categoria. O tamanho médio da fila para pacotes *IN* é calculado utilizando apenas o número de pacotes *IN*, enquanto que o tamanho médio da fila para pacotes *OUT* é calculado utilizando o número de pacotes *IN+OUT*. Diferentes limiares de descarte são definidos para cada um destes níveis. Com esta categoria, podemos ser tão agressivos quanto desejarmos no descarte de pacotes com maiores valores de preferência de descarte.

3 Marcadores de Tráfego

Os condicionadores de tráfego são considerados elementos essenciais na arquitetura de DiffServ na Internet [3] e consistem em entidades que executam funções de medição, marcação, descarte e moldagem, a fim de alterar as características temporais do tráfego, tornando-o concordante com um determinado perfil pré-estabelecido. Nesta seção são apresentadas as várias estratégias de marcação visando classificá-las e entender as suas diferenças. A marcação de pacotes pode ser realizada tanto pelo provedor quanto pelo usuário. No segundo caso, os pacotes podem ser marcados por uma estação de trabalho, dispositivo de acesso ou roteador. Porém, o provedor pode monitorar e remarcar os pacotes para garantir a adequação do tráfego ao contrato.

As estratégias de marcação podem ser classificadas em três categorias quanto ao tipo de informação em que provedor e (ou) usuário se baseiam para marcar os pacotes. São elas: (i) *marcação por fluxo (per-flow marking)*, a qual baseia-se no estado de cada fluxo individualmente; (ii) *marcação por agregação (per-aggregation marking)*, a qual baseia-se somente no estado da agregação como um todo; (iii) *marcação por agregação atenta a fluxos (flow aware aggregation marking)*, a qual baseia-se no estado da agregação, mas com um conhecimento parcial do estado dos fluxos que compõem a agregação.

Quando a marcação por fluxo é utilizada, o dispositivo responsável pela marcação precisa monitorar cada fluxo do tráfego agregado. Isto pode acarretar problemas de propriedade escalar quando o cliente gera um número muito grande de fluxos. No caso de um servidor WWW, o número gerado de fluxos de curta duração pode impossibilitar a utilização da marcação por fluxo, devido à quantidade de estados requerida pela tarefa de medição. Sobre este aspecto, a marcação por agregação é mais simples e adequada para situações onde existe um número elevado de fluxos. No entanto, a marcação por agregação pode apresentar problemas de injustiça na medida em que desconhece o estado dos fluxos individualmente dentro de um tráfego agregado. Esta injustiça surge em virtude de diferentes *round-trip times* (RTTs), diferentes taxas contratadas, diferentes larguras de faixa ou diferentes níveis de congestionamento enfrentados pelos fluxos através da rede [15], fazendo com que os fluxos obtenham diferentes taxas de marcação. Na marcação por agregação atenta a fluxos, os marcadores não sabem quantos fluxos no total estão sendo expostos à marcação e assim não mantêm nenhum parâmetro ou estado para algum fluxo em particular. Porém, o marcador pode manter informações para um número limitado de fluxos de forma dinâmica, ou seja, estados podem ser criados e destruídos ao longo do tempo, garantindo portanto uma propriedade escalar.

A maioria dos estudos de redes com DiffServ utiliza a marcação por fluxo [6, 7, 14, 15, 16, 17, 18, 19]. As estratégias de marcação utilizadas em [15] se baseiam no tráfego agregado mas apresentam três mecanismos para reduzir a injustiça entre os fluxos devido a diferentes RTTs e taxas contratadas. Além disso, é feita uma comparação entre marcação por fluxo e por agregação. Em [19], são propostas estratégias de marcação para serem usadas nas bordas da rede a fim de mitigar o efeito dos RTTs, interações entre tráfegos TCP e UDP, assim como, taxas contratadas diferentes. Em [1], é proposta uma estratégia de marcação por agregação atenta a fluxos. Este marcador (FM) é alvo principal deste trabalho e será abordado em detalhes nas seções 4 e 5.

Outra forma de classificar a marcação de pacotes se refere à forma de verificar a conformidade do tráfego com o perfil acordado, ou seja, o mecanismo de medição. Assim, os marcadores podem ser classificados em duas categorias: *marcação baseada em balde(s) de fichas (token-bucket based marking)*, predominante nos estudos recentes em redes DiffServ; e *marcação baseada em estimadores de taxa média (average rate estimator based marking)*. É importante frisar que este tipo de classificação é totalmente independente ao apresentado anteriormente, permitindo que um determinado marcador seja classificado simultaneamente de acordo com os dois critérios.

Na marcação baseada em baldes de fichas [7, 8, 9, 14, 16, 17, 20], os mecanismos de medição incluem um ou mais baldes de fichas para comparar o tráfego gerado com um ou mais níveis de taxa assegurada, dependendo do número de níveis de marcação (preferência no descarte) desejado. A maioria das propostas apresenta dois níveis de marcação (*IN* e *OUT*). Em [8, 9], três níveis de marcação (Verde, Amarelo e Vermelho) são utilizados para melhorar a justiça no compartilhamento da largura de faixa excedente entre fluxos TCP e UDP. Em [14, 16], a eficiência de três níveis de marcação é avaliada. Em [7], são apontadas vantagens da marcação baseada em baldes de fichas em relação à baseada em estimadores de taxa. Em [20], um modelo analítico é proposto para analisar o desempenho do balde de fichas. Conforme será visto na seção 4, o FM utiliza um balde de fichas como mecanismo de medição. A marcação baseada em estimadores de taxa é feita através da medição da taxa média de envio de pacotes por fluxos individuais ou do tráfego agregado [6, 15, 19, 21]. Na proposta pioneira [6], pacotes são marcados como *OUT* com uma probabilidade crescente a partir do momento em que a taxa estimada excede um determinado limite.

4 O Marcador Justo (*Fair-Marker*)

4.1 Objetivos e Descrição do Funcionamento

O FM (*fair-marker*) consiste em um marcador baseado em balde de fichas e que pratica a marcação por agregação atenta a fluxos, visando obter justiça entre os fluxos à medida em que os pacotes são marcados. Para isto, o FM deve de alguma maneira distribuir as fichas de uma forma justa entre os fluxos. A fim de atingir este objetivo, se faz necessário manter informação a respeito do consumo de fichas pelos fluxos monitorados. Todavia, para evitar uma explosão de estados, o FM armazena apenas os estados dos fluxos que consumiram fichas (tiveram pacotes marcados como *IN*) durante o último intervalo de tempo correspondente ao preenchimento de todo o balde, denominado $TBFT$ (*Token Bucket Fill Time*).

Para isso, lança-se mão de uma analogia entre um balde de fichas e uma fila, onde manter estados para os fluxos que consumiram fichas durante o ultimo $TBFT$ seria análogo a manter estados para fluxos que possuem pacotes numa fila, desde que esta possua o tamanho do balde. Indo mais adiante, pode-se imaginar um pacote consumindo fichas como uma situação análoga ao que seria o "rastro" (*trace*) deste pacote substituindo estas fichas no balde. Na prática, cria-se uma fila complementar onde este rastro é armazenado. Sempre que uma ficha é gerada, consulta-se a fila para saber se o número de fichas no balde é suficiente para retirar o rastro do pacote na cabeça da fila (o mais "antigo"). Além disso, como se deseja um marcador justo então a fila deve ter um algoritmo de enfileiramento justo. Ou seja, se o espaço da fila for compartilhado de forma justa, então o mesmo acontecerá para o consumo de fichas no balde. Portanto, determinar se um pacote deve consumir fichas (ser marcado como *IN*) equivale a determinar se o seu rastro deve ser posto na fila complementar.

Não há uma especificação exata de onde o FM deve estar localizado. Um requerimento contudo, é que este deve estar no ponto onde todos os fluxos dos assinantes convergem, antes de entrarem no domínio DS. A figura 1 mostra o diagrama de blocos do FM com todos os seus componentes, descritos a seguir:

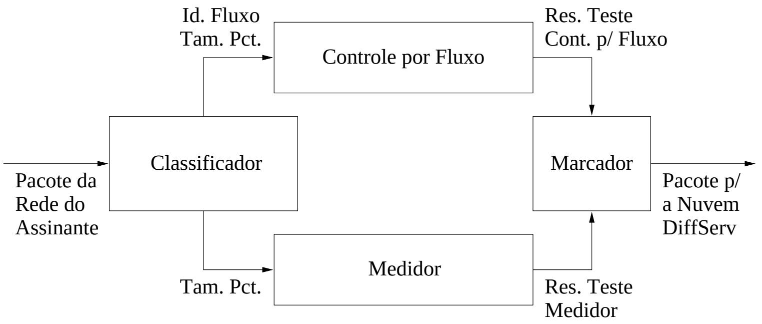


Figura 1: *Fair-marker.*

- Classificador: recebe um pacote da rede do assinante e extrai o identificador do fluxo ao qual pertence este pacote. Um fluxo pode ser especificado pelo conjunto de 5 elementos conforme abaixo (microfluxo). Porém, a granulosidade não é fixa na medida em que alguns destes campos podem não ser especificados.

  - (Protocolo, Endereço Fonte, Porta Fonte, Endereço Destino, Porta Destino)

  O identificador de fluxo é transmitido do classificador para o Controle de Alocação de Fichas por Fluxo após a sua obtenção.

- Controle de Alocação de Fichas por Fluxo: utiliza um algoritmo justo de distribuição de fichas para decidir se um fluxo irá ou não consumir uma quantidade desproporcional, se comparada aos demais fluxos que estão sendo monitorados no momento. O algoritmo faz esta análise com base não só na sua própria lógica mas também na informação armazenada, referente à contagem de pacotes (ou bytes) por fluxo. Em caso de desproporcionalidade no consumo de fichas o resultado da análise será negativo, caso contrário será positivo. O critério de proporcionalidade para o consumo de fichas dependerá do algoritmo escolhido para a implementação do FM. Em [1] são sugeridos dois algoritmos: o FRED [11] e o DTQ [22].

- Medidor: o medidor consiste num balde de fichas clássico, possuindo dois parâmetros. O primeiro é o tamanho do balde BS (*Burst Size*) e corresponde ao tamanho máximo de uma rajada de tráfego que pode ser produzido em sua saída. O segundo é a taxa de preenchimento de fichas PR (*Profile Rate*). BS é medido em bytes e PR em bytes de pacotes IP por segundo. O medidor do FM funciona conforme descrito a seguir. Seja T o número de fichas no balde (em bytes) e B o tamanho do pacote em bytes. O balde se encontra cheio no inicio ($T(0) = BS$) e é incrementado de um byte PR vezes por segundo, jamais ultrapassando BS. Na chegada de cada pacote num determinado instante de tempo t, o seguinte teste é realizado: se há fichas suficientes para o pacote ($T(t) \leq B$) o resultado será positivo, caso contrário ($T(t) < B$) será negativo.

- Marcador: reflete o resultado dos dois testes praticados pelos componentes anteriores, alterando o valor do campo DS de cada pacote IP. Assumindo o encaminhamento assegurado, caso os dois testes tenham produzido resultado positivo, o pacote será considerado dentro do perfil (*IN*) e marcado com o codepoint AFx1. Caso contrário, isto é, pelo menos um dos testes tenha produzido resultado negativo, o pacote será considerado fora do perfil (OUT) e marcado com o *codepoint* AFx2 ou AFx3. Nesta versão do FM portanto, apenas dois níveis de preferência de descarte são utilizados.

4.2 Implementação Utilizada

De acordo com tudo que foi dito podemos chegar a um algoritmo simples para a implementação do FM utilizando qualquer disciplina de enfileiramento justa, conforme descrito a seguir. Para cada pacote que chega para a marcação, o balde de fichas será atualizado primeiro. Serão acrescentadas $(t\_now\_ - t\_last\_) * PR$ fichas até BS, onde $t\_now\_$ é igual ao tempo atual, $t\_last\_$ o tempo em que um último pacote foi recebido, PR a taxa de preenchimento de fichas e BS o tamanho do balde. O próximo passo será então ver quantos rastros de pacotes podem ser retirados da fila complementar, bastando calcular $(T\_last\_ - T\_now\_)$ fichas, onde $T\_last\_$ é o número de fichas em $t\_last\_$ e $T\_now\_$ o número de fichas em $t\_now\_$ (atualizado acima). Em seguida, deve-se usar este número de fichas para retirar tantos pacotes quantos couberem nesta quantidade. Isto equivale a limpar a tabela de estados só deixando nela fluxos que consumiram fichas durante o último $TBFT$ $(= BS/PR)$. Conforme os rastros dos pacotes vão sendo retirados, as estatísticas de consumo de fichas no balde (utilização de espaço na fila) vão sendo atualizadas, tarefa realizada pelo próprio algoritmo justo de gerenciamento de filas.

O próximo passo então será analisar o número de fichas no balde. Se este for insuficiente, o pacote será marcado como OUT e enviado. Seu rastro não será colocado na fila complementar e nenhuma estatística por fluxo será alterada. Se o número de fichas for suficiente, então o consumo de fichas no último $TBFT$ (utilização de espaço na fila de rastros) determinará então se o pacote poderá consumir as fichas ou não. Para isso, será utilizado o algoritmo justo de gerenciamento de filas para determinar se o pacote será enfileirado ou não. Caso ele não possa ser enfileirado, o seu rastro não será colocado na fila, o pacote será marcado como OUT e enviado. Caso ele possa ser enfileirado, o seu rastro entrará na fila, as fichas serão consumidas e o pacote será marcado como *IN* e enviado. Vale notar que apesar da aplicação de um algoritmo de gerenciamento de filas, não ocorre descarte de nenhum pacote. O que seria um descarte na fila resulta na marcação do pacote como *OUT*. Em ambos os casos (*IN* ou *OUT*) o pacote sempre segue adiante pela rede. O pseudo código abaixo ilustra toda a lógica exposta anteriormente.

```
início:
   T_now_ = BS;
   T_last_ = T_now_;
   t_last_ = tempo_corrente();

a cada pacote que chega:
   atualizar_contagem_de_fichas;
   atualizar_fila_de_rastros;
   /* se existem fichas o suficiente */
   se (T_now_ >= tamanho_do_pacote)
      /* se o rastro do pacote pode ser enfileirado
         na fila complementar */
      /* (se o fluxo não irá consumir uma
         quantidade desproporcional de fichas) */
      se (pacote pode ser enfileirado)
         marcar_pacote_como_IN;
         /* consumir fichas */
         T_now_ = T_now_ - B;
         senão
            marcar_pacote_como_OUT;
      senão
         marcar_pacote_como_OUT;
      atualizar_estado_global_do_FM;
      enviar_pacote;

atualizar_contagem_de_fichas:
   T_now_ = T_now_ + (tempo_corrente()-t_last_)*PR;
   se (T_now_ > BS)
   T_now_ = BS;

atualizar_fila_de_rastros:
   /* atualiza número de fichas disponível
   para apagar rastros de pacotes */
      resto = (T_now_-T_last_);
      enquanto (resto > tamanho do pacote
                        na cabeça da fila)
         retirar_pacote_da_cabeca_da_fila;
         resto = resto - B;

atualizar_estado_global_do_FM:
   t_last_ = tempo_corrente();
   T_last_ = T_now_;
```

De acordo com o pseudo código acima, o desempenho do FM quanto à justiça na distribuição dos pacotes será função do algoritmo escolhido para o compartilhamento justo de *buffer*. Nesta implementação, o algoritmo utilizado é o FRED (*Flow Random Early Drop*) [11], o qual consiste numa versão modificada do RED [13]. O FRED introduz dois limiares novos correspondentes aos números mínimo (min_q) e máximo (max_q) de pacotes que cada fluxo pode ter na fila, além dos dois limiares mínimo (min_{th}) e máximo (max_{th}) presentes no RED. O FRED também controla o número ($qlen_i$) e a média (avg_q) de pacotes por fluxo, favorecendo os fluxos que possuem menos pacotes do que a média. Além disso, o FRED penaliza fluxos que tentam exceder o número máximo permitido de pacotes por fluxo. As informações por fluxo só são mantidas para os fluxos que possuem pacotes na fila.

O objetivo destas modificações é melhorar a justiça quando uma mistura de tráfegos de características diferentes compartilham um enlace de transmissão. Segundo [11], o RED se comporta de maneira injusta na medida que impõe a mesma taxa de descarte para os fluxos, independente de suas frações de utilização da largura de faixa compartilhada. No FRED, ocorre proteção aos fluxos pertencentes a tráfegos ditos adaptativos na medida em que fluxos de tráfegos não adaptativos são isolados de maneira mais agressiva.

Dentro do contexto do FM e de acordo com a analogia entre o enfileiramento de pacotes e a distribuição de fichas, em cada $TBFT$: (i) todo fluxo poderá marcar até min_q pacotes como *IN* desde que o número médio de pacotes marcados como *IN* não exceda max_{th}; (ii) assim como no RED, pacotes adicionais estarão sujeitos a serem marcados como *OUT* de forma aleatória (*random drop*), se o número médio de pacotes marcados como *IN* exceder min_{th}, porém isto só acontecerá para fluxos que já tenham marcado mais do que avg_q pacotes *IN*; (iii) um fluxo nunca poderá marcar mais do que max_q pacotes como *IN*, e cada tentativa de ultrapassar este limite será contabilizada, fazendo com que fluxos com alto valores de contagem não consigam marcar mais do que avg_q pacotes como *IN*. Desse modo, a implementação utilizando FRED aqui proposta obterá justiça entre fluxos na medida em que (i) protege fluxos que se adaptam mais lentamente a uma maior disponibilidade de largura de faixa (altos RTTs); (ii) ameniza a heterogeneidade entre os fluxos descartando mais de fluxos que estão tendo maior taxa de marcação de pacotes *IN*; e (iii) permitirá que fluxos adaptativos enviem rajadas de tráfego que será assegurado, mas evitará que tráfego não adaptativo monopolize o consumo de fichas do marcador. Além disso, a aplicação desta lógica apenas para fluxos que tenham marcado pacotes como *IN* no último $TBFT$ garante a propriedade escalar da solução.

5 Simulações

5.1 Estudo do Comportamento

Neste primeiro conjunto de simulações, foi utilizada a topologia da figura 2. A partir dela, várias conclusões a respeito da influência dos parâmetros no desempenho do FM puderam ser obtidas.

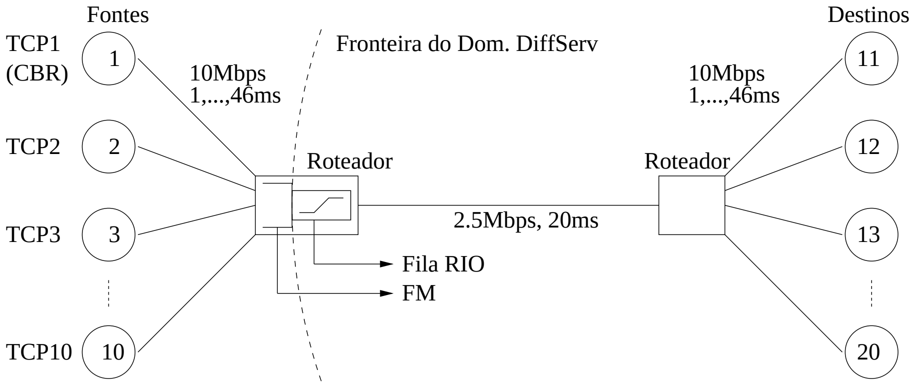


Figura 2: Topologia 1.

Foram simulados 4 cenários denominados TCP homogêneos (mesmo RTT) com e sem CBR (1 e 3), e TCP heterogêneos (RTTs distintos) com e sem CBR (2 e 4). Em todos eles existe uma fonte de tráfego FTP/TCP[1] da fonte n para o destino $n + 10$, onde $n = 1, 2, ..., 10$. Utilizamos TCP Reno (por ser o mais difundido na Internet) com janela máxima de congestionamento de 90 pacotes. Nos cenários TCP homogêneos, o retardo de propagação nos enlaces entre fontes/destinos e roteadores é de 1ms. Nos cenários TCP heterogêneos, este valor varia de 1ms até 46ms numa progressão aritmética de razão 5ms, com RTTs mínimos variando de 44ms até 224ms. Cenários com CBR possuem uma única fonte de tráfego CBR/UDP do nó 1 para o nó 11 com taxa de transmissão de 2.5Mbps (100% do gargalo). Todos os pacotes de dados são de 1000 bytes. A fila RIO é de 50 pacotes, com parâmetros para as filas *IN* e $IN + OUT$ valendo [0.3,0.6,0.002,0.1] e [0.14,0.3,0.002,0.1][2] respectivamente [6].

O tempo total de todas as simulações foi de 55s. As fontes começam a transmitir entre 0s e 0.5s de forma aleatória para evitar o sincronismo e param no instante 50s. Para desconsiderar efeitos do transiente das conexões TCP, todos os resultados são computados no intervalo de 10s a 50s.

No FM, o tamanho do balde BS é de 50 pacotes e a taxa de preenchimento PR vale 1Mbps (40% do gargalo). Para entender a influência dos parâmetros do FRED no desem-

[1] FTP pode ser utilizado no ns-2 para fazer com que o TCP sempre tenha dados a transmitir.

[2] $[min_{th}/qlim, max_{th}/qlim, w_q, max_p]$, onde $qlim$ corresponde ao tamanho da fila RIO em pacotes.

penho do FM, utilizamos um procedimento similar ao método de classificação (*ranking method*) descrito em [23]. Os parâmetros min_q, $max_q = min_{th}$ e max_{th} assumem os valores 2, 4, 8, 16, 32 e 50, respeitando as desigualdades $min_q < (max_q = min_{th}) < max_{th}$. Os demais parâmetros foram mantidos constantes com valores $w_q = 0.002$ e $max_p = 0.02$ de acordo com [11]. Logo, foram obtidas 20 configurações, numeradas conforme a primeira coluna da tabela 1 e cujos valores dos parâmetros estão especificados nas 3 colunas seguintes ($max_q = min_{th}$).

Tabela 1: Valores dos parâmetros do FM (FRED) nos 4 cenários

#	min_q	min_{th}	max_{th}	1	2	3	4	$Tot.$	$Cls.$
1	2	4	8	18	8	1	1	28	20
2	2	4	16	19	13	4	3	39	13
3	2	4	32	20	18	7	7	52	5
4	2	4	50	20	17	7	7	51	6
5	2	8	16	17	11	2	2	32	17
6	2	8	32	20	18	8	8	54	3
7	2	8	50	20	18	13	12	63	1
8	2	16	32	17	17	4	2	40	11
9	2	16	50	17	12	11	9	49	9
10	2	32	50	17	4	10	6	37	15
11	4	8	16	17	10	2	1	30	19
12	4	8	32	20	17	8	8	53	4
13	4	8	50	20	17	12	12	61	2
14	4	16	32	17	17	4	2	40	11
15	4	16	50	19	13	11	8	51	6
16	4	32	50	17	4	10	8	39	13
17	8	16	32	19	16	4	2	41	10
18	8	16	50	18	12	11	9	50	8
19	8	32	50	15	4	7	6	32	17
20	16	32	50	17	4	10	5	36	16

Tabela 2: Níveis de classificação

min	max	#	min	max	#	min	max	#	min	max	#
0.5000	0.6999	1	0.9100	0.9199	6	0.9600	0.9699	11	0.9940	0.9959	16
0.7000	0.7999	2	0.9200	0.9299	7	0.9700	0.9799	12	0.9960	0.9979	17
0.8000	0.8499	3	0.9300	0.9399	8	0.9800	0.9899	13	0.9980	0.9989	18
0.8500	0.8999	4	0.9400	0.9499	9	0.9900	0.9919	14	0.9990	0.9994	19
0.9000	0.9099	5	0.9500	0.9599	10	0.9920	0.9939	15	0.9995	1.0000	20

Foram feitas 5 simulações para cada configuração em cada um dos 4 cenários, e tiradas as médias do número de pacotes marcados como *IN* para cada fluxo (TCP ou CBR). As configurações foram então comparadas pelo índice de justiça fi (*fairness index*) conforme a equação 1 [24], onde x_i é o número de pacotes *IN* para o fluxo i e $N = 10, 11$ em cenários sem e com CBR respectivamente.

$$fi = \frac{(\sum_{i=1}^{N} x_i)^2}{N * \sum_{i=1}^{N} (x_i)^2} \tag{1}$$

A seguir, em cada cenário, foi atribuído um nível para os valores de fi conforme a tabela 2. A diferença entre os níveis possui uma granulosidade maior conforme o valor de fi se aproxima de 1. Os níveis de classificação para cada configuração e cenário traduzindo o resultado das simulações em termos de justiça na marcação de pacotes estão nas colunas numeradas de 1 a 4 da tabela 1. Em seguida, aparecem o total nos 4 cenários e a classificação de cada configuração em função deste.

Com base na tabela 1, recomendações bem definidas podem ser feitas a respeito da configuração do FM, a fim de maximizar a justiça no compartilhamento da largura de faixa assegurada:

- Valores de max_{th} mais baixos (próximos de min_{th}) degradam o desempenho do FM, o que pode ser verificado comparando as configurações [1, 2, 3 e 4], [5, 6 e 7], [8 e 9], [11, 12 e 13], [14 e 15] e [17 e 18] entre si. Nestes casos, o nível médio máximo de pacotes na fila de rastros *IN* decresce, e com ele a capacidade de armazenamento de estados. Nos cenários com CBR, este fluxo sempre terá seu espaço mas o mesmo não acontecerá para os fluxos adaptativos (TCP). Logo, o número de pacotes *IN* cai para o tráfego TCP e aumenta para o CBR, diminuindo a justiça. Para exemplificar, quando mudamos o valor de max_{th} de 50 para 16 nas configurações 7 e 5 para o cenário 4, o número total de pacotes marcados como *IN* para os fluxos TCP sofre uma redução de 19%, enquanto que para o fluxo CBR ocorre um aumento de 88%. Nos cenários TCP heterogêneos, fi também cai devido a maior sensibilidade às perdas por parte das conexões mais longas (RTTs maiores). De acordo com os resultados, recomendamos max_{th} próximo ou igual a BS.

- Quando max_q ultrapassa um certo limite em relação a max_{th}, o desempenho do FM cai conforme mostram as configurações [2 e 5], [6 e 8], [7, 9 e 10], [12 e 14], [13, 15 e 16], [18 e 19]. Isto ocorre porque o número de pacotes que podem ser marcados como *IN* num $TBFT$ aumenta em demasia, diminuindo a capacidade do algoritmo FRED penalizar tanto o fluxo CBR quanto os fluxos TCP de menor RTT. Isto garante a estes fluxos a maioria do espaço na fila de rastros em detrimento dos fluxos TCP de maior RTT. Para a configuração 10 no cenário 2 o fluxo TCP 1 chega a marcar 3,5 vezes mais pacotes *IN* do que o TCP 10.

- Por outro lado, max_q não deve ser muito baixo para que os fluxos TCP não passem a ser penalizados com pacotes OUT por atingirem rapidamente este número de pacotes na fila de rastros. Isto aumentará o número de perdas na fila RIO, diminuindo a janela de congestionamento e consequentemente o número de pacotes submetidos à marcação (configurações [3 e 6] e [4 e 7]). Quanto maior o RTT, mais sensível será o fluxo a estas perdas. Vale notar que isto só não ocorre para o cenário TCP homogêneos, pois todos os fluxos possuem o mesmo RTT.

- O parâmetro min_q praticamente não influencia no desempenho do FM, pois o mais importante é que fluxos não marquem mais do que o justo, sendo difícil não conseguirem marcar no mínimo min_q pacotes durante um $TBFT$. As configurações [5 e 11], [6 e 12], [7 e 13], [8 e 14] e [9, 15 e 18] ilustram esta afirmação.

Para reforçar estas argumentações apresentamos na tabela 3 os valores de correlação cruzada entre resultados totais e parâmetros do FM, assim como a diferença entre estes.

Os valores mostram que max_{th} possui uma alta influência assim como as diferenças entre max_{th} e max_q, e max_{th} e min_q, ao contrário de min_q.

Fica claro portanto que um bom desempenho do FM com relação à justiça na marcação de pacotes *IN* depende do ajuste correto de seus parâmetros. De acordo com o estudo realizado uma configuração recomendada seria $max_{th} = BS$, $min_q \leq 10\% BS$ e $2\ min_q \leq max_q \leq 25\%\ max_{th}$.

Tabela 3: Correlação entre parâmetros e justiça

min_q	max_q, min_{th}	max_{th}	$max_q - min_q$	$max_{th} - min_q$	$max_{th} - min_{th}$
0.5311	0.5811	0.9049	0.5004	0.9061	0.9139

5.2 Análise Comparativa de Desempenho

Nesta seção, objetivamos comparar o desempenho entre o FM e um marcador balde de fichas clássico, denominado TB (*Token-Bucket*). Para isto, utilizamos a topologia da figura 3, a qual permite simular uma situação mais realista.

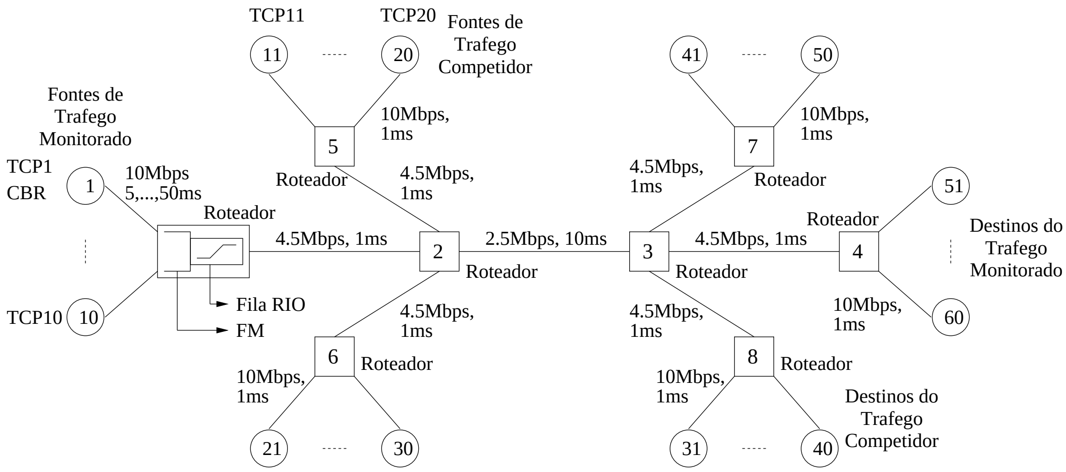


Figura 3: Topologia 2.

O tráfego a ser monitorado consiste em dez fontes de tráfego TCP Reno dos nós 1,...,10 para os nós 51,...,60, além de uma fonte de tráfego CBR/UDP do nó 1 para o nó 51 com taxa de transmissão de 2.5Mbps (100% do gargalo). A janela máxima de congestionamento do TCP vale 100 pacotes. Os retardos nos enlaces entre os nós 1,..,10 e o roteador 1 variam de 5ms até 50ms numa progressão aritmética de razão 5ms, com RTTs mínimos variando de 36ms até 126ms. Dez fontes de tráfego TCP adicionais, dos nós 11,...,20 para os nós 31,...,40, competem com o tráfego monitorado. Entre os roteadores, todas as filas são RIO idênticas à utilizada na topologia 1.

No FM, o tamanho do balde é de 50 pacotes e a taxa de preenchimento PR varia de 8% à 80% do gargalo (200kbps à 2Mbps). Os parâmetros min_q, $max_q = min_{th}$ e max_{th} assumem os valores 2, 8, 8 e 50 respectivamente e foram obtidos da configuração 7 do conjunto de simulações da seção 5.1. Assim como nas simulações anteriores, $w_q = 0.002$ e $max_p = 0.02$.

Foram feitas 5 simulações para cada valor de PR e tiradas as médias de todas as grandezas de interesse. As barras em cada ponto delimitam um intervalo de confiança de

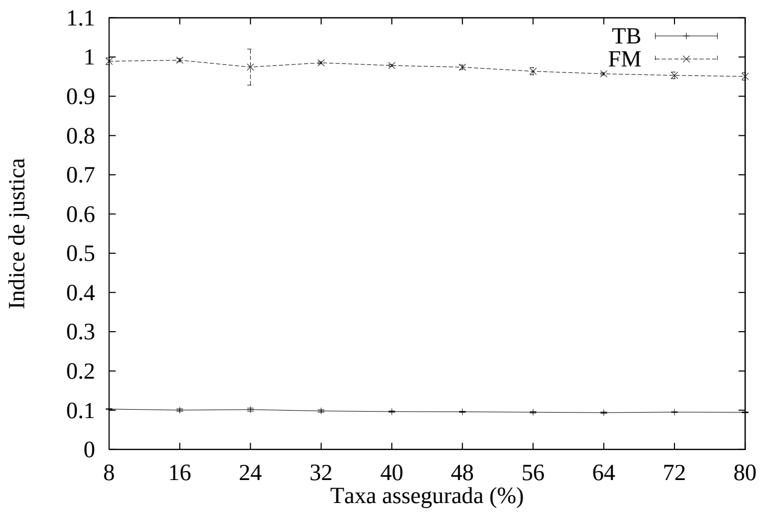


Figura 4: Índices de justiça no compartilhamento da largura de faixa assegurada

95%. O tempo total de todas as simulações foi de 60s. As fontes começam a transmitir entre 0s e 5s de forma aleatória e param no instante 59s. Para desconsiderar efeitos do transiente das conexões TCP, todos os resultados são computados no intervalo de 10s a 59s.

Na figura 4, comparamos o índice de justiça na vazão de pacotes *IN*, isto é, no compartilhamento da largura de faixa assegurada. O desempenho do FM é bastante superior ao do TB, obtendo índices de justiça sempre acima de 0.90. O TB, por sua vez, independente do valor de PR, mantém o índice pouco acima de $1/11$. Isto é, o fluxo CBR praticamente utiliza toda a largura de faixa assegurada. Este baixo desempenho é explicado pelo fato do TB realizar a *marcação por agregação* sem nenhum mecanismo para obtenção de justiça entre os fluxos. Pode-se notar uma pequena queda no índice de justiça para o FM com o aumento de PR. Isto ocorre porque um número excessivo de pacotes *IN* causa um aumento gradativo do nível médio de pacotes *IN* na fila RIO. Como consequência, a perda de pacotes OUT é cada vez maior, fazendo com que as conexões TCP reduzam a janela de congestionamento, diminuindo suas vazões proporcionalmente ao valor do RTT. No entanto, o fluxo CBR mantém a sua taxa de transmissão. Estes dois fatores em conjunto explicam a leve redução no índice de justiça com o aumento de PR.

A figura 5 mostra como o FM consegue fazer com que a vazão das conexões TCP, em média, suba quase que linearmente com o aumento da taxa assegurada, embora a justiça entre elas sofra uma degradação conforme explicado anteriormente. O TB, no entanto, mantém um baixo nível de vazão em comparação à taxa assegurada.

Resta então analisar o desempenho do FM com relação ao compartilhamento da largura de faixa excedente. As figuras 6.a e 6.b mostram os índices de justiça das vazões excedente (pacotes OUT) e total (pacotes $IN + OUT$). A figura 6.a mostra que o FM e o TB têm o mesmo desempenho com relação à largura de faixa excedente, o que pode ser explicado pelo fato de ambos os mecanismos procederem da mesma forma quanto ao tratamento de pacotes OUT, sem nenhuma ação no sentido de garantir justiça. Na figura 6.b, a justiça entre os fluxos na taxa assegurada, faz com que a curva para o FM suba à medida que o número de pacotes *IN* na rede aumenta.

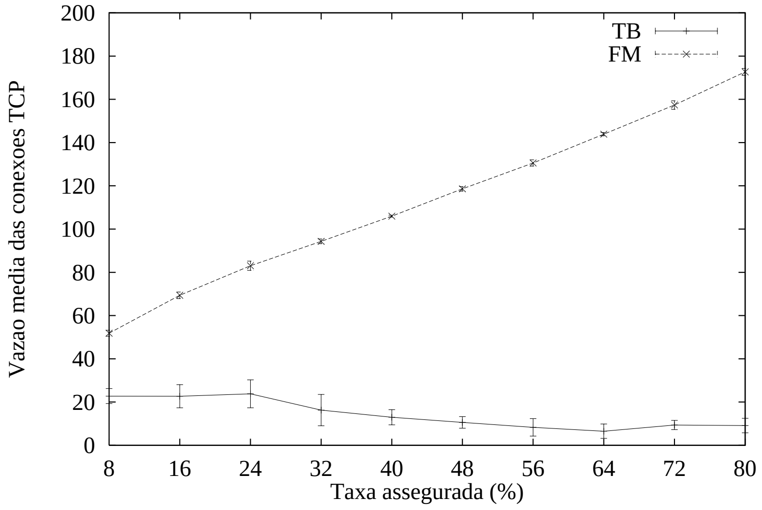
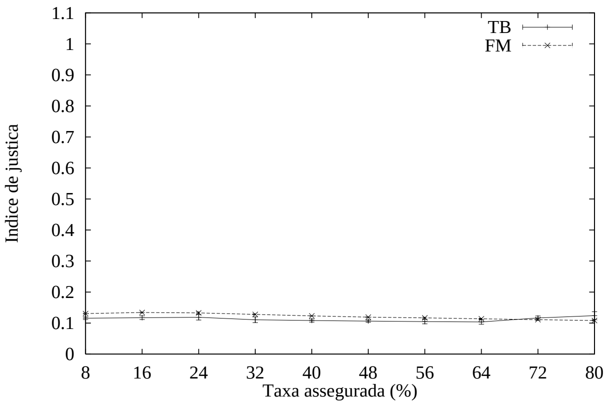
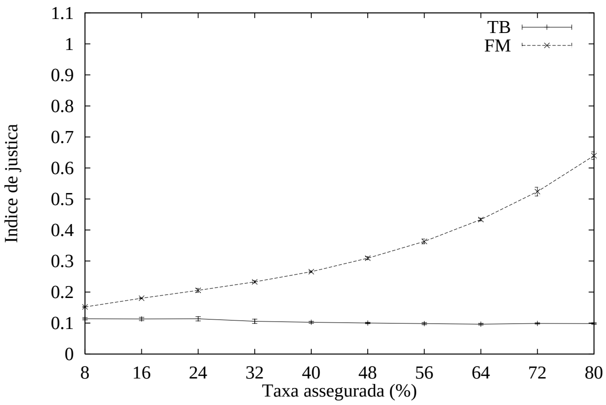


Figura 5: Vazão média das conexões TCP

Sendo assim, uma conclusão importante é que o FM demonstrou, de acordo com os resultados obtidos, ser extremamente eficiente no compartilhamento da banda assegurada possuindo um desempenho muito superior ao marcador TB. Porém, o FM carece de um mecanismo adicional para atacar o problema do compartilhamento da banda excedente ao estabelecido entre usuário e provedor de uma forma justa entre os diversos fluxos de tráfego.



(a) Largura de faixa excedente



(b) Largura de faixa total

Figura 6: Índices de justiça no compartilhamento das larguras de faixa excedente e total.

6 Conclusões

Neste trabalho, apresentamos na seção 3 uma classificação dos diferentes tipos de marcadores presentes na literatura, ressaltando a necessidade da marcação do tráfego agregado na entrada de um domínio DS e o problema de justiça entre os fluxos. Foram definidos dois critérios distintos dependendo de (i) se a marcação dos fluxos é feita utilizando informação a respeito somente dos fluxos, somente da agregação ou de um composto de ambos; (ii) em que se baseia a marcação: disponibilidade de fichas em um balde ou no nível da taxa de transmissão do tráfego.

Para obtenção de justiça entre fluxos de um tráfego agregado no Serviço Assegurado, foi apresentada uma implementação do marcador de tráfego definido em [1], a qual utiliza o algoritmo de gerenciamento ativo de filas FRED [11]. Segundo os critérios definidos na seção 3, esta implementação do FM utiliza a marcação por agregação atenta a fluxos baseada em balde de fichas segundo (i) e (ii) respectivamente.

No que se refere ao compartilhamento da largura de faixa assegurada entre fluxos de um tráfego agregado, o FM apresenta um desempenho bastante superior ao balde de fichas clássico. Porém, ficou evidenciado que este desempenho pode ser degradado em função de um ajuste inadequado dos parâmetros herdados do algoritmo FRED. Para solucionar este problema realizamos o estudo do comportamento do FM em função dos principais parâmetros do algoritmo FRED, propondo critérios bem definidos para a escolha dos valores de seus parâmetros.

Finalmente, no que se refere ao compartilhamento da largura de faixa excedente entre fluxos de tráfego agregado, o FM se mostra incapaz de assegurar justiça na medida em que nenhum tratamento diferenciado é aplicado em cima dos pacotes que não são marcados como *IN*. Como consequência deste trabalho, estudos estão sendo feitos para obtenção de um marcador justo que implemente três níveis de preferência de descarte, proporcionando justiça também no compartilhamento da largura de faixa excedente.

Referências

[1] H. Kim, "A Fair Marker," `draft-kim-fairmarker-diffserv-00`, Apr. 1999.

[2] K. Nichols, S. Blake, F. Baker, and D. L. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," *RFC 2474*, Dec. 1998.

[3] S. Blake, D. L. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," *RFC 2475*, Dec. 1998.

[4] V. Jacobson, K. Nichols, and K. Poduri, "An Expedited Forwarding PHB," *RFC 2598*, June 1999.

[5] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured Forwarding PHB Group," *RFC 2597*, June 1999.

[6] D. D. Clark and W. Fang, "Explicit Allocation of Best Effort Packet Delivery Service," *IEEE/ACM Transactions on Networking*, vol. 6, pp. 362–373, Aug. 1998.

[7] J. Ibanez and K. Nichols, "Preliminary Simulation Evaluation of an Assured Service," `draft-ibanez-diffserv-assured-eval-00`, Aug. 1998.

[8] J. Heinanen, "A Single Rate Three Color Marker," `draft-heinanen-diffserv-srtcm-01`, May 1999.

[9] J. Heinanen, "A Two Rate Three Color Marker," `draft-heinanen-diffserv-trtcm-01`, May 1999.

[10] O. Bonaventure and S. D. Cnodder, "A Rate Adaptive Shaper for Differentiated Services," `draft-bonaventure-diffserv-rashaper-01`, Oct. 1999.

[11] D. Lin and R. Morris, "Dynamics of Random Early Detection," in *Proceedings of ACM SIGCOMM'97*, (Cannes, France), Sept. 1997.

[12] Network Simulator - NS (version 2). `http://www-mash.cs.berkeley.edu/ns/`.

[13] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 397–413, Aug. 1993.

[14] M. Goyal, A. Durresi, P. Misra, C. Liu, and R. Jain, "Effect of Number of Drop Precedences in Assured Forwarding," in *IEEE/Globecomm'99*, (Rio de Janeiro), Dec. 1999.

[15] I. Yeom and A. L. N. Reddy, "Marking for QoS Improvement," in *submitted to IEEE INFOCOM'2000*, Mar. 2000.

[16] O. Elloumi, S. D. Cnodder, and K. Pauwels, "Usefullness of Three Drop Precedences in Assured Forwarding Service," `draft-elloumi-diffserv-threevstwo-00`, July 1999.

[17] J. F. de Rezende, "Assured Service Evaluation," in *IEEE/Globecomm'99*, (Rio de Janeiro), Dec. 1999.

[18] S. Sahu and D. Towsley, "A Quantitative Study of Differentiated Services for the Internet," in *IEEE/Globecomm'99*, (Rio de Janeiro), Dec. 1999.

[19] B. Nandy, N. Seddigh, P. Pieda, and J. Ethridge, "Intelligent Traffic Conditioners for Assured Forwarding Based Differentiated Services Networks," *to appear in Proceedings of Netw'2000*, May 2000.

[20] S. Sahu, P. Nain, D. Towsley, C. Diot, and V. Firoiu, "On Achievable Service Differentiation with Token Bucket Marking for TCP," *to appear in ACM SIGMETRICS*, May 2000.

[21] W. Lin, R. Zheng, and J. Hou, "How to Make Assured Services More Assured," in *Proceedings of ICNP'99*, (Toronto, Canada), Oct. 1999.

[22] A. Choudhury and E. Hahne, "Dynamic Queue Length Thresholds in a Shared Memory ATM Switch," in *Proceedings of IEEE INFOCOM'96*, (San Francisco, CA), pp. 679–687, Mar. 1996.

[23] R. Jain, *The Art of Computer Systems Performance Analysis.* John Wiley and Sons Inc., 1991.

[24] R. K. Jain, D.-M. W. Chiu, and W. R. Hawe, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer System," *submitted for publication to ACM Transaction on Computer Systems*, Sept. 1984.