Um Método para Verificação do Funcionamento de Filas RED

Marcus Vinicius Soldera Grando[1] **, Germano Veit Michel**[1] **, Sérgio Luis Cechin**[1]

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{mvsgrando,gvmichel,cechin}@inf.ufrgs.br

Abstract. *This work presents a verification methodology to guarantee that a Random Early Detect(RED) queue discipline behaves as expected. REDs are commonly used in QoS solutions for computer networks. This methodology is platform and implementation independent and can be used in a wide array of systems.*

Resumo. *Este trabalho introduz uma metodologia para verificar o correto funcionamento das disciplinas de enfileiramento Random Early Detection (RED), comumente usadas em soluções QoS em redes de computadores. Esta metodologia tem como característica ser independente de plataforma e implementação, podendo ser usada nas mais diversas configurações.*

1. Introdução

Os trabalhos, como os citados na seção 2, com filas RED geralmente verificam o seu efeito prático na solução dos problemas encontrados em redes de computadores, mas não buscam demonstrar uma metodologia para verificar se a fila está funcionando de acordo com os parâmetros configurados.

A necessidade de identificar o correto comportamento das filas *Random Early Detection* (RED), em um caso prático, ocasionou a oportunidade de desenvolver um novo método para isso. A solução encontrada foi definir um cenário de teste onde fosse possível controlar e inferir o tamanho da fila em qualquer instante de tempo. Além de diagnosticar se cada pacote foi despachado, enfileirado ou descartado. Para tanto, foram utilizados fluxos de datagramas UDP [RFC 768] e controle de vazão de saída no roteador.

Conseguir verificar este funcionamento é importante para assegurar que as implementações disponíveis estão de acordo com a especificação das filas RED. Inferir o funcionamento interno torna isso possível e mostra de maneira mais clara o correto funcionamento da fila.

O restante deste artigo é organizado da seguinte maneira: Na seção 2 a seguir estão relacionados alguns trabalhos sobre filas RED, na seção 3 é apresentado os conceitos e o funcionamento teórico da política de filas RED, a seção 4 descreve o cenário de testes e método proposto. Por fim a seção 5 apresenta os resultados obtidos na execução desses testes.

2. Trabalhos relacionados

Existem várias pesquisas mostrando as filas *Random Early Drop* como um mecanismo para evitar congestionamentos. [Hashem] discute algumas vantagens e desvantagens de roteadores com *Random Drop* e *Drop Tail* e investiga brevemente roteadores com RED.

O trabalho de [Elloumi] mostra os benefícios do uso de filas RED, dentre eles: grandes vazões, atrasos pequenos e justiça, além de mostrar que ela favorece os fluxos com pouca quantidade de dados.

Não foi encontrado nenhuma referência em que se mostra o funcionamento interno da fila RED, ou seja, relacionar a quantidade de pacotes com a quantidade de pacotes descartados, fato que levou ao desenvolvimento deste trabalho.

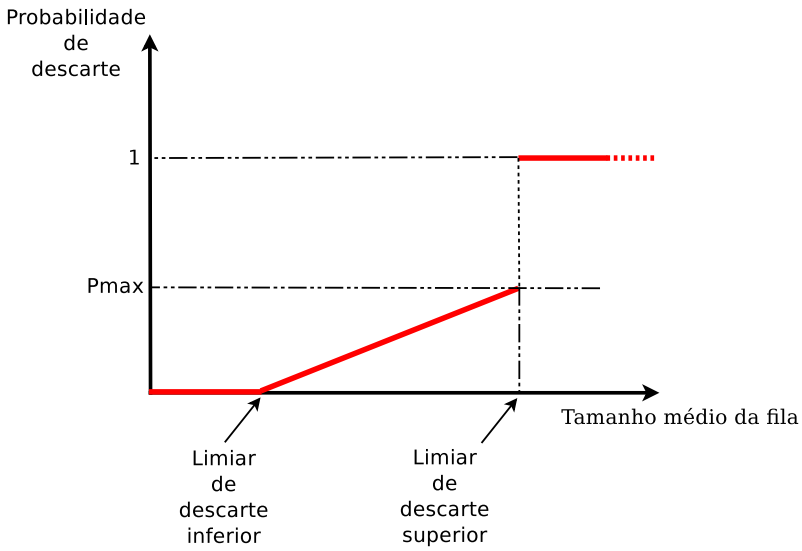
3. Filas RED

Random Early Detection é uma política de gerenciamento de filas que age de maneira ativa, ou seja, descarta pacotes antes mesmo de estar cheia. Isto é feito visto que uma das características das conexões TCP [RFC 793] é de adaptar-se, diminuindo a taxa de envio ao notar perda de pacotes. Assim uma saturação da banda é evitada antes de acontecer.

As filas RED também evitam que fluxos de grande vazão monopolizem a banda disponível. Como estes enviam grande quantidade de pacotes há uma alta probabilidade de que seus pacotes sejam marcados para descarte, diminuindo assim a sua vazão. A seguir, está descrito o algoritmo usado pelas filas RED.

3.1. O algoritmo RED

A utilização da política de filas RED só faz sentido quando o tráfego que passa pela fila é do tipo TCP, pois este tem um mecanismo de adaptação de fluxo que atua quando percebe perdas. Com isso o descarte antecipado do algoritmo tem validade.



.

Figura 1. Curva característica

O algoritmo RED funciona da seguinte maneira: a cada pacote que ingressa no roteador é calculado o tamanho médio da fila usando um filtro passa-baixa com uma média móvel exponencial [Floyd]. Este valor é então comparado com dois limiares: limiar máximo e limiar mínimo. Quando o tamanho médio da fila é menor que o limiar mínimo o pacote não é descartado e quando é maior que o limiar máximo o pacote é descartado. Se o tamanho médio da fila está entre os limiares máximo e mínimo o pacote é ou não

descartado de acordo com uma função de probabilidade. Os pacotes não descartados são enfileirados e portando alteram o tamanho da fila.

A função de probabilidade é uma curva linear que começa em zero no limiar mínimo e chega até um valor (Pmax) estipulado pelo usuário no limiar máximo como pode ser visto na figura 1.

4. Testes

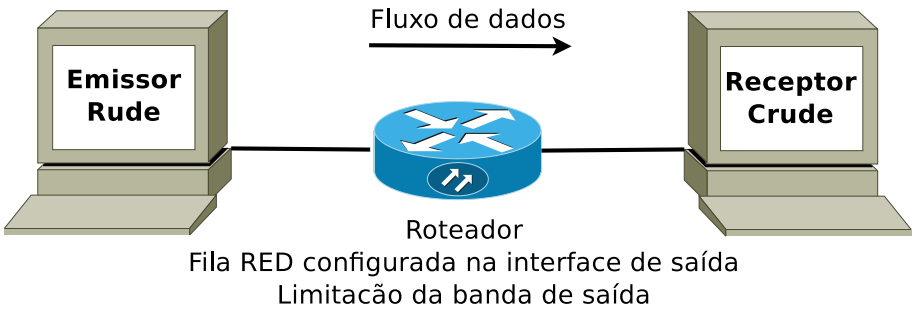
Para identificar o funcionamento de uma fila RED é necessário conhecer o tamanho médio instantâneo da fila e quantos pacotes foram descartados. Sem ter acesso a variáveis do roteador é preciso inferir o tamanho médio da fila e para saber quando os pacotes foram descartados é necessário enviá-los ordenadamente e computar sua chegada.

Na seção a seguir será mostrado o cenário de teste utilizado e após os princípios que validam esse teste.

4.1. Cenário de Teste

O cenário de teste utilizado é o ilustrado na figura 2, os seguintes equipamentos e ferramentas foram usados para a montagem deste cenário:

- Um roteador com kernel Linux 2.6.32.10 [Linux];
- Dois computadores equipados com processador Intel Core 2 Quad Q6600, 4GB de memória RAM, placa de rede gigabit e rodando sistema operacional Linux (kernel 2.6.32.10);
- Ferramentas para geração de tráfego RUDE e CRUDE [RUDE].



.

Figura 2. Cenário de Teste

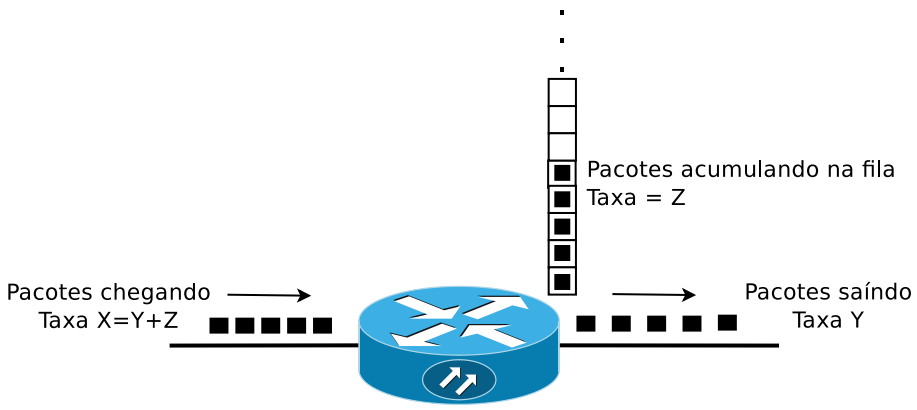
Para esta metodologia, são irrelevantes questões de sincronização e tempo pois, para a apuração dos resultados, não são levados em conta os tempos absolutos, mas apenas o intervalo relativo entre dois instantes. Por exemplo, o emissor envia pacotes P1,P2....P10 no intervalo de tempo entre T1 e T2. No receptor, são recebidos os pacotes P1,P2...P8 num intervalo de tempo entre T3 e T4. Não interessam os valores absolutos de T1,T2,T3 e T4, portanto, os relógios das duas máquinas podem estar não sincronizados. O que realmente importa é que os intervalos entre T1 e T2 e entre T3 e T4 sejam iguais. Possibilitando inferir que os pacotes P9 e P10 foram descartados ou ainda não chegaram.

4.2. Funcionamento da Metodologia de Teste

Embora a RED seja orientada para conexões TCP o fluxo utilizado para os testes foi com datagramas UDP [RFC 768]. Este foi utilizado pois o emissor envia os pacotes a uma taxa constante que não é reduzida devido a perdas. A ideia principal do teste é limitar a banda de saída do roteador em uma taxa X e enviar os pacotes em uma taxa Y, onde Y é ligeiramente maior que X. Dessa forma os pacotes irão se acumular, aos poucos, na fila, possibilitando inferir o número de pacotes descartados para várias tamanhos de fila.

A figura 3 apresenta um esquemático de como o teste funciona. É possível notar os pacotes chegando no roteador, alguns saindo na interface que possui limitação de velocidade e outros acumulando na fila. Eventuais pacotes que são descartados da fila estão omitidos nesta figura pois esta tem o propósito de demonstrar como os pacotes se acumulam na fila.

Para obter dados e demonstrar o funcionamento da fila RED é necessário que o envio de dados dure tempo suficiente para encher a fila, ou seja, ultrapassar o limiar máximo. Além disso é necessário que a taxa de envio seja suficientemente maior que a velocidade do enlace de saída a fim de suprir os pacotes descartados pelo gerenciador da RED. Infelizmente não é fácil calcular um valor exato de diferença dessas velocidades de acordo com os parâmetros da fila mas não é muito difícil executar alguns testes e notar quais os valores apropriados. Nos testes executados a velocidade extra necessária foi em torno de 25% a mais que a velocidade do enlace de saída do roteador, podendo esse ser um valor tomado como referência para testes futuros.



.

Figura 3. Acumulação de pacotes na fila antes do limiar mínimo

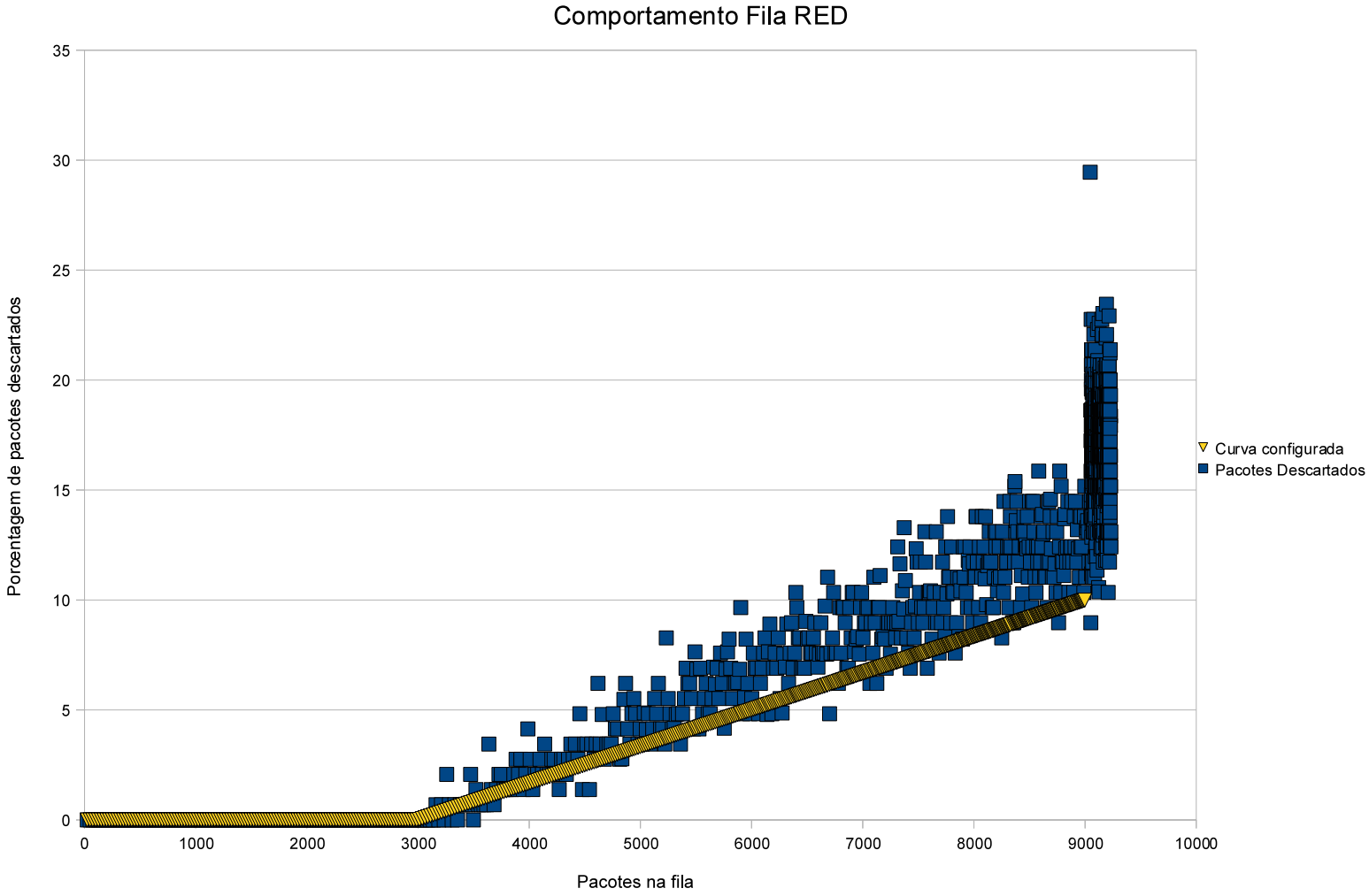
5. Resultados obtidos

Com a execução de alguns testes é possível adaptar os valores de velocidade de saída do enlace, velocidade de envio, limiar mínimo e máximo e tamanhos dos pacotes para conseguir resultados interessantes e para a duração da execução do teste ser próximo da desejada. Os resultados a seguir foram obtidos com a seguinte configuração:

- Velocidade do enlace de saída: 100kbps;
- Velocidade de envio dos pacotes: 125kbps;
- Limiar mínimo: 3000 pacotes;

- Limiar máximo: 9000 pacotes;
- Tamanho dos pacotes: 100 bytes.

O software RUDE fornece um *log* bastante completo que tornou mais fácil a análise dos dados. Esta foi feita com o auxílio de um *script* para interpretar o *log*, que funciona da seguinte maneira: recorta o *log* em intervalos de um segundo (tempo no emissor), contabiliza quantos pacotes foram enviados, contabiliza a quantidade de pacotes que foram descartados e infere quantos pacotes que estão na fila para cada segundo.



.

Figura 4. Resultados do teste da fila RED

No final têm-se um resumo para cada segundo de teste com os pacotes na fila e os descartados tornando assim possível traçar uma curva de teste e comparar com a curva configurada e assim observar se os parâmetros configurados condizem com o comportamento teórico da fila.

Na figura 4 está traçado o gráfico do teste executado, este ilustra o comportamento da disciplina de enfileiramento em relação ao descarte de pacotes considerando o número de pacotes enfileirados. Nela o eixo X corresponde a quantidade de pacotes na fila e o eixo Y corresponde a porcentagem de pacotes descartados. Está também traçada a curva teórica de descarte, como configurada no roteador.

Como esperado, entre 0 e 3000 pacotes na fila nenhum que ingressava foi descartado. No intervalo entre 3000 e 9000 pacotes na fila há um percentual de descarte condizente com os parâmetros utilizados, ou seja, 0% em 3000 pacotes que sobe linearmente até atingir 10% em 9000 pacotes. Após 9000 pacotes na fila todos os pacotes são descartados, fazendo com que o número de pacotes na fila diminua rapidamente. Devido a isso não é possível observar 100% de pacotes descartados no gráfico para valores de pacote na fila maiores que 9000.

Os testes apontaram um erro médio menor que 2 pontos percentuais, o que torna os resultados obtidos bastante satisfatórios. Pode-se identificar então que o funcionamento da fila é semelhante ao configurado, verificando assim que a implementação das filas RED no Linux está de acordo com a especificação, conforme esperado.

6. Conclusão

Os dados obtidos pelo método são inferências e não medidas internas. Isto é bom pelo lado de não atrelar o método a implementações existentes e ruim pelo lado de não fornecer uma precisão ótima. Entretanto, por se tratarem de números pequenos e difíceis de medir, os erros introduzidos pela abordagem proposta não comprometem os resultados. Além disso, uma precisão maior não agregaria muito valor aos testes.

Portanto, baseado nos resultados mostrados neste artigo, é possível identificar que a metologia apresentada serve, de fato, para verificar o correto funcionamento de filas RED.

Referências

O. Elloumi and H. Afifi. Dept. of Networks and Multimedia, Ecole Nat. Superieure des Telecommun. de Bretagne, Brest **RED Algorithm in ATM Networks** IEEE ATM Workshop 1997. Proceedings, p.312-319 May 1997

S. Floyd and Van Jacobsen. **Random Early Detection Gateways for Congestion Avoidance**. IEEE/ACM Transactions on Networking, Aug 1993, Issue 4, p.397-412

Hashem, E. **Analysis of random drop for gateway congestion control** Report LCS TR-465, Laboratory for Computer Science, MIT, Cambridge, MA, 1989, p.103.

The Kernel Linux Archive. http://www.kernel.org/ Acessado em Julho de 2010.

J. Postel. **RFC 793 - Transmission Control Protocol**. 1981

J. Postel. **RFC 768 - User Datagram protocol**. 1980

RUDE - Real-Time UDP Data Emitter. http://rude.sourceforge.net Acessado em Julho de 2010.