Compouter Network HW1

Yiping Deng

February 21, 2019

Contents

1 Problem 1.1

1.1 a)

```
~/hw/cn/hw1 > fping -version
```

```
fping: Version 4.1
fping: comments to david@schweikert.ch
~/hw/cn/hw1 > fping amazon.com www.amazon.com www.jacobs-university.de moodle.jacobs-u
www.amazon.com                : [0], 84 bytes, 17.7 ms (17.7 avg, 0% loss)
www.jacobs-university.de      : [0], 84 bytes, 18.4 ms (18.4 avg, 0% loss)
moodle.jacobs-university.de : [0], 84 bytes, 8.58 ms (8.58 avg, 0% loss)
amazon.com                    : [0], 84 bytes, 108 ms (108 avg, 0% loss)
moodle.jacobs-university.de : [1], 84 bytes, 3.21 ms (5.89 avg, 0% loss)
www.jacobs-university.de      : [1], 84 bytes, 16.1 ms (17.2 avg, 0% loss)
www.amazon.com                : [1], 84 bytes, 31.7 ms (24.7 avg, 0% loss)
amazon.com                    : [1], 84 bytes, 132 ms (120 avg, 0% loss)
www.amazon.com                : [2], 84 bytes, 18.7 ms (22.7 avg, 0% loss)
moodle.jacobs-university.de : [2], 84 bytes, 3.57 ms (5.12 avg, 0% loss)
www.jacobs-university.de      : [2], 84 bytes, 22.6 ms (19.0 avg, 0% loss)
amazon.com                    : [2], 84 bytes, 108 ms (116 avg, 0% loss)
www.amazon.com                : [3], 84 bytes, 17.4 ms (21.4 avg, 0% loss)
www.jacobs-university.de      : [3], 84 bytes, 16.4 ms (18.4 avg, 0% loss)
moodle.jacobs-university.de : [3], 84 bytes, 5.04 ms (5.10 avg, 0% loss)
amazon.com                    : [3], 84 bytes, 107 ms (114 avg, 0% loss)
www.amazon.com                : [4], 84 bytes, 28.5 ms (22.8 avg, 0% loss)
www.jacobs-university.de      : [4], 84 bytes, 16.1 ms (17.9 avg, 0% loss)
moodle.jacobs-university.de : [4], 84 bytes, 4.27 ms (4.93 avg, 0% loss)
amazon.com                    : [4], 84 bytes, 119 ms (115 avg, 0% loss)
www.amazon.com                : [5], 84 bytes, 17.3 ms (21.9 avg, 0% loss)
www.jacobs-university.de      : [5], 84 bytes, 16.5 ms (17.7 avg, 0% loss)
moodle.jacobs-university.de : [5], 84 bytes, 8.18 ms (5.47 avg, 0% loss)
amazon.com                    : [5], 84 bytes, 110 ms (114 avg, 0% loss)
www.amazon.com                : [6], 84 bytes, 17.7 ms (21.3 avg, 0% loss)
moodle.jacobs-university.de : [6], 84 bytes, 1.68 ms (4.93 avg, 0% loss)
www.jacobs-university.de      : [6], 84 bytes, 18.2 ms (17.7 avg, 0% loss)
amazon.com                    : [6], 84 bytes, 109 ms (113 avg, 0% loss)
www.amazon.com                : [7], 84 bytes, 172 ms (40.2 avg, 0% loss)
amazon.com                    : [7], 84 bytes, 184 ms (122 avg, 0% loss)
moodle.jacobs-university.de : [7], 84 bytes, 153 ms (23.5 avg, 0% loss)
www.jacobs-university.de      : [7], 84 bytes, 173 ms (37.2 avg, 0% loss)
www.jacobs-university.de      : [8], 84 bytes, 14.8 ms (34.7 avg, 0% loss)
www.amazon.com                : [8], 84 bytes, 26.7 ms (38.7 avg, 0% loss)
moodle.jacobs-university.de : [8], 84 bytes, 2.20 ms (21.1 avg, 0% loss)
amazon.com                    : [8], 84 bytes, 202 ms (131 avg, 0% loss)
moodle.jacobs-university.de : [9], 84 bytes, 7.14 ms (19.7 avg, 0% loss)
```

```
www.jacobs-university.de    : [9], 84 bytes, 33.2 ms (34.5 avg, 0% loss)
www.amazon.com              : [9], 84 bytes, 48.6 ms (39.7 avg, 0% loss)
amazon.com                  : [9], 84 bytes, 226 ms (141 avg, 0% loss)

amazon.com                  : 108.58 132.96 108.60 107.63 119.61 110.51 109.99 184.01 2
www.amazon.com              : 17.75 31.74 18.73 17.42 28.53 17.36 17.75 172.75 26.70 48
www.jacobs-university.de    : 18.44 16.12 22.66 16.42 16.14 16.52 18.26 173.24 14.88 33
moodle.jacobs-university.de : 8.58 3.21 3.57 5.04 4.27 8.18 1.68 153.68 2.20 7.14
```

Measure time: Feb 21, 14:21PM CET Measure tool: fping version 4.1 Observation: moodle is the fastest, and amazon.com is the slowest. Despite the fact that amazon.com and www.amazon.com are all owned by amazon, www.amazon.com is significantly faster due to cdn

Table 1: Minimu round-trip time

amazon.com	www.amazon.com	www.jacobs-university.de	moodle.jacobs-university.de
108 ms	17.3 ms	14.8 ms	1.68 ms

1.2 b)

For amazon.com

```
 ~ > sudo mtr -z amazon.com --report --report-cycles 10 -b
Start: 2019-02-21T15:27:26+0100
HOST: Yipings-MBP.wlan.jacobs-uni Loss%   Snt   Last   Avg  Best  Wrst StDev
  1. AS???    10.81.255.251        0.0%    10   81.0  18.0   1.4  81.0  27.6
  2. AS???    192.168.242.3        0.0%    10   36.2  18.8   1.4 118.0  36.5
  3. AS680    vkr-g2-5-1.x-win.un  0.0%    10    3.1  10.0   1.9  72.5  22.0
  4. AS680    cr-han2-be15.x-win.  0.0%    10  151.8  24.0   4.2 151.8  45.4
  5. AS680    cr-fra2-be12.x-win.  0.0%    10  106.9  49.1   9.3 161.2  60.7
  6. AS1299   ffm-b12-link.telia.  0.0%    10   65.7  37.8  10.9 116.8  39.2
  7. AS1299   ffm-bb3-link.telia.  0.0%    10  159.2 142.5 108.6 278.4  57.3
  8. AS1299   prs-bb3-link.telia.  0.0%    10  290.3 147.7 109.5 290.3  63.8
  9. AS1299   ash-bb3-link.telia.  0.0%    10  244.5 144.9 101.8 291.0  70.6
 10. AS1299   ash-b1-link.telia.n  0.0%    10  200.9 149.4 101.6 298.8  71.8
 11. AS1299   vadata-ic-333120-as  0.0%    10  152.5 145.3 103.4 255.1  51.7
 12. AS???    ???                 100.0    10    0.0   0.0   0.0   0.0   0.0
 13. AS???    ???                 100.0    10    0.0   0.0   0.0   0.0   0.0
 14. AS???    ???                 100.0    10    0.0   0.0   0.0   0.0   0.0
 15. AS???    ???                 100.0    10    0.0   0.0   0.0   0.0   0.0
 16. AS???    ???                 100.0    10    0.0   0.0   0.0   0.0   0.0
 17. AS???    ???                 100.0    10    0.0   0.0   0.0   0.0   0.0
 18. AS???    ???                 100.0    10    0.0   0.0   0.0   0.0   0.0
 19. AS???    ???                 100.0    10    0.0   0.0   0.0   0.0   0.0
 20. AS???    ???                 100.0    10    0.0   0.0   0.0   0.0   0.0
 21. AS???    ???                 100.0    10    0.0   0.0   0.0   0.0   0.0
 22. AS???    ???                 100.0    10    0.0   0.0   0.0   0.0   0.0
 23. AS16509  176.32.103.205       0.0%    10  263.5 191.8 112.1 375.7  98.2
```

AS-path: AS???(1 hop) -> AS680(2 hop) -> AS1299(5 hop) -> AS???(10 hop) -> AS16509(0 hop)

For www.amazon.com

```
~ > sudo mtr -z www.amazon.com --report --report-cycles 10 -b
Start: 2019-02-21T15:29:24+0100
HOST: Yipings-MBP.wlan.jacobs-uni Loss%   Snt   Last   Avg  Best  Wrst StDev
  1. AS???    10.81.255.251        0.0%    10    1.4   5.3   1.4  23.9   7.4
  2. AS???    192.168.242.3        0.0%    10    1.5   4.1   1.1  17.6   4.9
  3. AS680    vkr-g2-5-1.x-win.un  0.0%    10    6.1   9.8   1.9  53.7  15.6
  4. AS680    cr-han2-be15.x-win. 10.0%    10    6.9  15.3   4.1  59.9  21.0
  5. AS680    cr-tub2-be9.x-win.d 10.0%    10   10.9  33.1   8.5 138.4  43.9
  6. AS???    amazon.bcix.de (193  0.0%    10   10.8  44.5   8.7 298.2  90.8
  7. AS???    ???                 100.0    10    0.0   0.0   0.0   0.0   0.0
  8. AS???    ???                 100.0    10    0.0   0.0   0.0   0.0   0.0
  9. AS???    ???                 100.0    10    0.0   0.0   0.0   0.0   0.0
 10. AS???    ???                 100.0    10    0.0   0.0   0.0   0.0   0.0
 11. AS???    ???                 100.0    10    0.0   0.0   0.0   0.0   0.0
 12. AS???    ???                 100.0    10    0.0   0.0   0.0   0.0   0.0
 13. AS???    ???                 100.0    10    0.0   0.0   0.0   0.0   0.0
 14. AS16509  server-99-84-149-24  0.0%    10   13.7  11.5   8.6  24.8   4.9
```

AS-path: AS???(1 hop) -> AS680(2 hop) -> AS???(7 hop) -> AS16509(0 hop)

For www.jacobs-university.de

```
 ~ > sudo mtr -z www.jacobs-university.de --report --report-cycles 10 -b
Password:
Start: 2019-02-21T15:36:46+0100
HOST: Yipings-MBP.wlan.jacobs-uni Loss%   Snt   Last   Avg  Best  Wrst StDev
```

```
  1. AS???    10.81.255.251        0.0%    10    1.3   3.3   1.2  14.2   3.9
  2. AS???    192.168.242.3        0.0%    10    6.5   5.2   1.2  20.8   5.9
  3. AS680    vkr-g2-5-1.x-win.un  0.0%    10    2.1   4.8   1.7  21.4   6.0
  4. AS680    cr-han2-be15.x-win.  0.0%    10    5.6  10.0   4.1  31.9   8.5
  5. AS680    cr-fra2-be12.x-win.  0.0%    10    9.3  15.9   9.0  57.6  14.8
  6. AS???    decix-gw.hetzner.de  0.0%    10   19.3  31.9  15.7  85.3  24.0
  7. AS24940  core24.fsn1.hetzner  0.0%    10   20.0  23.7  15.6  61.0  13.8
  8. AS24940  ex9k1.dc11.fsn1.het  0.0%    10   63.1  26.6  21.7  63.1  12.9
  9. AS24940  static.204.219.251.  0.0%    10   13.9  15.6  13.4  20.2   2.9
```

AS-path: AS???(1 hop)-> AS680(2 hop) -> AS???(1 hop) -> AS24840(2 hop)

For moodle.jacobs-university.de

```
 ~ > sudo mtr -z moodle.jacobs-university.de --report --report-cycles 10 -b
Start: 2019-02-21T15:38:11+0100
HOST: Yipings-MBP.wlan.jacobs-uni Loss%   Snt   Last   Avg  Best  Wrst StDev
  1. AS???    10.81.255.251        0.0%    10    5.8   6.2   1.4  25.8   7.2
  2. AS680    moodle.jacobs-unive  0.0%    10    7.6  10.7   1.5  43.3  12.5
```

AS-path: AS???(0 hop) -> AS680(0 hop)

Measure tool: mtr, version 0.92 Measure time: Feb 21, 3:20PM CET

Observation:

- All the outbound traffic path through AS680
- 10.81.255.251 is always the first destination
- amazon.com and www.amazon.com take completely different route after AS680
- moodle is hosted on campus, thus the route is short
- www.jacobs-university.de takes significantly longer route since it is not hosted locally

2 Problem 1.2

2.1 a)

Below attached a table for the solution

Table 2: AS registers

AS numbers	register
680	RIPE
16509	ARIN
1299	RIPE
24940	RIPE

2.2 b)

2001:638:709::/48. is not globally announced, but 2001:638::/32 is globally announced by AS680. In the whois record, "Campus Network of the International University Bremen" is using it. IUB-NET is its name.

3 Problem 1.3

On a ubuntu 18.04.02 machine, to install mininet and the necessary dependencies, simply execute

```
apt install mininet
sudo apt-get install openvswitch-testcontroller
```

Also, in the process of running the script, I encountered several situation that the port has been taken. Hence, I use the fuser command to kill the process occupying the port.

3.1 a)

After running the script, we enter the mininet command line interface. First thing we do is to figure out the corrent ip address of the two host

```
h1 ifconfig
```

will give you the following

```
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
        inet6 fe80::200:ff:fe00:1  prefixlen 64  scopeid 0x20<link>
        ether 00:00:00:00:00:01  txqueuelen 1000  (Ethernet)
        RX packets 11  bytes 866 (866.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 13  bytes 1086 (1.0 KB)
        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
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        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
```

Do the same for host 2

```
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
        inet6 fe80::200:ff:fe00:2  prefixlen 64  scopeid 0x20<link>
        ether 00:00:00:00:00:02  txqueuelen 1000  (Ethernet)
        RX packets 14  bytes 1156 (1.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12  bytes 936 (936.0 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
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        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
```

We also test the connectivity of the two host using ping, and it seems correct.

Now we start to execute the command

```
h2 iperf -s &
```

this will start the server to test the transfer rate. Then we use a client.

```
h1 iperf -c h2 -i 10 -t 60
```

The following is the test result.

```
------------------------------------------------------------
Client connecting to 10.0.0.2, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.1 port 41780 connected with 10.0.0.2 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec  11.9 MBytes  9.96 Mbits/sec
[  3] 10.0-20.0 sec  11.4 MBytes  9.54 Mbits/sec
[  3] 20.0-30.0 sec  11.4 MBytes  9.54 Mbits/sec
[  3] 30.0-40.0 sec  11.4 MBytes  9.54 Mbits/sec
[  3] 40.0-50.0 sec  11.5 MBytes  9.65 Mbits/sec
[  3] 50.0-60.0 sec  11.4 MBytes  9.54 Mbits/sec
[  3]  0.0-60.1 sec  68.9 MBytes  9.61 Mbits/sec
```

Apparently, the data are transfered at a rate of 10Mbits/s. However, when we execute the command to check the transfer rate at interface level, we have

```
mininet> h1 ethtool h1-eth0
Settings for h1-eth0:
  Supported ports: [ ]
  Supported link modes:   Not reported
  Supported pause frame use: No
  Supports auto-negotiation: No
  Supported FEC modes: Not reported
  Advertised link modes:  Not reported
  Advertised pause frame use: No
  Advertised auto-negotiation: No
  Advertised FEC modes: Not reported
  Speed: 10000Mb/s
  Duplex: Full
  Port: Twisted Pair
  PHYAD: 0
  Transceiver: internal
  Auto-negotiation: off
  MDI-X: Unknown
  Link detected: yes
```

The interface clearly runs on 10000Mbits/s. Doing a little digging, we found out that it is set in the point to point topology of the python file.

```
class PointToPoint(Topo):
    def __init__(self, **opts):
        Topo.__init__(self, **opts)
        h1 = self.addHost(’h1’)
        h2 = self.addHost(’h2’)
        self.addLink(h1, h2, bw=10) # note the bw here
                                    # it only allows 10mbits/s transfer rate
```

3.2 b)

Before running the measurement of transfer rate, we execute a ping

```
h2 ping h1
```

and here are the results

```
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  4] local 10.0.0.2 port 5001 connected with 10.0.0.1 port 41780
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0-60.4 sec  68.9 MBytes  9.56 Mbits/sec
[  4] local 10.0.0.2 port 5001 connected with 10.0.0.1 port 41782
[  4]  0.0-15.0 sec  17.1 MBytes  9.57 Mbits/sec
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=0.032 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.046 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.046 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.044 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.045 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.058 ms
64 bytes from 10.0.0.1: icmp_seq=7 ttl=64 time=0.047 ms
64 bytes from 10.0.0.1: icmp_seq=8 ttl=64 time=0.048 ms
64 bytes from 10.0.0.1: icmp_seq=9 ttl=64 time=0.045 ms
64 bytes from 10.0.0.1: icmp_seq=10 ttl=64 time=0.047 ms
64 bytes from 10.0.0.1: icmp_seq=11 ttl=64 time=0.045 ms
64 bytes from 10.0.0.1: icmp_seq=12 ttl=64 time=0.044 ms
64 bytes from 10.0.0.1: icmp_seq=13 ttl=64 time=0.044 ms
64 bytes from 10.0.0.1: icmp_seq=14 ttl=64 time=0.047 ms
```

Current delay is 0.047ms, it is due to transmission.(there is no visible propagation delay on mininet). we run the following code to have the measure of transfer rate on the background, and we started to run ping

```
h1 iperf -c h2 -i 10 -t 120 &
h2 ping h1
```

and the results

```
[  4] local 10.0.0.2 port 5001 connected with 10.0.0.1 port 41786
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=17.0 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=14.7 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=12.6 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=17.7 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=16.3 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=13.8 ms
64 bytes from 10.0.0.1: icmp_seq=7 ttl=64 time=18.1 ms
64 bytes from 10.0.0.1: icmp_seq=8 ttl=64 time=15.8 ms
64 bytes from 10.0.0.1: icmp_seq=9 ttl=64 time=13.8 ms
64 bytes from 10.0.0.1: icmp_seq=10 ttl=64 time=18.1 ms
64 bytes from 10.0.0.1: icmp_seq=11 ttl=64 time=15.7 ms
64 bytes from 10.0.0.1: icmp_seq=12 ttl=64 time=13.9 ms
```

Almost certain, we see a delay here. It is not transmission delay(previous ping is mostly due to transmission, and there is no change in the data rate)(notice: transmission delay = packet size / transfer rate), and it is not propagation delay(we are on a single machine). Hence, the queuing delay is the main reason.

By running our measurement of transfer rate on the background, we took most of the bandwidth, and the queue started to build up in order to diliver all the packets. Hence, it causes the delay.

4 Problem 1.4

4.1 a)

No, there is no impact on the round-trip time on h3 and h4 Before iperf:

```
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=0.230 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=0.040 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=0.043 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=0.049 ms
64 bytes from 10.0.0.4: icmp_seq=5 ttl=64 time=0.045 ms
```

```
64 bytes from 10.0.0.4: icmp_seq=6 ttl=64 time=0.041 ms
64 bytes from 10.0.0.4: icmp_seq=7 ttl=64 time=0.040 ms
```

After iperf:

```
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=0.241 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=0.052 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=0.049 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=0.043 ms
64 bytes from 10.0.0.4: icmp_seq=5 ttl=64 time=0.044 ms
64 bytes from 10.0.0.4: icmp_seq=6 ttl=64 time=0.040 ms
64 bytes from 10.0.0.4: icmp_seq=7 ttl=64 time=0.068 ms
64 bytes from 10.0.0.4: icmp_seq=8 ttl=64 time=0.036 ms
64 bytes from 10.0.0.4: icmp_seq=9 ttl=64 time=0.037 ms
64 bytes from 10.0.0.4: icmp_seq=10 ttl=64 time=0.039 ms
64 bytes from 10.0.0.4: icmp_seq=11 ttl=64 time=0.052 ms
```

4.2 b)

We start another server on h4, and we run the measurement from h1 to h2 and from h3 to h4 result of h1 to h2:

```
[  3] local 10.0.0.1 port 41812 connected with 10.0.0.2 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 5.0 sec  6.00 MBytes  10.1 Mbits/sec
[  3]  5.0-10.0 sec  5.62 MBytes  9.44 Mbits/sec
[  3] 10.0-15.0 sec  5.75 MBytes  9.65 Mbits/sec
[  3] 15.0-20.0 sec  5.75 MBytes  9.65 Mbits/sec
```

result of h3 to h4:

```
[  3] local 10.0.0.3 port 59466 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 5.0 sec  6.00 MBytes  10.1 Mbits/sec
[  3]  5.0-10.0 sec  5.62 MBytes  9.44 Mbits/sec
[  3] 10.0-15.0 sec  5.75 MBytes  9.65 Mbits/sec
[  3] 15.0-20.0 sec  5.75 MBytes  9.65 Mbits/sec
[  3] 20.0-25.0 sec  5.62 MBytes  9.44 Mbits/sec
[  3] 25.0-30.0 sec  5.62 MBytes  9.44 Mbits/sec
```

Clearly, there is no impact on each other

5 Problem 1.5

5.1 a)

5.1.1 from h1 to h4, from h3 to h2

data from h1 to h4

```
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 5.0 sec  6.00 MBytes  10.1 Mbits/sec
[  3]  5.0-10.0 sec  5.62 MBytes  9.44 Mbits/sec
[  3] 10.0-15.0 sec  7.12 MBytes  12.0 Mbits/sec
[  3] 15.0-20.0 sec  6.25 MBytes  10.5 Mbits/sec
[  3] 20.0-25.0 sec  6.00 MBytes  10.1 Mbits/sec
[  3] 25.0-30.0 sec  7.88 MBytes  13.2 Mbits/sec
[  3] 30.0-35.0 sec  4.75 MBytes  7.97 Mbits/sec
[  3] 35.0-40.0 sec  8.75 MBytes  14.7 Mbits/sec
[  3] 40.0-45.0 sec  3.24 MBytes  5.44 Mbits/sec
[  3] 45.0-50.0 sec  7.08 MBytes  11.9 Mbits/sec
[  3] 50.0-55.0 sec  6.90 MBytes  11.6 Mbits/sec
[  3] 55.0-60.0 sec  6.90 MBytes  11.6 Mbits/sec
[  3]  0.0-60.5 sec  76.5 MBytes  10.6 Mbits/sec
```

data from h3 to h2

```
------------------------------------------------------------
[  3] local 10.0.0.3 port 36712 connected with 10.0.0.2 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 5.0 sec  6.38 MBytes  10.7 Mbits/sec
[  3]  5.0-10.0 sec  6.38 MBytes  10.7 Mbits/sec
[  3] 10.0-15.0 sec  6.12 MBytes  10.3 Mbits/sec
[  3] 15.0-20.0 sec  6.75 MBytes  11.3 Mbits/sec
[  3] 20.0-25.0 sec  7.38 MBytes  12.4 Mbits/sec
[  3] 25.0-30.0 sec  4.62 MBytes  7.76 Mbits/sec
[  3] 30.0-35.0 sec  8.00 MBytes  13.4 Mbits/sec
[  3] 35.0-40.0 sec  6.38 MBytes  10.7 Mbits/sec
[  3] 40.0-45.0 sec  7.02 MBytes  11.8 Mbits/sec
[  3] 45.0-50.0 sec  6.96 MBytes  11.7 Mbits/sec
[  3] 50.0-55.0 sec  6.77 MBytes  11.4 Mbits/sec
[  3] 55.0-60.0 sec  6.96 MBytes  11.7 Mbits/sec
[  3]  0.0-62.1 sec  79.7 MBytes  10.8 Mbits/sec
```

the measure rate are all 10mbits/s

5.1.2 from h1 to h3, from h2 to h4

data from h1 to h3

```
[  3] local 10.0.0.1 port 35804 connected with 10.0.0.3 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 5.0 sec  5.88 MBytes  9.86 Mbits/sec
[  3]  5.0-10.0 sec  5.62 MBytes  9.44 Mbits/sec
[  3] 10.0-15.0 sec  5.75 MBytes  9.65 Mbits/sec
[  3] 15.0-20.0 sec  5.75 MBytes  9.65 Mbits/sec
[  3] 20.0-25.0 sec  8.38 MBytes  14.1 Mbits/sec
[  3] 25.0-30.0 sec  9.62 MBytes  16.1 Mbits/sec
[  3] 30.0-35.0 sec  8.95 MBytes  15.0 Mbits/sec
[  3] 35.0-40.0 sec  5.90 MBytes  9.90 Mbits/sec
[  3] 40.0-45.0 sec  2.80 MBytes  4.69 Mbits/sec
[  3] 45.0-50.0 sec  3.60 MBytes  6.05 Mbits/sec
[  3] 50.0-55.0 sec  3.60 MBytes  6.05 Mbits/sec
[  3] 55.0-60.0 sec  2.42 MBytes  4.07 Mbits/sec
[  3] 60.0-65.0 sec  5.10 MBytes  8.55 Mbits/sec
[  3] 65.0-70.0 sec  3.48 MBytes  5.84 Mbits/sec
[  3] 70.0-75.0 sec  3.42 MBytes  5.73 Mbits/sec
[  3] 75.0-80.0 sec  1.74 MBytes  2.92 Mbits/sec
[  3]  0.0-80.1 sec  82.0 MBytes  8.59 Mbits/sec
```

data from h2 to h4

```
[  3] local 10.0.0.2 port 46354 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 5.0 sec  1.50 MBytes  2.52 Mbits/sec
[  3]  5.0-10.0 sec  2.25 MBytes  3.77 Mbits/sec
[  3] 10.0-15.0 sec  2.44 MBytes  4.09 Mbits/sec
[  3] 15.0-20.0 sec  3.79 MBytes  6.36 Mbits/sec
[  3] 20.0-25.0 sec  2.24 MBytes  3.75 Mbits/sec
[  3] 25.0-30.0 sec  2.17 MBytes  3.65 Mbits/sec
[  3] 30.0-35.0 sec  2.98 MBytes  5.00 Mbits/sec
[  3] 35.0-40.0 sec  2.11 MBytes  3.54 Mbits/sec
[  3] 40.0-45.0 sec  3.11 MBytes  5.21 Mbits/sec
[  3] 45.0-50.0 sec  2.36 MBytes  3.96 Mbits/sec
[  3] 50.0-55.0 sec  2.24 MBytes  3.75 Mbits/sec
[  3] 55.0-60.0 sec  2.49 MBytes  4.18 Mbits/sec
[  3]  0.0-62.0 sec  29.7 MBytes  4.02 Mbits/sec
```

the measure rate for h1 to h3 is 8.59mbits/s, and h2 to h4 is 4.02mbits/s.

5.1.3 explaination

As stated in the iperf document, the measurements are unidirectional. In the first experiment, h1 to h4 comsumes uplink from s1 to s2, and from h3 to h2 comsumes downlink from s1 to s2

In the second experiment, the two measurements all comsume the uplink from s1 to s2

5.2 b)

data from h1 to h4

```
[  3] local 10.0.0.1 port 49412 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 5.0 sec  6.00 MBytes  10.1 Mbits/sec
[  3]  5.0-10.0 sec  5.75 MBytes  9.65 Mbits/sec
[  3] 10.0-15.0 sec  5.75 MBytes  9.65 Mbits/sec
[  3] 15.0-20.0 sec  5.75 MBytes  9.65 Mbits/sec
[  3] 20.0-25.0 sec  5.62 MBytes  9.44 Mbits/sec
[  3] 25.0-30.0 sec  5.62 MBytes  9.44 Mbits/sec
[  3] 30.0-35.0 sec  5.75 MBytes  9.65 Mbits/sec
[  3] 35.0-40.0 sec  5.75 MBytes  9.65 Mbits/sec
[  3] 40.0-45.0 sec  5.62 MBytes  9.44 Mbits/sec
[  3] 45.0-50.0 sec  5.75 MBytes  9.65 Mbits/sec
[  3] 50.0-55.0 sec  5.75 MBytes  9.65 Mbits/sec
[  3] 55.0-60.0 sec  5.62 MBytes  9.44 Mbits/sec
[  3] 60.0-65.0 sec  5.62 MBytes  9.44 Mbits/sec
[  3] 65.0-70.0 sec  5.75 MBytes  9.65 Mbits/sec
[  3] 70.0-75.0 sec  5.75 MBytes  9.65 Mbits/sec
[  3] 75.0-80.0 sec  5.62 MBytes  9.44 Mbits/sec
[  3]  0.0-80.1 sec  91.5 MBytes  9.59 Mbits/sec
```

data from h3 to h6

```
[  3] local 10.0.0.3 port 36582 connected with 10.0.0.6 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 5.0 sec  5.00 MBytes  8.39 Mbits/sec
[  3]  5.0-10.0 sec  4.38 MBytes  7.34 Mbits/sec
[  3] 10.0-15.0 sec  3.12 MBytes  5.24 Mbits/sec
[  3] 15.0-20.0 sec  1.88 MBytes  3.15 Mbits/sec
[  3] 20.0-25.0 sec  4.50 MBytes  7.55 Mbits/sec
[  3] 25.0-30.0 sec  3.25 MBytes  5.45 Mbits/sec
[  3] 30.0-35.0 sec  3.75 MBytes  6.29 Mbits/sec
[  3] 35.0-40.0 sec  3.75 MBytes  6.29 Mbits/sec
[  3] 40.0-45.0 sec  4.12 MBytes  6.92 Mbits/sec
[  3] 45.0-50.0 sec  3.00 MBytes  5.03 Mbits/sec
[  3] 50.0-55.0 sec  4.62 MBytes  7.76 Mbits/sec
[  3] 55.0-60.0 sec  3.88 MBytes  6.50 Mbits/sec
[  3]  0.0-60.2 sec  45.2 MBytes  6.31 Mbits/sec
```

the measurement from h1 to h4 reaches its expected 10mbps, since the link from h1 to h4 doesn't collide with h3 to h6.

However, for link from h3 to h6, the maximum bandwidth is limited by the link from h3 to s2. Due to loss from s2 to s3, the bandwidth is reduced again(due to resending of packets). Hence, on average, it has 6.31 mbits/s