

به نام خدا



درس آزمایشگاه شبکه‌های کامپیوتری

تمرین پنجم

مدرس درس:
سرکار خانم دکتر رشیدی

تهیه کنندگان:
حوریه سبزواری، الناز رضایی

تاریخ ارسال: ۱۴۰۲/۰۲/۱۳

الف) روال آزمایش

ابتدا برنامه‌های موجود در فایل lab5 را با دستور `chmod +x tcp/tcpclient tcp/tcpserver` و `udp/udpclient udp/udpserver` به حالت اجرایی در می‌آوریم.

```
mininet@TCPIP-VM:~/Desktop/shared/HW5/lab5$ chmod +x tcp/tcpclient tcp/tcpserver
udp/udpclient udp/udpserver
```

حال با استفاده از دستور `cat /proc/sys/net/ipv4/tcp_congestion_control`، مکانیزم کنترل ازدحام را بررسی کرده و می‌بینیم در ترمینال reno چاپ می‌شود.

```
mininet@TCPIP-VM:~/Desktop/shared/HW5/lab5$ cat /proc/sys/net/ipv4/tcp_congestion_control
reno
```

چون در اینجا مکانیزم مورد استفاده به صورت پیش‌فرض reno بود، نیازی به استفاده از دستورات دیگر نداریم.

ب) جریان‌های TCP و UDP

ابتدا کد داده شده را مطابق اطلاعات داده شده، به صورت زیر تکمیل می‌کنیم.

```
info("\n** Adding hosts\n")
r1 = net.addhost('r1', ip='10.10.0.10/24', hostname='r1', privateLogDir=True, privateRunDir=True, inMountNamespace=True, inPIDNamespace=True, inUTSNamespace=True)
# Space to add any commands for configuring the IP addresses
h1 = net.addhost('h1', ip='10.10.0.1/24', hostname='h1', privateLogDir=True, privateRunDir=True, inMountNamespace=True, inPIDNamespace=True, inUTSNamespace=True)
h2 = net.addhost('h2', ip='10.10.0.2/24', hostname='h2', privateLogDir=True, privateRunDir=True, inMountNamespace=True, inPIDNamespace=True, inUTSNamespace=True)
h3 = net.addhost('h3', ip='10.10.1.3/24', hostname='h3', privateLogDir=True, privateRunDir=True, inMountNamespace=True, inPIDNamespace=True, inUTSNamespace=True)
```

```
info('*** Configuring hosts\n')
r1.cmd('ifconfig r1-eth1 10.10.1.10 netmask 255.255.255.0')
# Space to add commands for configuring routing tables and default gateways
r1.cmd('ifconfig r1-eth0 10.10.0.10 netmask 255.255.255.0')
r1.cmd('echo 1 > /proc/sys/net/ipv4/ip_forward')
h1.cmd('ip route add default via 10.10.0.10')
h2.cmd('ip route add default via 10.10.0.10')
h3.cmd('ip route add default via 10.10.1.10')
```

حال با استفاده از دستور `python lab5.py` فایل اصلاح شده را ران کرده و با استفاده از دستور `pingall` اتصالات را چک می‌کنیم.

```
mininet> pingall
*** Ping: testing ping reachability
r1 -> h1 h2 h3
h1 -> r1 h2 h3
h2 -> r1 h1 h3
h3 -> r1 h1 h2
*** Results: 0% dropped (12/12 received)
```

ب-۱) تست اتصال بین کلاینت و سرورهای TCP و UDP

با استفاده از دستور 10000 /udpserver ، یک سرور با پورت ۱۰۰۰۰ بر روی h3 راه اندازی می‌کنیم.

```
0.0s - received: 1/ sent: 1 pkts (loss 0.000%), 0.2 kbit/s
43.0s - received: 14/ sent: 14 pkts (loss 0.000%), 100.0 kbit/s
44.0s - received: 27/ sent: 27 pkts (loss 0.000%), 100.0 kbit/s
45.1s - received: 40/ sent: 40 pkts (loss 0.000%), 100.0 kbit/s
46.1s - received: 53/ sent: 53 pkts (loss 0.000%), 100.0 kbit/s
47.1s - received: 66/ sent: 66 pkts (loss 0.000%), 99.8 kbit/s
48.2s - received: 79/ sent: 79 pkts (loss 0.000%), 100.3 kbit/s
49.2s - received: 92/ sent: 92 pkts (loss 0.000%), 100.0 kbit/s
50.3s - received: 105/ sent: 105 pkts (loss 0.000%), 100.1 kbit/s
51.3s - received: 118/ sent: 118 pkts (loss 0.000%), 100.0 kbit/s
52.3s - received: 131/ sent: 131 pkts (loss 0.000%), 100.0 kbit/s
53.4s - received: 144/ sent: 144 pkts (loss 0.000%), 100.0 kbit/s
54.4s - received: 157/ sent: 157 pkts (loss 0.000%), 100.0 kbit/s
55.5s - received: 170/ sent: 170 pkts (loss 0.000%), 100.0 kbit/s
56.5s - received: 183/ sent: 183 pkts (loss 0.000%), 100.0 kbit/s
57.5s - received: 196/ sent: 196 pkts (loss 0.000%), 100.0 kbit/s
packet received = 203 / 203 sent: 0.000% loss
58.6s - received: 1/ sent: 1 pkts (loss 0.000%), 18.3 kbit/s
62.1s - received: 14/ sent: 14 pkts (loss 0.000%), 100.4 kbit/s
63.1s - received: 27/ sent: 27 pkts (loss 0.000%), 100.0 kbit/s
64.1s - received: 40/ sent: 40 pkts (loss 0.000%), 99.8 kbit/s
65.2s - received: 53/ sent: 53 pkts (loss 0.000%), 100.2 kbit/s
66.2s - received: 66/ sent: 66 pkts (loss 0.000%), 100.0 kbit/s
packet received = 70 / 70 sent: 0.000% loss
67.3s - received: 1/ sent: 1 pkts (loss 0.000%), 25.9 kbit/s
68.8s - received: 14/ sent: 14 pkts (loss 0.000%), 100.1 kbit/s
69.9s - received: 27/ sent: 27 pkts (loss 0.000%), 100.0 kbit/s
70.9s - received: 40/ sent: 40 pkts (loss 0.000%), 99.9 kbit/s
71.9s - received: 53/ sent: 53 pkts (loss 0.000%), 100.1 kbit/s
73.0s - received: 66/ sent: 66 pkts (loss 0.000%), 99.9 kbit/s
74.0s - received: 79/ sent: 79 pkts (loss 0.000%), 100.0 kbit/s
packet received = 82 / 82 sent: 0.000% loss
```

پاسخ ۱: همانطور که در شکل بالا نیز مشاهده می‌شود احتمال loss برابر با صفر و مقدار goodput تقریباً 100 kbit/s می‌باشد.