

Отчет по лабораторной работе №5 Дисциплина: Дисциплина: Моделирование сетей передачи данных

Леснухин Даниил Дмитриевич
Российский университет дружбы народов
Москва

Цель работы

Основной целью работы является получение навыков проведения интерактивных экспериментов в среде Mininet по исследованию параметров сети, связанных с потерей, дублированием, изменением порядка и повреждением пакетов при передаче данных. Эти параметры влияют на производительность протоколов и сетей.

Задание

1. Задайте простейшую топологию, состоящую из двух хостов и коммутатора с назначенной по умолчанию mininet сетью 10.0.0.0/8.
2. Проведите интерактивные эксперименты по исследованию параметров сети, связанных с потерей, дублированием, изменением порядка и повреждением пакетов при передаче данных.
3. Реализуйте воспроизводимый эксперимент по добавлению правила отбрасывания пакетов в эмулируемой глобальной сети. На экран выведите сводную информацию о потерянных пакетах.
4. Самостоятельно реализуйте воспроизводимые эксперименты по исследованию параметров сети, связанных с потерей, изменением порядка и повреждением пакетов при передаче данных. На экран выведите сводную информацию о потерянных пакетах.

Теоретическое введение

Mininet[@mininet] – это эмулятор компьютерной сети. Под компьютерной сетью подразумеваются простые компьютеры — хосты, коммутаторы, а так же OpenFlow-контроллеры. С помощью простейшего синтаксиса в примитивном интерпретаторе команд можно разворачивать сети из произвольного количества хостов, коммутаторов в различных топологиях и все это в рамках одной виртуальной машины(ВМ). На всех хостах можно изменять сетевую конфигурацию, пользоваться стандартными утилитами(ifconfig, ping) и даже получать доступ к терминалу. На коммутаторы можно добавлять различные правила и маршрутизировать трафик.

Выполнение лабораторной работы

Запуск лабораторной топологии

Запустим виртуальную среду с mininet. Из основной ОС подключимся к виртуальной машине. В виртуальной машине mininet при необходимости исправим права запуска X-соединения. Скопируем значение куки (MIT magic cookie) своего пользователя mininet в файл для пользователя root.

Зададим простейшую топологию, состоящую из двух хостов и коммутатора с назначенной по умолчанию mininet сетью 10.0.0.0/8. На хостах h1 и h2 введем команду ifconfig, чтобы отобразить информацию, относящуюся к их сетевым интерфейсам и назначенным им IP-адресам. В дальнейшем при работе с NETEM и командой tc будут использоваться интерфейсы h1-eth0 и h2-eth0

Проверим подключение между хостами h1 и h2 с помощью команды ping с параметром -c 6

```
"host: h2"@mininet-vm
root@mininet-vm:/home/mininet# ifconfig
h2-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.2 netmask 255.0.0.0 broadcast 10.255.255.255
    ether 12:c9:66:3e:7b:2a txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 1185 bytes 256388 (256.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1185 bytes 256388 (256.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet#

"host: h1"@mininet-vm
root@mininet-vm:/home/mininet# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.1 netmask 255.0.0.0 broadcast 10.255.255.255
    ether da:bf:dc:62:00:88 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 1138 bytes 250124 (250.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1138 bytes 250124 (250.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet#
```

Рис. 1: ifconfig

```
TX packets 0  bytes 0 (0.0 B)
TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    loop txqueuelen 1000  (Local Loopback)
    RX packets 1185  bytes 256388 (256.3 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 1185  bytes 256388 (256.3 KB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
root@mininet-vm:/home/mininet# ping 10.0.0.1 -c 6
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=1.08 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.060 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.049 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.070 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.047 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.043 ms
```

```
--- 10.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5079ms
rtt min/avg/max/mdev = 0.043/0.225/1.083/0.383 ms
root@mininet-vm:/home/mininet#
```

```
    inet 127.0.0.1  netmask 255.0.0.0
    loop txqueuelen 1000  (Local Loopback)
    RX packets 1138  bytes 250124 (250.1 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 1138  bytes 250124 (250.1 KB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
root@mininet-vm:/home/mininet# ping 10.0.0.2 -c 6
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=2.00 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.210 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.045 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.057 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.065 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.099 ms
```

```
--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5094ms
rtt min/avg/max/mdev = 0.045/0.413/2.004/0.713 ms
root@mininet-vm:/home/mininet#
```

Интерактивные эксперименты

Добавление потери пакетов на интерфейс, подключённый к эмулируемой глобальной сети

Пакеты могут быть потеряны в процессе передачи из-за таких факторов, как битовые ошибки и перегрузка сети. Скорость потери данных часто измеряется как процентная доля потерянных пакетов по отношению к количеству отправленных пакетов.

1. На хосте h1 добавим 10% потерь пакетов к интерфейсу h1-eth0: `sudo tc qdisc add dev`

```
h1-eth0 root netem loss 10%
```

```
host: h1@mininet-vm
64 bytes from 10.0.0.2: icmp_seq=78 ttl=64 time=0.159 ms
64 bytes from 10.0.0.2: icmp_seq=79 ttl=64 time=0.070 ms
64 bytes from 10.0.0.2: icmp_seq=80 ttl=64 time=0.158 ms
64 bytes from 10.0.0.2: icmp_seq=83 ttl=64 time=0.167 ms
64 bytes from 10.0.0.2: icmp_seq=84 ttl=64 time=0.094 ms
64 bytes from 10.0.0.2: icmp_seq=85 ttl=64 time=0.197 ms
64 bytes from 10.0.0.2: icmp_seq=86 ttl=64 time=0.069 ms
64 bytes from 10.0.0.2: icmp_seq=87 ttl=64 time=0.053 ms
64 bytes from 10.0.0.2: icmp_seq=88 ttl=64 time=0.073 ms
64 bytes from 10.0.0.2: icmp_seq=89 ttl=64 time=0.074 ms
64 bytes from 10.0.0.2: icmp_seq=90 ttl=64 time=0.057 ms
64 bytes from 10.0.0.2: icmp_seq=91 ttl=64 time=0.078 ms
64 bytes from 10.0.0.2: icmp_seq=93 ttl=64 time=0.153 ms
64 bytes from 10.0.0.2: icmp_seq=94 ttl=64 time=0.062 ms
64 bytes from 10.0.0.2: icmp_seq=95 ttl=64 time=0.112 ms
64 bytes from 10.0.0.2: icmp_seq=96 ttl=64 time=0.116 ms
64 bytes from 10.0.0.2: icmp_seq=97 ttl=64 time=0.067 ms
64 bytes from 10.0.0.2: icmp_seq=99 ttl=64 time=0.076 ms
64 bytes from 10.0.0.2: icmp_seq=100 ttl=64 time=0.091 ms

--- 10.0.0.2 ping statistics ---
100 packets transmitted, 88 received, 12% packet loss, time 101340ms
rtt min/avg/max/mdev = 0.042/0.098/1.692/0.177 ms
root@mininet-vm:/home/mininet#
```

Рис. 2: Добавим 10% потерь пакетов к интерфейсу h1-eth0

Здесь: - sudo: выполнить команду с более высокими привилегиями; - tc: вызвать управление трафиком Linux; - qdisc: изменить дисциплину очередей сетевого планировщика; - add: создать новое правило; - dev h1-eth0: указать интерфейс, на котором будет применяться правило; - netem: использовать эмулятор сети; - loss 10%: 10% потерь пакетов.

2. Проверьте, что на соединении от хоста h1 к хосту h2 имеются потери пакетов, используя команду ping с параметром -c 100 с хоста h1. Параметр -c указывает общее количество пакетов для отправки. Обратите внимание на значения icmp_seq. Некоторые номера последовательности отсутствуют из-за потери пакетов. В сводном отчёте ping сообщает о проценте потерянных пакетов после завершения передачи
3. Для эмуляции глобальной сети с потерей пакетов в обоих направлениях необходимо к соответствующему интерфейсу на хосте h2 также добавить 10% потерь пакетов: `sudo tc qdisc add dev h2-eth0 root netem loss 10%`.

Восстановите конфигурацию по умолчанию, удалив все правила, применённые к сетевому планировщику соответствующего интерфейса. Для отправителя h1: `sudo tc qdisc del dev h1-eth0 root netem`. Для получателя h2: `sudo tc qdisc del dev h2-eth0 root netem`

```

root@mininet-vm:/home/mininet# sudo tc qdisc add dev h2-eth0 root netem loss 10
%
root@mininet-vm:/home/mininet# █
64 bytes from 10.0.0.2: icmp_seq=80 ttl=64 time=0.045 ms
64 bytes from 10.0.0.2: icmp_seq=82 ttl=64 time=0.064 ms
64 bytes from 10.0.0.2: icmp_seq=83 ttl=64 time=0.076 ms
64 bytes from 10.0.0.2: icmp_seq=84 ttl=64 time=0.052 ms
64 bytes from 10.0.0.2: icmp_seq=85 ttl=64 time=0.067 ms
64 bytes from 10.0.0.2: icmp_seq=88 ttl=64 time=0.054 ms
64 bytes from 10.0.0.2: icmp_seq=89 ttl=64 time=0.109 ms
64 bytes from 10.0.0.2: icmp_seq=90 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=91 ttl=64 time=0.163 ms
64 bytes from 10.0.0.2: icmp_seq=92 ttl=64 time=0.065 ms
64 bytes from 10.0.0.2: icmp_seq=94 ttl=64 time=0.133 ms
64 bytes from 10.0.0.2: icmp_seq=95 ttl=64 time=0.131 ms
64 bytes from 10.0.0.2: icmp_seq=98 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=99 ttl=64 time=0.058 ms
64 bytes from 10.0.0.2: icmp_seq=100 ttl=64 time=0.106 ms

--- 10.0.0.2 ping statistics ---
100 packets transmitted, 80 received, 20% packet loss, time 101355ms
rtt min/avg/max/mdev = 0.041/0.080/0.373/0.046 ms
root@mininet-vm:/home/mininet# █

```

Рис. 3: Добавим 10% потерь пакетов к интерфейсу h2-eth0

Добавление дублирования пакетов в интерфейс подключения к эмулируемой глобальной сети

1. При необходимости восстановите конфигурацию интерфейсов по умолчанию на узлах h1 и h2.
2. Для интерфейса узла h1 задайте правило с дублированием 50% пакетов (т.е. 50% пакетов должны быть получены дважды): `sudo tc qdisc add dev h1-eth0 root netem duplicate 50%`

Добавление потери пакетов на интерфейс, подключённый к эмулируемой глобальной сети

С помощью API Mininet воспроизведите эксперимент по добавлению потери пакетов для интерфейса хоста, подключающегося к эмулируемой глобальной сети.

1. В виртуальной среде mininet в своём рабочем каталоге с проектами создайте каталог `simple-drop` и перейдите в него: `mkdir -p ~/work/lab_netem_ii/simple-drop, cd ~/work/lab_netem_ii/simple-drop`.
2. Создаёте скрипт для эксперимента `lab_netem_ii.py`:

```
"host: h2"@mininet-vm
Accepted connection from 10.0.0.1, port 60282
[ 7] local 10.0.0.2 port 5201 connected to 10.0.0.1 port 60284
[ ID] Interval          Transfer      Bitrate
[ 7]  0.00-1.00      sec  4.44 GBytes  38.2 Gbits/sec
[ 7]  1.00-2.00      sec  4.00 GBytes  34.4 Gbits/sec
[ 7]  2.00-3.00      sec  3.43 GBytes  29.4 Gbits/sec
[ 7]  3.00-4.00      sec  3.94 GBytes  33.8 Gbits/sec
[ 7]  4.00-5.00      sec  4.40 GBytes  37.8 Gbits/sec
[ 7]  5.00-6.00      sec  2.42 GBytes  20.8 Gbits/sec
[ 7]  6.00-7.00      sec  2.11 GBytes  18.2 Gbits/sec
[ 7]  7.00-8.00      sec  2.69 GBytes  23.1 Gbits/sec
[ 7]  8.00-9.00      sec  2.71 GBytes  23.3 Gbits/sec
[ 7]  9.00-10.00     sec  2.27 GBytes  19.5 Gbits/sec
[ 7] 10.00-10.00     sec  6.12 MBytes  12.0 Gbits/sec
-----
[ ID] Interval          Transfer      Bitrate
[ 7]  0.00-10.00     sec  32.4 GBytes  27.8 Gbits/sec
-----
Server listening on 5201
-----
^Ciperf3: interrupt - the server has terminated
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h2-eth0 root netem
Error: Invalid qdisc name.
root@mininet-vm:/home/mininet#
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.055 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.070 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.070 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.054 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.054 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.066 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.066 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=0.051 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=0.051 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=0.055 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=0.061 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=0.061 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=0.048 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=0.051 ms

--- 10.0.0.2 ping statistics ---
20 packets transmitted, 20 received, +12 duplicates, 0% packet loss, time 19410ms
rtt min/avg/max/mdev = 0.045/0.138/2.009/0.340 ms
root@mininet-vm:/home/mininet#
```

Рис. 4: Добавим на интерфейсе узла h1 правило с дублированием 50% пакетов. Проверка потери пакетов

```
GNU nano 4.8 lab netem ii.py
h2.cmdPrint( 'tc qdisc add dev h2-eth0>

time.sleep(10) # Wait 10 seconds

info( '*** Ping\n')
h1.cmdPrint( 'ping -c 100', h2.IP(), '>

info( '*** Stopping network' )
net.stop()

if __name__ == '__main__':
    setLogLevel( 'info' )
    emptyNet()
```

Рис. 5: Скрипт жддля эксперимента

```
mininet@mininet-vm: ~/work/lab_netem_ii/simple-drop
GNU nano 4.8
all: ping.dat

ping.dat:
    sudo python lab_netem_ii
    sudo chown mininet:mininet

clean:
    -rm -f *.dat
```

Создайте Makefile для управления процессом проведения эксперимента

Выполните эксперимент: make

6. Очистите каталог от результатов проведения экспериментов: make clean.

Выводы

В результате выполнения данной лабораторной работы я получил навыки проведения интерактивных экспериментов в среде Mininet по исследованию параметров сети, связанных с потерей, дублированием, изменением порядка и повреждением пакетов при передаче данных.


```

mininet@mininet-vm:~/work/lab_netem_ii/simple-drop$ make
sudo python lab_netem_ii.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
*** Starting network
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
s1
*** Set delay
*** h1 : ('tc qdisc add dev h1-eth0 root netem loss 10%',)
*** h2 : ('tc qdisc add dev h2-eth0 root netem loss 10%',)
*** Ping
*** h1 : ('ping -c 100', '10.0.0.2', '| grep "time=" | awk \'{print $5, $7}\'' | s
ed -e \'/s/time=//g\' -e \'/s/icmp_seq=//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2

```

Рис. 6: Выполнение эксперимента. Просмотр информации о потерях пакетов

```

mininet@mininet-vm:~/work/lab_netem_ii/simple-drop$ make clean
rm -f *.dat
mininet@mininet-vm:~/work/lab_netem_ii/simple-drop$ cat ping.dat
cat: ping.dat: No such file or directory
mininet@mininet-vm:~/work/lab_netem_ii/simple-drop$

```

Рис. 7: makeclean