

UC Redes de Computadores

Prof. Bruno Kimura

16/12/2020

Trabalho 1: Atraso, Vazão/Gargalos, Perdas

- O trabalho poderá ser realizado em grupo de até 3 alunos.
- Verifique em anexo os comandos de interesse.
- A submissão no classroom estará aberta até **12/01/21**.
 - Entregáveis:
 - Arquivo *Core Scenario File* (.imn ou .xml).
 - Video-relatório:
 - Suba o vídeo em sua nuvem/youtube e envie somente o link do video (NÃO envie o arquivo de video pelo classroom).
- Requisitos do video-relatório:
 - Tempo esperado para o video: ~20 min.
 - Apresentar explicação do trabalho desenvolvido:
 - o que, como e porque foi feito em cada questão.
 - Descrever as contribuições de cada integrante do grupo no desenvolvimento do trabalho.
 - Apresentar uma auto-avaliação do grupo sobre o trabalho realizado.

1) Implementação da rede no emulador CORE

Em uma visão macro, segundo J. Kurose, a Internet é uma rede (complexa) composta por interconexões de redes. Tal interconexão se dá entre Redes de Acesso e Redes Núcleo. Nas Redes de Acesso estão localizados os sistemas finais que disponibilizam serviços (aplicações servidoras), os sistemas que consomem serviços (aplicações clientes), os enlaces de acesso (guiados e não-guiados) de diferentes tecnologias de transmissão. Há diversos tipos de redes de acesso, como redes institucionais, redes residências, redes de móveis de operador de telefonia, redes de sensores/IoT, redes veiculares, entre outros.

Já nas Redes de Núcleo (da Internet) estão localizados os provedores ISPs (*Internet Service Providers*). As redes de núcleo são compostas, tipicamente, por roteadores conectados tanto aos roteadores das redes de acesso quanto aos roteadores de outros ISPs. A infraestrutura dos ISPs, muitas vezes chamada de PoP (*Point of Presence*), é a infraestrutura que permite o acesso aos usuários, sendo composta, tipicamente, por equipamentos de telecomunicações (roteadores, comutadores, multiplexadores) e servidores AAA (*Authentication, Authorization, Accounting*). Os enlaces que conectam os ISPs costumam ser implementos por fibras óticas. Cabos submarinos de fibra ótica fazem conexão intercontinental de provedores.

Tanto as redes de acesso quanto as redes de núcleo são denominadas AS (*autonomous system*), “sistemas autônomos”, pois possui um domínio administrativamente independente e, principalmente, um gerenciamento e operação de rede próprio. Nesse caso, um AS possui sua topologia onde sub-redes estão conectadas por roteadores internos, sob um protocolo de roteamento intra-AS.

Utilizando o emulador CORE, defina e implemente uma topologia de rede que possa representar essa visão macro da Internet. Por exemplo, uma rede institucional onde serviços são providos através de um conjunto de nós servidores; uma rede residencial contendo um conjunto de PCs de usuários; um núcleo contendo um conjunto de redes de ISPs, de modo que um ISP_a proveja acesso à rede institucional e um IPS_b à rede residencial.

2) Sobre gargalos

2.1) Considerando a topologia de rede que você implementou, explique onde estão os possíveis gargalos na comunicação fim-a-fim entre os sistemas finais da rede residencial (clientes) e os sistemas da rede institucional (servidores).

2.2) Considerando a sua topologia, configure um enlace de gargalo considerando capacidades assimétricas, ou seja, uma taxa x para *upstream* e uma taxa y para *downstream*, tipicamente, $x < y$.

2.3) Uma vez configurado o enlace de gargalo, realize testes com a ferramenta Iperf e explique o impacto do gargalo na vazão fim-a-fim sobre os seguintes cenários:

- Um cliente fazendo download de conteúdo de um servidor remoto. Estresse a rede com dois clientes simultâneos em download.
- Um cliente fazendo upload de conteúdo para um servidor remoto. Estresse a rede com dois clientes simultâneos em upload.

3) Sobre Atrasos

3.1) Considerando a sua topologia de rede, explique quais são as possíveis fontes de atraso no caminho fim-a-fim, dos sistemas finais clientes na rede residencial aos servidores na rede institucional. Utilizando a ferramenta ping, explique os atrasos observados e como eles são determinados.

3.2) Altere o MTU (*maximum transmission unit*) do pacote na ferramenta ping e verifique o impacto na latência. Para fins de testes mais amplos, mostre as latências observadas no intervalo de 100 Bytes às 1500 Bytes. Tente identificar a relação de bytes e latência. Explique as possíveis fontes de atraso que impactam na latência. Explique porque ocorrem erros com ping sobre pacotes grandes (e.g., 1500 Bytes).

3.3) Gere carga na rede com os testes dos cenários a) e b) de transmissão da questão 2.2. Durante o download e upload, execute a ferramenta ping e verifique as latências observadas. Explique as possíveis fontes de atraso que impactam na latência.

3.4) Agora, sem carga na rede, utilizando a ferramenta tracepath ou traceroute, explique os atrasos observados e como eles são determinados.

3.5) Novamente, gere carga na rede com os testes dos cenários a) e b) de transmissão da questão 2.2. e, durante o download/upload, execute a ferramenta tracepath ou traceroute. Verifique as latências observadas e explique se é possível determinar onde está ocorrendo congestionamento na rede.

4) Sobre Perdas

4.1) Uma vez configurado o enlace de gargalo, realize testes com a ferramenta Iperf, desta vez utilizando transmissões UDP (*user datagram protocol*). Conecte um cliente da rede residencial a um servidor na rede na institucional. Configure o enlace de gargalo com uma taxa de perda $\geq 1\%$. Verifique a taxa de perda no lado do servidor e explique por que essas perdas ocorrem.

4.2) Agora, configure o enlace de gargalo com uma taxa de perda de 0%. Repita o teste da questão anterior. Verifique a taxa de perda no lado do servidor e explique porque essas perdas ainda ocorrem.

ANEXO I

Comandos de Interesse

Obs.: no nó alvo da rede no CORE, dê duplo clique e, no terminal que se abriu, aplique os comandos.

1) Teste de vazão da transmissão fim-a-fim:

Servidor TCP:

```
# iperf -s -i <intervalo>
```

Cliente TCP:

```
# iperf -c <ip_servidor> -r
```

No servidor UDP:

```
# iperf -s -u -i <intervalo>
```

No cliente TCP:

```
# iperf -c <ip_servidor> -r -u
```

Parâmetros importantes:

- c: operação do IPERF como cliente.
- r: medição de desempenho nos dois sentidos (*upstream* e *downstream*).
- t: tempo de execução do cliente em segundos.
- u: transmissão sobre UDP.
- i: intervalo em segundos para exposição das métricas de saída.

2) Teste latência e disponibilidade do nó alvo:

Ping

```
# ping <ip_maquina_destino>
```

```
# ping -M do -s <MTU> <ip_maquina_destino>
```

Parâmetros importantes:

- M: do não fragmenta o pacote.
- s: tamanho do pacote/MTU em bytes.

Traceroute

```
# traceroute <ip_maquina_destino>
```

Tracepath

```
# tracepath <ip_maquina_destino>
```