

UC Redes de Computadores

Prof. Bruno Kimura
29/10/2021

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 **13/11/21**.
 - Entregáveis:
 - Arquivo *Core Scenario File* (.imn ou .xml).
 - Video-relatório:
 - Suba o vídeo em sua nuvem/youtube e envie somente o link do vídeo (NÃO envie o arquivo de vídeo pelo classroom).
- Requisitos do video-relatório:
 - Tempo esperado para o vídeo: ~20 min.
 - Apresentar explicação do trabalho desenvolvido:
 - O que, como e porque foi feito em cada questão.
 - Todos(as) os(as) integrantes do grupos devem participar do vídeo.
 - 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 móveis de operadora de telefonia, redes de sensores/IoT, 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 de borda 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 administrativo independente e, principalmente, um gerenciamento e operação de rede próprio. Nesse caso, um AS possui sua própria rede física (topologia) e rede lógica (IP), onde as sub-redes endereçadas 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, conforme exposto acima.

2) Sobre gargalos

2.1) Considerando a topologia de rede que você implementou no emulador, 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:

- a) Um cliente fazendo download de conteúdo de um servidor remoto. Estresse a rede com outros clientes simultâneos em download sobre o mesmo enlace de gargalo e justifique a vazão obtida em cada download.
- b) Um cliente fazendo upload de conteúdo para um servidor remoto. Estresse a rede com dois clientes simultâneos em upload sobre o mesmo enlace de gargalo e justifique a vazão obtida em cada upload.

3) Sobre Atrasos

3.1) Considerando a sua topologia de rede, explique quais são as 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 fontes de atraso que estão mais impactando 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 estão impactando na latência.

3.4) Agora, sem carga na rede, utilizando a ferramenta 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 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 a um servidor passando por um gargalo. Gere carga UDP com taxa superior ao gargalo. Configure o enlace de gargalo com uma taxa de perda (loss) maior que zero. Verifique a taxa de perda no lado do servidor e explique.

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

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>
```