	<b>INSTITUTO FEDERAL DA PARAÍBA</b>	
	<b>DIRETORIA DE ENSINO</b>	
	<b>DEPARTAMENTO DE ENSINO SUPERIOR - CAMPUS JOÃO PESSOA</b>	
	<b>DISCIPLINA: PROTOCOLOS DE APLICAÇÃO</b>	
Campus João Pessoa	<b>PROFESSOR: LEONIDAS LIMA</b>	<b>PERÍODO: 2022.2</b>

## Analizando Cubic e BBR no controle de congestionamento TCP

### 1) Organizando o ambiente:

- Usando o VirtualBox, acesse a máquina virtual fornecida pelo professor.
- Abra uma sessão de trabalho para o usuário `root` (senha `ifpb`).
- Abra um terminal e instale o utilitário `iperf3`:  

```
$ apt -o Acquire::ForceIPv4=true install iperf3
```
- Verifique se o utilitário `iperf3` está funcionando:  

```
$ iperf3 -v
iperf 3.1.3
Linux debian-gui 4.9.0-8-amd64 #1 SMP Debian 4.9.144-3.1 (2019-02-19) x86_64
...
```
- Consulte o parâmetro atual do kernel referente ao TCP congestion control:  

```
$ sysctl net.ipv4.tcp_congestion_control
net.ipv4.tcp_congestion_control = cubic
```
- Consulte as informações de controle de tráfego do sistema:  

```
$ tc qdisc show dev enp0s3
qdisc pfifo_fast 0: root refcnt 2 bands 3 priomap  1 2 2 2 1 2 0 0 1 1 1 1 1 1 1 1
```

### 2) Iniciando o `iperf3` no modo servidor:

- Inicie o aplicativo `iperf3` no papel de servidor:  

```
$ iperf3 -s
-----
Server listening on 5201
-----
```

### 3) Testando o desempenho do controle de congestionamento Cubic em condições normais de tráfego:

- Abra um novo terminal.
- Consulte o endereço IP da sua máquina e informe ao seu parceiro de experimento:  

```
$ hostname -I
10.0.W.X
```
- Altere o parâmetro atual do kernel `tcp_congestion_control` para o algoritmo Cubic:  

```
$ sysctl net.ipv4.tcp_congestion_control=cubic
net.ipv4.tcp_congestion_control = cubic
```
- Trabalhando de forma alternada com seu parceiro de experimento, utilize o aplicativo `iperf3` no papel de cliente e conecte-se ao servidor `iperf3` do colega, sendo `10.0.W.Y` o endereço IP informado por este:  

```
$ iperf3 -c 10.0.W.Y
Connecting to host 10.0.W.Y, port 5201
...
- - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]  0.00-10.00  sec   2.75 GBytes  2.36 Gbits/sec  15221
[  4]  0.00-10.00  sec   2.75 GBytes  2.36 Gbits/sec
                                     sender
                                     receiver
```
- Observe o desempenho obtido, conforme indicado na coluna `Bandwidth`. Veja também qual o resultado do teste realizado pelo seu parceiro, os valores dos resultados devem ser próximos.

### 4) Testando o desempenho do controle de congestionamento BBR em condições normais de tráfego:

- Altere o parâmetro atual do kernel `tcp_congestion_control` para o algoritmo BBR:  

```
$ sysctl net.ipv4.tcp_congestion_control=bbr
net.ipv4.tcp_congestion_control = bbr
```

## Analizando Cubic e BBR no controle de congestionamento TCP

- b) Trabalhando de forma alternada com seu parceiro de experimento, utilize o aplicativo `iperf3` no papel de cliente e conecte-se ao servidor `iperf3` do colega, sendo 10.0.W.Y o endereço IP informado por este:
- ```
$ iperf3 -c 10.0.W.Y
Connecting to host 10.0.W.Y, port 5201
```

...

```

[ ID] Interval           Transfer     Bandwidth       Retr
[  4]  0.00-10.00  sec   2.78 GBytes  2.39 Gbits/sec   25973           sender
[  4]  0.00-10.00  sec   2.78 GBytes  2.39 Gbits/sec                   receiver
```

- c) Observe o desempenho obtido, conforme indicado na coluna `Bandwidth`. Veja também qual o resultado do teste realizado pelo seu parceiro, os valores dos resultados devem ser próximos. Compare os dados obtidos com o `Cubic` e com o `BBR`. Estes resultados são semelhantes?

### 5) Alterando o controle de tráfego para introduzir uma probabilidade de 1,5% na perda de pacotes:

- a) Use o aplicativo `tc` para introduzir no controle de tráfego do sistema 1,5% de probabilidade de perda:

```
$ tc qdisc replace dev enp0s3 root netem loss 1.5%
```

- b) Consulte as novas informações de controle de tráfego do sistema:

```
$ tc qdisc show dev enp0s3
qdisc netem 8001: root refcnt 2 limit 1000 loss 1.5%
```

### 6) Testando o desempenho do controle de congestionamento `Cubic` com 1,5% de perda de pacotes:

- a) Altere o parâmetro atual do kernel `tcp_congestion_control` para o algoritmo `Cubic`:

```
$ sysctl net.ipv4.tcp_congestion_control=cubic
net.ipv4.tcp_congestion_control = cubic
```

- b) Trabalhando de forma alternada com seu parceiro de experimento, utilize o aplicativo `iperf3` no papel de cliente e conecte-se ao servidor `iperf3` do colega, sendo 10.0.W.Y o endereço IP informado por este:

```
$ iperf3 -c 10.0.W.Y
Connecting to host 10.0.W.Y, port 5201
```

...

```

[ ID] Interval           Transfer     Bandwidth       Retr
[  4]  0.00-10.00  sec   1.05 GBytes  906 Mbits/sec   12827           sender
[  4]  0.00-10.00  sec   1.05 GBytes  904 Mbits/sec                   receiver
```

- c) Observe o desempenho obtido, conforme indicado na coluna `Bandwidth`. Veja também qual o resultado do teste realizado pelo seu parceiro, os valores dos resultados devem ser próximos.

### 7) Testando o desempenho do controle de congestionamento `BBR` com 1,5% de perda de pacotes:

- a) Altere o parâmetro atual do kernel `tcp_congestion_control` para o algoritmo `BBR`:

```
$ sysctl net.ipv4.tcp_congestion_control=bbr
net.ipv4.tcp_congestion_control = bbr
```

- b) Trabalhando de forma alternada com seu parceiro de experimento, utilize o aplicativo `iperf3` no papel de cliente e conecte-se ao servidor `iperf3` do colega, sendo 10.0.W.Y o endereço IP informado por este:


```
$ iperf3 -c 10.0.W.Y
Connecting to host 10.0.W.Y, port 5201
```

...

```

[ ID] Interval           Transfer     Bandwidth       Retr
[  4]  0.00-10.00  sec   2.76 GBytes  2.37 Gbits/sec   72478           sender
[  4]  0.00-10.00  sec   2.76 GBytes  2.37 Gbits/sec                   receiver
```

- c) Observe o desempenho obtido, conforme indicado na coluna `Bandwidth`. Veja também qual o resultado do teste realizado pelo seu parceiro, os valores dos resultados devem ser próximos. Compare os dados

|                                                                                                                                                   |                                                             |                        |
|---------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|------------------------|
| <br><b>INSTITUTO FEDERAL</b><br>Paraíba<br>Campus<br>João Pessoa | <b>INSTITUTO FEDERAL DA PARAÍBA</b>                         |                        |
|                                                                                                                                                   | <b>DIRETORIA DE ENSINO</b>                                  |                        |
|                                                                                                                                                   | <b>DEPARTAMENTO DE ENSINO SUPERIOR - CAMPUS JOÃO PESSOA</b> |                        |
|                                                                                                                                                   | <b>DISCIPLINA: PROTOCOLOS DE APLICAÇÃO</b>                  |                        |
|                                                                                                                                                   | <b>PROFESSOR: LEONIDAS LIMA</b>                             | <b>PERÍODO: 2022.2</b> |

### Analizando Cubic e BBR no controle de congestionamento TCP

obtidos com o Cubic e com o BBR. Estes resultados são semelhantes? Explique!

8) Retornando aos parâmetros originais do controle de tráfego do sistema e controle de congestionamento TCP:

a) Use o aplicativo `tc` para remover alterações no controle de tráfego do sistema:

```
$ tc qdisc delete dev enp0s3 root netem
```

b) Consulte as novas informações de controle de tráfego do sistema:

```
$ tc qdisc show dev enp0s3
qdisc pfifo_fast 0: root refcnt 2 bands 3 priomap  1 2 2 2 1 2 0 0 1 1 1 1 1 1 1 1
```

c) Altere o parâmetro atual do kernel `tcp_congestion_control` para o algoritmo Cubic:

```
$ sysctl net.ipv4.tcp_congestion_control=cubic
net.ipv4.tcp_congestion_control = cubic
```

9) Encerrando o sistema e desligando a máquina:

a) Acesse o terminal da máquina virtual e desligue a mesma:

```
$ poweroff
```