Universidade do Estado de Santa Catarina
Centro de Ciências Tecnológicas
Departamento de Ciência da Computação
Redes de Computadores – 2021/02
Prof. Guilherme Koslovski

Trabalho prático sobre análise de desempenho do TCP

A execução do trabalho permitirá a aplicação prática de alguns conceitos e tecnologias estudados durante o semestre. Assim, iremos estudar o desempenho do protocolo TCP em um ambiente controlado. Inicialmente, é preciso lembrar que o protocolo TCP é dependente de fatores internos à rede. Por exemplo, para implementar a identificação e controle de congestionamento nas extremidades, o TCP infere informações sobre os canais de comunicação intermediários. É plausível admitir que o TCP oferece um serviço confiável genérico, ou seja, nenhuma aplicação alvo foi analisada para definição do protocolo, mas sim a justiça no compartilhamento dos recursos entre diversos fluxos.

O trabalho compreenderá a comparação de no mínimo 3 algoritmos para controle de congestionamento do TCP. Quanto ao desenvolvimento, o trabalho poderá ser desenvolvido em duplas (no máximo).

1) Como entregar o trabalho?

a) Resultados e relatório

Elabore um relatório detalhando o cenário composto, os gráficos e os resultados obtidos. O relatório deve explicar os algoritmos selecionados (para algoritmos novos, a explicação deve ser consultada na documentação oficial – ou artigo).

- Discuta os resultados e indique suas conclusões.
- Qual algoritmo teve o melhor comportamento?
- Qual algoritmo tem maior sensibilidade a segmentos perdidos?
- Qual algoritmo tem maior sensibilidade aos atrasos?

b) Ferramentas auxiliares utilizadas

Agrupe e compacte os arquivos auxiliares que você utilizou para coletar os resultados (por exemplo, *scripts*, arquivos de dados, entre outros). Envie o arquivo compactado junto com o relatório final.

c) Data e procedimento para entrega

17/Fevereiro/2022 23:55 pelo Moodle. O prazo não será alterado.

2) Como montar um ambiente de testes?

Um dos desafios do trabalho está relacionado com o gerenciamento do ambiente de testes no seu computador pessoal. Se optar, o trabalho pode ser desenvolvido em uma máquina virtual. Inicialmente, é necessário realizar a instalação da ferramenta iperf3 (versão 3).

Um estudo detalhado sobre os parâmetros do iperf3 demonstrará como ele pode ser utilizado para montar os cenários experimentais. Ferramenta disponível em: https://iperf.fr/iperf-download.php. Documentação disponível em: https://iperf.fr/iperf-doc.php.

Observe, por exemplo, o parâmetro -Z fornecido pelo iperf3. Ele permite a seleção do algoritmo para controle de congestionamento que deve ser usado para aquela conexão em particular. A figura abaixo apresenta um exemplo de saída do *iperf*. Note que a vazão contabilizada pela aplicação é apresentada na última coluna.

```
[ ID] Interval           Transfer     Bandwidth
[  3]   0.0- 1.0 sec   290 MBytes  2.43 Gbits/sec
[  3]   1.0- 2.0 sec   302 MBytes  2.53 Gbits/sec
[  3]   2.0- 3.0 sec   311 MBytes  2.61 Gbits/sec
[  3]   3.0- 4.0 sec   310 MBytes  2.60 Gbits/sec
[  3]   4.0- 5.0 sec   307 MBytes  2.58 Gbits/sec
[  3]   5.0- 6.0 sec   308 MBytes  2.58 Gbits/sec
[  3]   6.0- 7.0 sec   307 MBytes  2.58 Gbits/sec
[  3]   7.0- 8.0 sec   307 MBytes  2.58 Gbits/sec
[  3]   8.0- 9.0 sec   316 MBytes  2.65 Gbits/sec
[  3]   9.0-10.0 sec   308 MBytes  2.59 Gbits/sec
[  3]  10.0-11.0 sec   308 MBytes  2.58 Gbits/sec
[  3]  11.0-12.0 sec   310 MBytes  2.60 Gbits/sec
[  3]  12.0-13.0 sec   308 MBytes  2.59 Gbits/sec
[  3]  13.0-14.0 sec   308 MBytes  2.59 Gbits/sec
[  3]  14.0-15.0 sec   306 MBytes  2.57 Gbits/sec
[  3]  15.0-16.0 sec   308 MBytes  2.58 Gbits/sec
[  3]  16.0-17.0 sec   306 MBytes  2.56 Gbits/sec
```

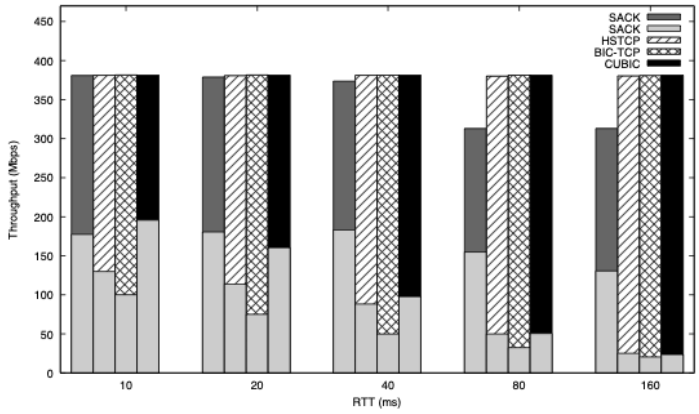
3) Dicas para o desenvolvimento do trabalho

a) Quais algoritmos para controle de congestionamento estão disponíveis em meu computador?
Execute "cat /proc/sys/net/ipv4/tcp_allowed_congestion_control" para obter a lista dos algoritmos previamente configurados. A opção -Z do *iperf3* permite a seleção do algoritmo.

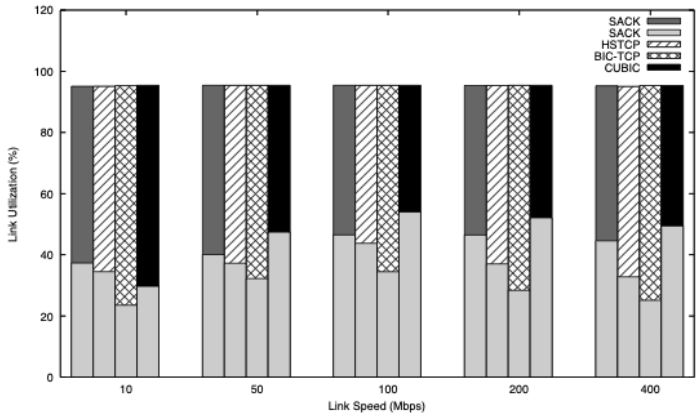
b) Como executar 2 ou mais *iperfs* em paralelo?
Será necessário controlar as portas de origem e destino da ferramenta iperf3.

4) O que será comparado?

O gráfico abaixo demonstra um exemplo de comparação obtido do artigo que apresentou o algoritmo Cubic.



(a) Bandwidth 400Mbps with varying RTT



(b) RTT 10ms with varying bandwidth

Observe na figura que os algoritmos foram comparados (par de algoritmos) em cenários com configurações específicas. Por exemplo: em (a) existia um limite de 400Mbps para ser compartilhado entre os 2 pares com uma configuração de RTT variada. Em (b) o RTT era fixo (10ms), aplicado para cenários distintos de vazão (Mbps). Maiores detalhes sobre os gráficos usados como exemplo (CUBIC é a versão padrão para Windows e Linux): https://dl.acm.org/doi/10.1145/1400097.1400105

a) Execução dos algoritmos com perda de segmentos

Para forçar a perda de dados na sua conexão, execute em seu computador:

sudo tc qdisc add dev lo root netem loss 0.1

Após informar sua senha, o equipamento passou a ter uma perda de dados simulada de 0.1% (opção loss 0.1). Note que lo representa sua interface local.

Quando concluir o teste, apague a configuração com

sudo tc qdisc del dev lo root

Observações:
- Leia sobre netem. É uma ótima opção para simular problemas em redes (https://wiki.linuxfoundation.org/networking/netem).

b) Execução dos algoritmos com perda de segmentos e atraso de comunicação
Para forçar a perda de dados e atraso na sua conexão, execute em seu computador:

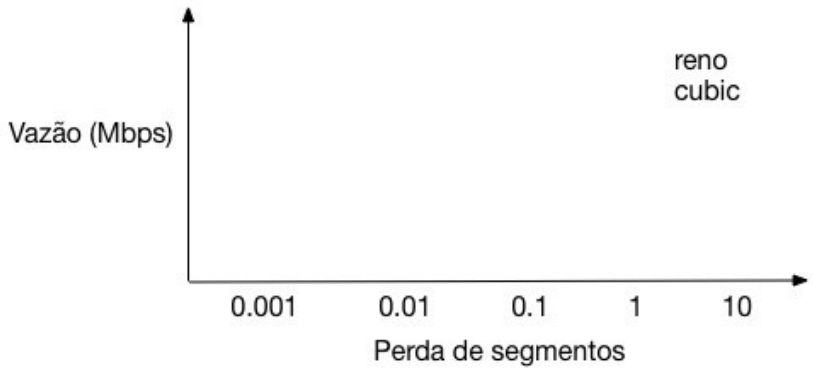
sudo tc qdisc add dev lo root netem loss 0.1 delay 12.5ms

Após informar sua senha, o equipamento passou a ter uma perda de dados simulada de 0.1% (opção loss 0.1) e um atraso simulado de 12.5ms (opção delay 12.5ms). Se o comando não funcionar no seu equipamento, leia a documentação sobre netem.

5) Como preparar os resultados?

O objetivo do trabalho é comparar o desempenho dos algoritmos (reno, cubic ou outro que você selecionar) frente as variações de perda de dados e atraso. Assim, monte o cenário e colete dados para compor os seguintes gráficos

a) Como preparar os gráficos?
O gráfico abaixo apresenta um exemplo de comparação entre *reno* e *cubic*. Você pode (deve) usar outros gráficos.



b) Gráficos que devem ser preparados

i) Compare a vazão de 3 ou mais algoritmos (por exemplo, *reno* e *cubic*) frente ao aumento da perda de segmentos.

ii) Compare a vazão de 3 ou mais algoritmos (por exemplo, *reno* e *cubic*) frente ao aumento da perda de segmentos e atraso de constante de 12.5ms.

iii) Compare a vazão de 3 ou mais algoritmos (por exemplo, *reno* e *cubic*) frente ao aumento da perda de segmentos e atraso de constante de 100ms.

iv) Compare a vazão de 3 ou mais algoritmos (por exemplo, *reno* e *cubic*) frente ao aumento da perda de segmentos e atraso de constante de 200ms.

c) Desafio
Repita a letra (b) limitando as configurações de vazão do netem (10Mbps).