

TP3: Estudo de *QoS* e Mecanismos de Controlo de Tráfego em redes IP

João Guedes (PG47329), Paulo Sousa (PG47556), and Renata Teixeira (PG47603)

Universidade do Minho, Braga, Portugal
Mestrado em Engenharia Informática
Qualidade de Serviço em Redes IP
Grupo 10
22 de Abril de 2022

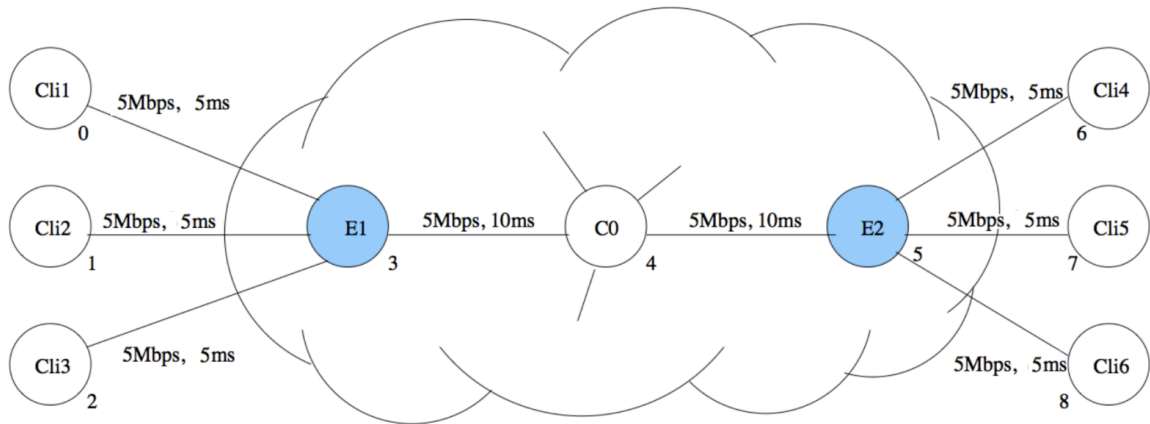


Figura 1: Topologia de Rede

1 Identifique os links em congestão.

Tendo em consideração que, observando a topologia apresentada na Figura 1, as ligações de acesso dos clientes têm capacidade de 5 Mbps e que as ligações entre os dois *routers* de *edge* e o *router* de *core* têm, também, capacidade de 5 Mbps, e sabendo que os 6 fluxos: 0->8, 1->7, 2->6, 8->0, 7->1 e 6->2 têm uma taxa constante de 3 Mbps, podemos afirmar que as ligações entre os *routers* de *edge* e o *router* de *core* estarão congestionadas.

Atentando no facto de que os links de acesso dos clientes têm capacidade de 5Mbps e estão apenas a utilizar 3Mbps, estes não se encontram congestionados, no entanto, os *router* E1 e E2 têm 3 links para 3 clientes respetivamente, o que faz com que as suas taxas de entrada sejam de 9Mbps, sendo um valor superior à capacidade dos seus links com o *router* de *core*(5Mbps), fazendo com que, por isso, estes links estejam congestionados.

- 2 Usando os scripts `loss.awk` e `bw.awk`, obtenha os gráficos que ilustram os níveis de perda e utilização da largura de banda ao longo do tempo.

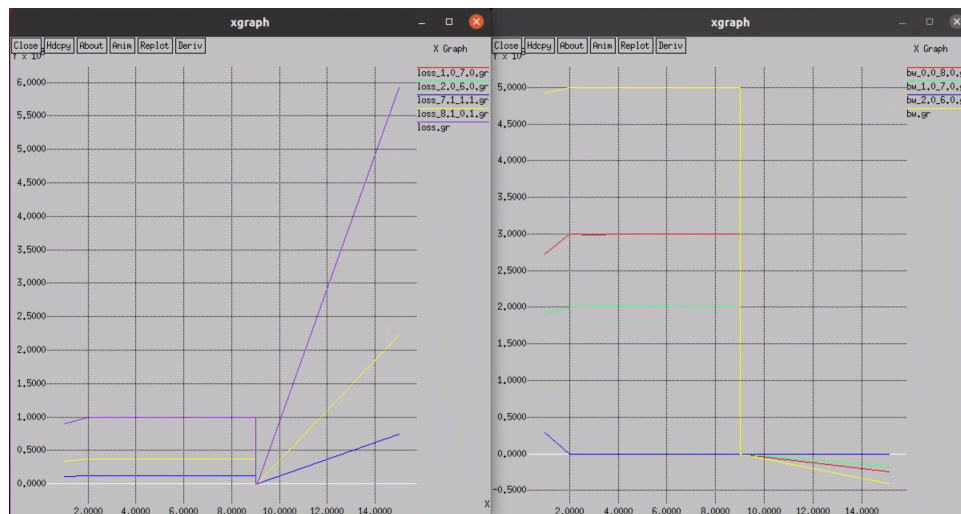


Figura 2: Gráficos de Perda (esquerda) e Utilização de Bandwidth(direita)(Tail Drop)

O *Drop Tail* é um algoritmo simples de gestão de filas, que controla o comprimento das filas de espera definindo um comprimento máximo para cada fila no *router*. Enquanto o comprimento da fila é menor que o comprimento máximo, todos os pacotes são aceites, no entanto se a fila atinge o seu comprimento máximo, todos os pacotes de entrada subsequentes são descartados até ao comprimento da fila diminuir de forma a ser menor que o comprimento máximo.

As fontes detectando perdas, abrandam a transmissão dos pacotes, nesse ponto, a utilização da largura de banda normalmente cai significativamente, as filas ficam menos congestionadas e os fluxos começam a aumentar os seus tamanhos de janela. Eventualmente, as interfaces ficam congestionadas novamente, os *drops* acontecem e o ciclo repete-se, esta situação é chamada de sincronização global.

No entanto, e como é possível verificar pela Figura 2, este algoritmo tem outra desvantagem, não faz *drops* de forma diferenciada ou preferencial. Como se pode verificar pela utilização de *bandwidth*, há uma grande discrepância nestes valores que o algoritmo é incapaz de resolver, pois, quando as filas ficam cheias e ocorre o *tail drop*, os fluxos mais agressivos não são punidos seletivamente.

- 2.1 Agora, altere as filas associadas aos links em congestionamento de DropTail para RED e obtenha os resultados e gráficos correspondentes. Comente os resultados.

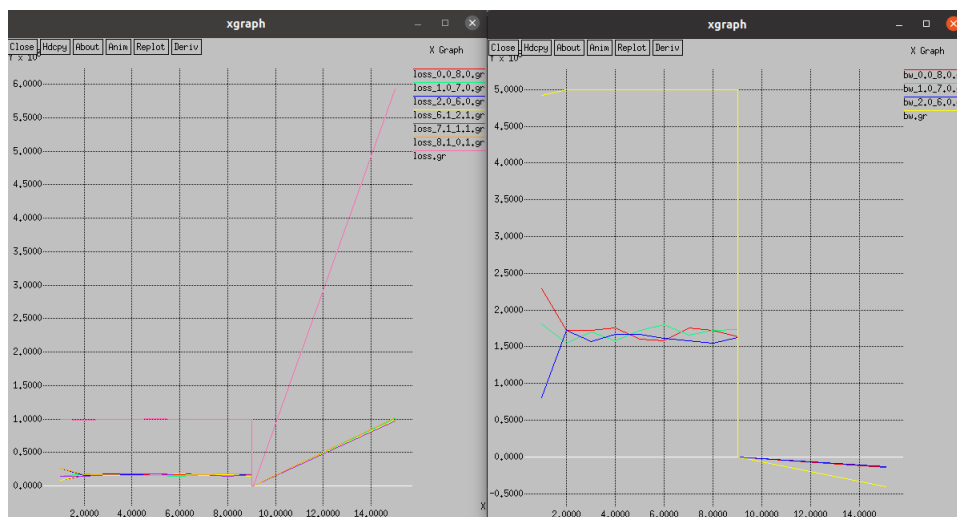


Figura 3: Gráficos de Perda(esquerda) e Utilização de Bandwidth(direita)(RED)

O *Droptail* distribui o espaço do *buffer* de forma injusta entre os fluxos de tráfego, podendo também levar à sincronização global do TCP, visto que todas as conexões TCP "retêm" simultaneamente e, em seguida, avançam simultaneamente. As redes tornam-se subutilizadas e inundadas – alternadamente, em ondas, como já descrito anteriormente. Por sua vez, o RED descarta pacotes selecionados aleatoriamente antes que a fila fique cheia. A taxa de perda aumenta à medida que o tamanho da fila cresce; melhor dito, à medida que o tamanho da fila aumenta, aumenta também a probabilidade de descartar os pacotes recebidos.

O RED não diferencia fluxos; não é orientado a fluxo. Basicamente, como o RED seleciona os pacotes a serem descartados aleatoriamente, é (estatisticamente) esperado que os pacotes pertencentes a fluxos agressivos (de alto volume) sejam descartados mais do que os pacotes dos fluxos menos agressivos.

Observando a Figura 3 e o gráfico relativo à utilização de largura de banda, é possível observar que os respectivos valores dos vários fluxos são bastante mais próximos, havendo uma melhor distribuição da largura de banda, comprovando que de facto são mais descartados pacotes de fluxos mais agressivos, tornando a utilização de *bandwidth* mais equilibrada entre os diversos fluxos.

- 2.2 Em um cenário mais realista, seria de esperar tráfego UDP e TCP com outras características (FTP, HTTP, etc.). Utilizando os procedimentos já incluídos no script de simulação, altere-os para obter o seguinte cenário:
- Uma aplicação CBR enviando 3 Mbps do cliente 1 para o cliente 6 e outro do cliente 6 para o cliente 1.
 - Uma conexão FTP do cliente 2 para o cliente 5 e outra do cliente 4 para o cliente 2.
 - A conexão de voz sobre UDP do cliente 3 para o cliente 4 e vice-versa.
- Identifique quais são as aplicações mais afetadas pelo congestionamento? E porquê?

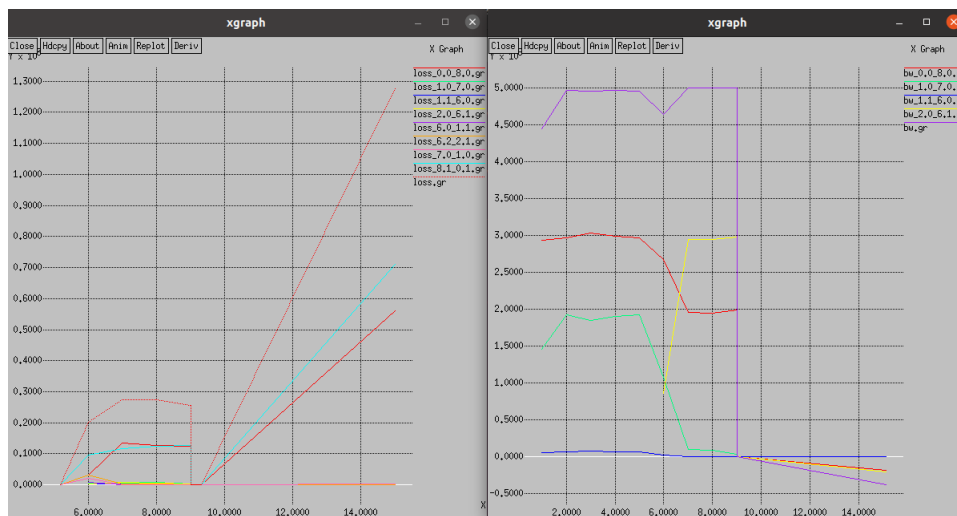


Figura 4: Gráficos de Perda(esquerda) e Utilização de Bandwidth(direita) - Novo Cenário

Através da observação dos gráficos, é possível verificar que os maiores níveis de perda e de menor utilização de largura de banda dão-se na conexão FTP e na aplicação CBR, respetivamente.

3 Com base no código do *script* e no manual, e para o link E1 - C0 (ou C0 - E2), identifique:

1. o número de filas existentes e o agendador de tráfego em uso;
2. a disciplina de filas em uso e a configuração de cada fila;
3. a quantidade de memória alocada para as filas;
4. as filas que manipulam fluxos de dados;

1. O número de filas existente é de 2, isto é, pela variável *numQueues*. concluímos que existe 1 fila física com 2 filas virtuais, como pudemos constatar através da variável *setnumprec*. No caso do agendador de tráfego, na fila de E1-C0 é usada o FIFO, pois é o que é usado por default.

2. A disciplina de filas em uso na fila de E1-C0 é usada o "Random Early Detection"(RED). Além disso, também são configuradas as 2 filas virtuais, sendo que a primeira é configurada tendo 20, 40 e 0.02 de *minimum threshold*, *maximum threshold* e *maximum dropping probability* respetivamente. Já a segunda fila apresenta 10 de *minimum threshold*, 20 de *maximum threshold* e 0.10 de *maximum dropping probability*.

- 3.
 - 4.
- 4 Considerando os resultados/estatísticas da simulação, identifique a fila que sofre maior perda de pacotes. Tente justificar esse comportamento.

Estatísticas da Fila de Espera de E1 para C0				
Packets Statistics				
CP	TotPkts	TxPkts	ldrops	edrops
--	-----	-----	-----	-----
All	5619	3136	2097	386
10	1252	1252	0	0
11	4367	1884	2097	386

Estatísticas da Fila de Espera de C0 para E2				
Packets Statistics				
CP	TotPkts	TxPkts	ldrops	edrops
--	-----	-----	-----	-----
All	3111	3111	0	0
10	1242	1242	0	0
11	1869	1869	0	0

Figura 5: Tabela das Estatísticas das filas

Como podemos verificar pela figura 5, a fila de espera de E1 para C0 é onde existe a maior perda de pacotes. Isto acontece pois é onde está situado o ponto de bottleneck, isto é, o link não aguenta o tráfego todo que vem dos clientes. Além disso, também conseguimos verificar que a fila de espera de C0 para E2 não tem qualquer perda de pacotes pois como é fácil de perceber, tendo as ligações entre E1, C0 e E2 a mesma bandwidth, todo o tráfego que consegue "passar" em E1, também irá conseguir fazê-lo em E2.

Sendo esta topologia simétrica, o mesmo acontece para os links simétricos.

- 5 Gere os gráficos que reflectam a perda de pacotes e a utilização da largura de banda ao longo do tempo. Comente os resultados quando comparado ao cenário de melhor esforço e explique as diferenças.

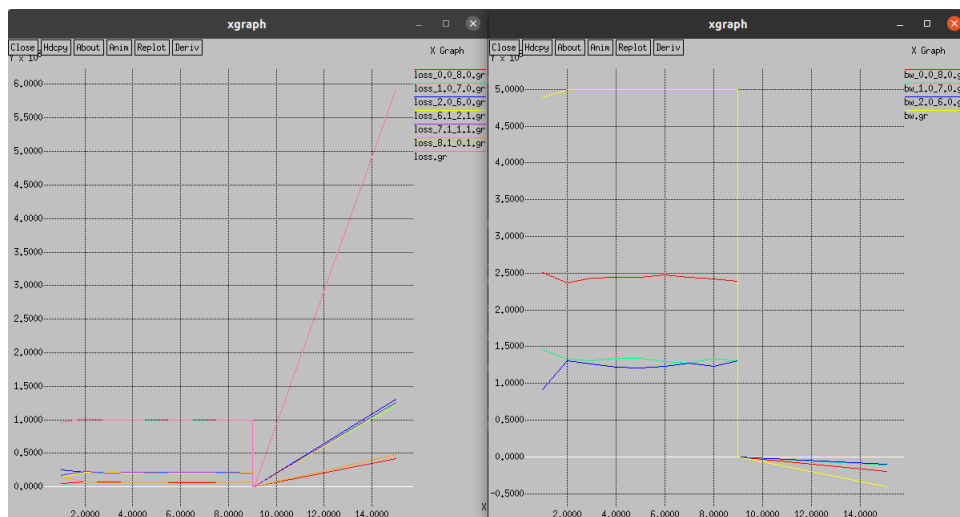


Figura 6: Gráficos de Perda(esquerda) e Utilização de Bandwidth(direita) - qos1

Comparando a figura 2 e figura 6, conseguimos verificar uma descida da quantidade de perda de tráfego no link o C1 para o C6, devendo-se ao facto de se ter sido dada uma prioridade maior de pacotes que transitam de C1 para C6 e vice-versa neste novo script. Além disso, conseguimos também observar uma melhor divisão da bandwidth utilizada.

- 6 Suponha que o fornecedor de serviços pretenda implementar a seguinte política:
- Para garantir 30% de capacidade para clientes com características idênticas (a capacidade total pode ser usada se disponível).
 - O tráfego que excede a taxa negociada deve ser reclassificado, ou seja, encaminhado com prioridade mais baixa.
- Espera-se que os alunos proponham e justifiquem uma solução concreta para implementar essa política, usando mecanismos de controlo de tráfego adequados disponíveis no NS-2. Inclua resultados de simulação ilustrativos e discuta os prós e contras da solução proposta da perspectiva do provedor de serviços.

```
# Definição dos parâmetros
set packetSize 1000 ;# em bytes
set cir0 1500000 ;# em bits/seg
set cbs0 5000 ;# em bytes

# Set DS RED parameters from Edgel to Core:
sq(E1C0) meanPktSize $packetSize
sq(E1C0) set numQueues 1
sq(E1C0) setNumPrec 2 -
sq(E1C0) addPolicyEntry [$Cli1 id] [$Cli6 id] TokenBucket 10 $cir0 $cbs0
sq(E1C0) addPolicyEntry [$Cli2 id] [$Cli5 id] TokenBucket 10 $cir0 $cbs0
sq(E1C0) addPolicyEntry [$Cli3 id] [$Cli4 id] TokenBucket 10 $cir0 $cbs0
sq(E1C0) addPolicyEntry -1 -1 Dumb 11 ;# todo o restante trafego...
sq(E1C0) addPolicerEntry TokenBucket 10 11
sq(E1C0) addPolicerEntry Dumb 11
sq(E1C0) addPHBEntry 10 0 0
sq(E1C0) addPHBEntry 11 0 1
sq(E1C0) config0 0 0 20 40 0.02
sq(E1C0) config0 0 1 10 20 0.10

# Set DS RED parameters from Edge2 to Core:
sq(E2C0) meanPktSize $packetSize
sq(E2C0) set numQueues 1
sq(E2C0) setNumPrec 2 -
sq(E2C0) addPolicyEntry [$Cli6 id] [$Cli1 id] TokenBucket 10 $cir0 $cbs0
sq(E1C0) addPolicyEntry [$Cli5 id] [$Cli2 id] TokenBucket 10 $cir0 $cbs0
sq(E1C0) addPolicyEntry [$Cli4 id] [$Cli3 id] TokenBucket 10 $cir0 $cbs0
sq(E2C0) addPolicyEntry -1 -1 Dumb 11
sq(E2C0) addPolicerEntry TokenBucket 10 11
```

Figura 7: Código do problema

A figura 7 mostra o código alterado, onde foram adicionadas novas entradas na tabela de políticas, dando igual prioridade aos clientes com características idênticas, utilizando o Token-Bucket como policer. Assim, todo o tráfego que excede o "commit information rate", é "marcado" com uma prioridade mais baixa. Pela figura 8 conseguimos confirmar que após esta alteração, a bandwidth é distribuída uniformemente, não evitando, no entanto, a perda de pacotes.

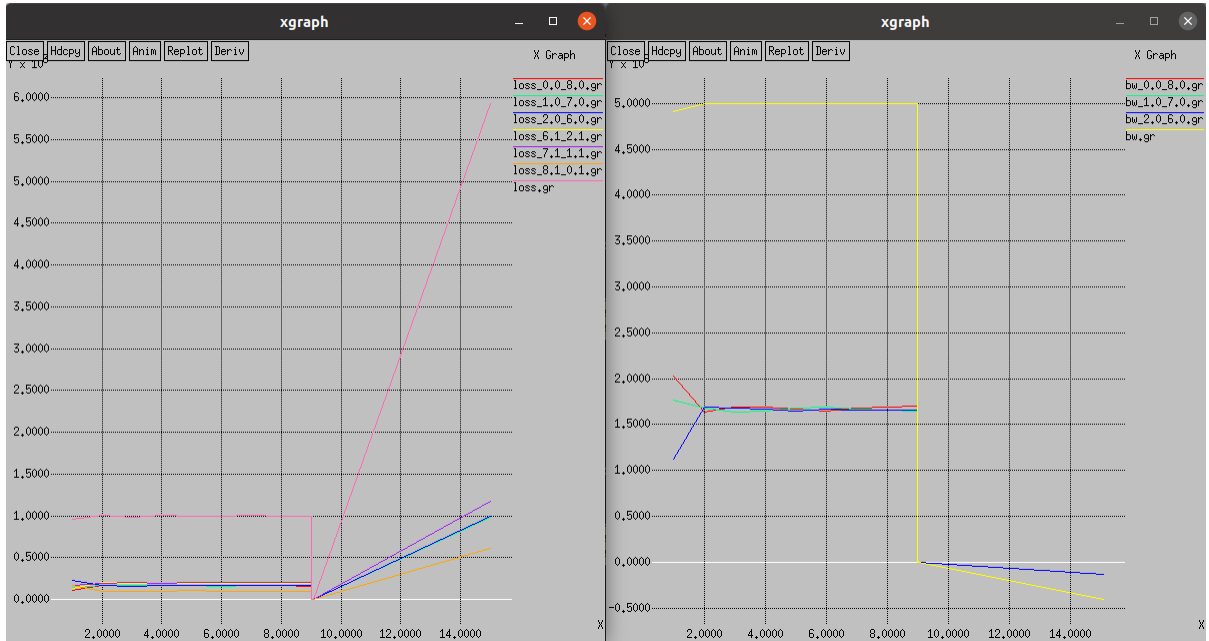


Figura 8: Gráficos de Perda(esquerda) e Utilização de Bandwidth(direita) após a alteração do código