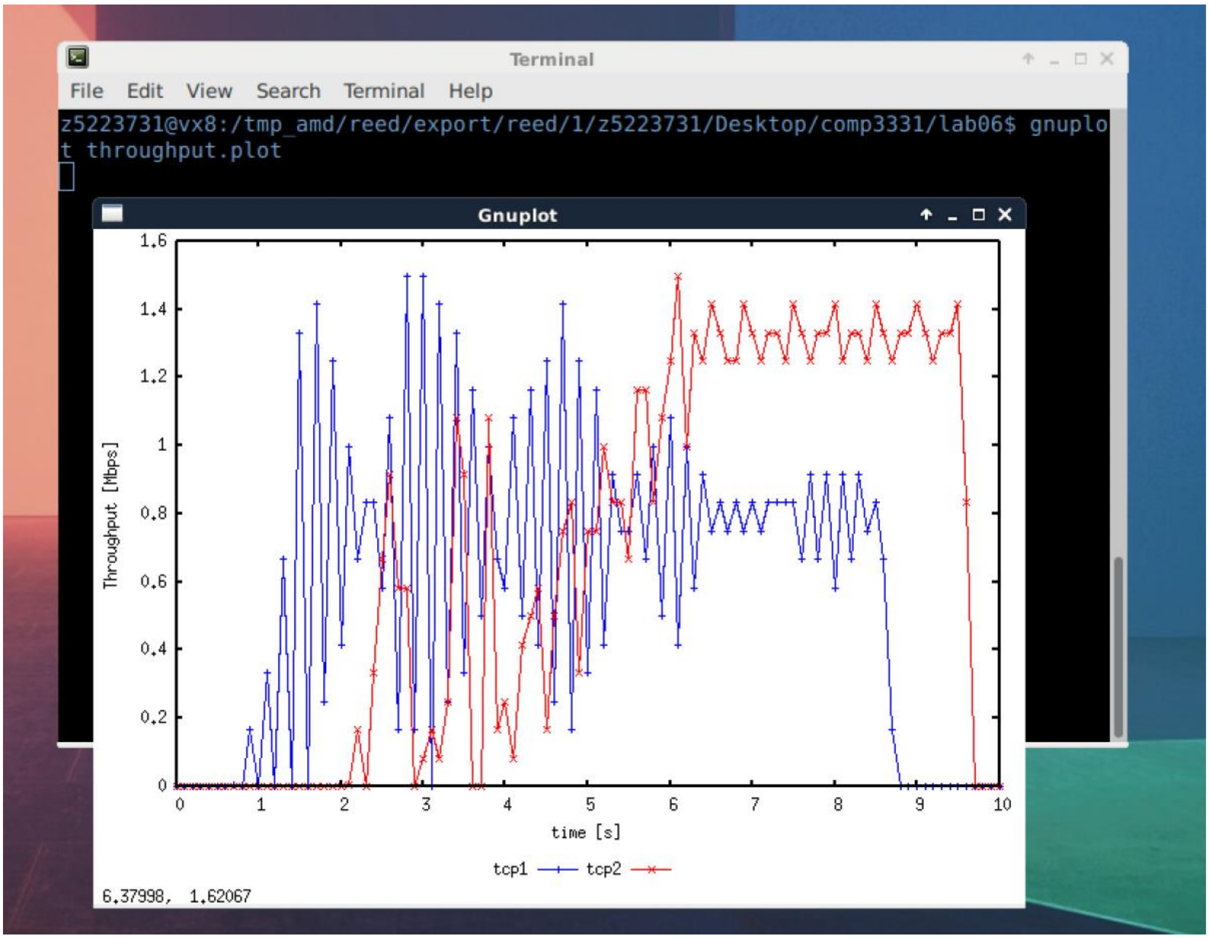
Written by XiaoHu z5223731

Exercise 1 Answer:



```
z5223731@vx8:/tmp_amd/reed/export/reed/1/z5223731/Desktop/comp3331/lab06$ gnuplo
t throughput.plot
```

Question 1: Why the throughput achieved by flow tcp2 is higher than tcp1 between time span 6 sec to 8 sec?

Answer:

Because the n3 -> n2 has a larger bandwidth than n0 -> n1 -> n2. N3 -> n2 will have more package than n1 -> n2, thus the TCP2 will get more bandwidth after adjustment at 6s. therefore, the TCP2 is higher than tcp1.

Question 2: Why the throughput for flow tcp1 is fluctuating between time span 0.5 sec to 2 sec?

Answer: The Congestion control, the TCP1 was doing slow-start at that time.

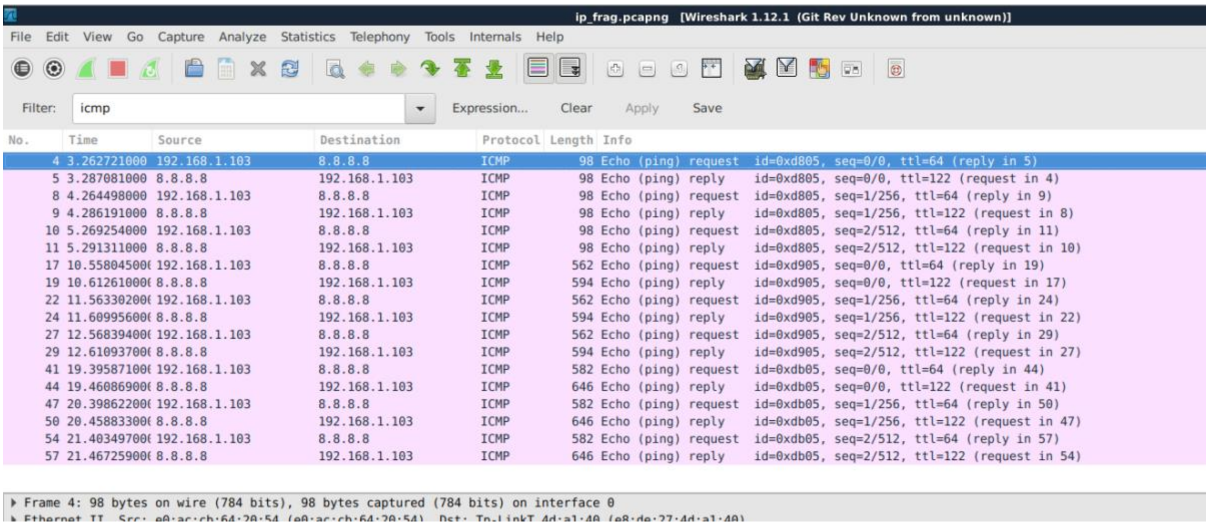
Question 3: Why is the maximum throughput achieved by any one flow capped at around 1.5Mbps?

Answer: because at the beginning, the TCP1 was doing the slow-start and not achieve the maximum bandwidth. After 2s, the TCP2 is joining, both of them have to share the bandwidth, hence the none of them can achieve higher throughput.

Exercise 2 Answer:

Question 1: Which data size has caused fragmentation and why? Which host/router has fragmented the original datagram? How many fragments have been created when data size is specified as 2000?

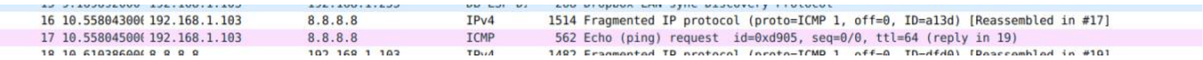
Answer:


ip_frag.pcapng [Wireshark 1.12.1 (Git Rev Unknown from unknown)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: icmp
Expression... Clear Apply Save

```
No.  Time          Source           Destination     Protocol Length Info
   4 3.262721000  192.168.1.103    8.8.8.8         ICMP       98 Echo (ping) request  id=0xd805, seq=0/0, ttl=64 (reply in 5)
   5 3.287081000  8.8.8.8          192.168.1.103   ICMP       98 Echo (ping) reply    id=0xd805, seq=0/0, ttl=122 (request in 4)
   8 4.264498000  192.168.1.103    8.8.8.8         ICMP       98 Echo (ping) request  id=0xd805, seq=1/256, ttl=64 (reply in 9)
   9 4.286191000  8.8.8.8          192.168.1.103   ICMP       98 Echo (ping) reply    id=0xd805, seq=1/256, ttl=122 (request in 8)
  10 5.269254000  192.168.1.103    8.8.8.8         ICMP       98 Echo (ping) request  id=0xd805, seq=2/512, ttl=64 (reply in 11)
  11 5.291311000  8.8.8.8          192.168.1.103   ICMP       98 Echo (ping) reply    id=0xd805, seq=2/512, ttl=122 (request in 10)
  17 10.55804500( 192.168.1.103    8.8.8.8         ICMP      562 Echo (ping) request  id=0xd905, seq=0/0, ttl=64 (reply in 19)
  19 10.61261000( 8.8.8.8          192.168.1.103   ICMP      594 Echo (ping) reply    id=0xd905, seq=0/0, ttl=122 (request in 17)
  22 11.56330200( 192.168.1.103    8.8.8.8         ICMP      562 Echo (ping) request  id=0xd905, seq=1/256, ttl=64 (reply in 24)
  24 11.60995600( 8.8.8.8          192.168.1.103   ICMP      594 Echo (ping) reply    id=0xd905, seq=1/256, ttl=122 (request in 22)
  27 12.56839400( 192.168.1.103    8.8.8.8         ICMP      562 Echo (ping) request  id=0xd905, seq=2/512, ttl=64 (reply in 29)
  29 12.61093700( 8.8.8.8          192.168.1.103   ICMP      594 Echo (ping) reply    id=0xd905, seq=2/512, ttl=122 (request in 27)
  41 19.39587100( 192.168.1.103    8.8.8.8         ICMP      582 Echo (ping) request  id=0xdb05, seq=0/0, ttl=64 (reply in 44)
  44 19.46086900( 8.8.8.8          192.168.1.103   ICMP      646 Echo (ping) reply    id=0xdb05, seq=0/0, ttl=122 (request in 41)
  47 20.39862200( 192.168.1.103    8.8.8.8         ICMP      582 Echo (ping) request  id=0xdb05, seq=1/256, ttl=64 (reply in 50)
  50 20.45883300( 8.8.8.8          192.168.1.103   ICMP      646 Echo (ping) reply    id=0xdb05, seq=1/256, ttl=122 (request in 47)
  54 21.40349700( 192.168.1.103    8.8.8.8         ICMP      582 Echo (ping) request  id=0xdb05, seq=2/512, ttl=64 (reply in 57)
  57 21.46725900( 8.8.8.8          192.168.1.103   ICMP      646 Echo (ping) reply    id=0xdb05, seq=2/512, ttl=122 (request in 54)

▸ Frame 4: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
▸ Ethernet II, Src: e0:ac:cb:64:20:54 (e0:ac:cb:64:20:54), Dst: Tp-LinkT_4d:a1:40 (e8:de:27:4d:a1:40)
```

The data size 2000 and 3500 because the default MTU is 1500.



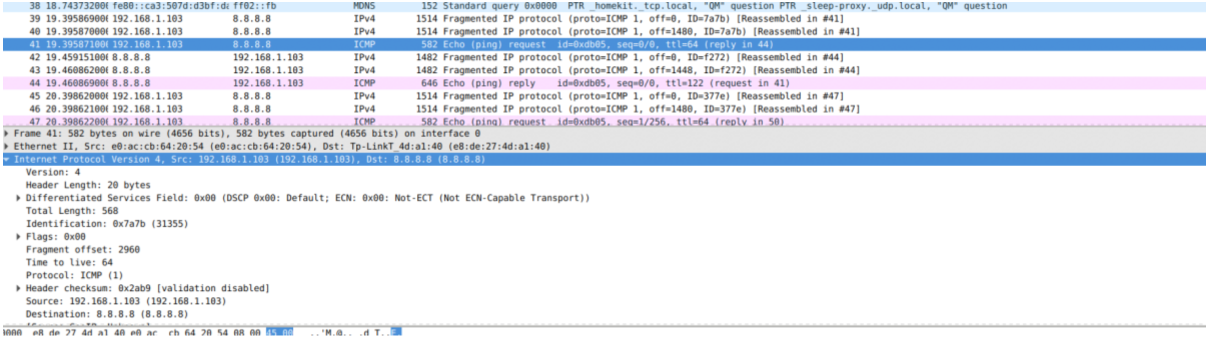
```
  16 10.55804300( 192.168.1.103    8.8.8.8         IPv4     1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=a13d) [Reassembled in #17]
  17 10.55804500( 192.168.1.103    8.8.8.8         ICMP      562 Echo (ping) request  id=0xd905, seq=0/0, ttl=64 (reply in 19)
  18 10.61038600( 8.8.8.8          192.168.1.103   IPv4     1482 Fragmented IP protocol (proto=ICMP 1, off=0, ID=dfd0) [Reassembled in #19]
```

192.168.1.103 has fragmented the original datagram and two fragments have been created.

Question 2: Did the reply from the destination 8.8.8.8. for 3500-byte data size also get fragmented? Why and why not?

Yes, because the MTU is 1500 so that the 3500 has to be fragmented to smaller segments.

Question 3: Give the ID, length, flag and offset values for all the fragments of the first packet sent by 192.168.1.103 with data size of 3500 bytes?



```
  38 18.74373200( fe80::ca3:507d:d3bf:d ff02::fb        MDNS      152 Standard query 0x0000  PTR _homekit._tcp.local, "QM" question PTR _sleep-proxy._udp.local, "QM" question
  39 19.39586900( 192.168.1.103    8.8.8.8         IPv4     1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=7a7b) [Reassembled in #41]
  40 19.39587000( 192.168.1.103    8.8.8.8         IPv4     1514 Fragmented IP protocol (proto=ICMP 1, off=1480, ID=7a7b) [Reassembled in #41]
  41 19.39587100( 192.168.1.103    8.8.8.8         ICMP      582 Echo (ping) request  id=0xdb05, seq=0/0, ttl=64 (reply in 44)
  42 19.45915100( 8.8.8.8          192.168.1.103   IPv4     1482 Fragmented IP protocol (proto=ICMP 1, off=0, ID=f272) [Reassembled in #44]
  43 19.46086200( 8.8.8.8          192.168.1.103   IPv4     1482 Fragmented IP protocol (proto=ICMP 1, off=1448, ID=f272) [Reassembled in #44]
  44 19.46086900( 8.8.8.8          192.168.1.103   ICMP      646 Echo (ping) reply    id=0xdb05, seq=0/0, ttl=122 (request in 41)
  45 20.39862000( 192.168.1.103    8.8.8.8         IPv4     1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=377e) [Reassembled in #47]
  46 20.39862100( 192.168.1.103    8.8.8.8         IPv4     1514 Fragmented IP protocol (proto=ICMP 1, off=1480, ID=377e) [Reassembled in #47]
  47 20.39862200( 192.168.1.103    8.8.8.8         ICMP      582 Echo (ping) request  id=0xdb05, seq=1/256, ttl=64 (reply in 50)
▸ Frame 41: 582 bytes on wire (4656 bits), 582 bytes captured (4656 bits) on interface 0
▸ Ethernet II, Src: e0:ac:cb:64:20:54 (e0:ac:cb:64:20:54), Dst: Tp-LinkT_4d:a1:40 (e8:de:27:4d:a1:40)
▾ Internet Protocol Version 4, Src: 192.168.1.103 (192.168.1.103), Dst: 8.8.8.8 (8.8.8.8)
    Version: 4
    Header Length: 20 bytes
  ▸ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
    Total Length: 568
    Identification: 0x7a7b (31355)
  ▸ Flags: 0x00
    Fragment offset: 2960
    Time to live: 64
    Protocol: ICMP (1)
  ▸ Header checksum: 0x2ab9 [validation disabled]
    Source: 192.168.1.103 (192.168.1.103)
    Destination: 8.8.8.8 (8.8.8.8)
0000  e8 de 27 4d a1 40 e0 ac  cb 64 20 54 08 00 45 00   ..'M.@.. .d T..E.
```

Id:	length:	flag:	offset:
0x7a7a	1500	0x01	0
0x7a7a	1500	0x01	1480
0x7a7a	568	0x00	2960

Question 4: Has fragmentation of fragments occurred when data of size 3500 bytes has been used? Why and why not?

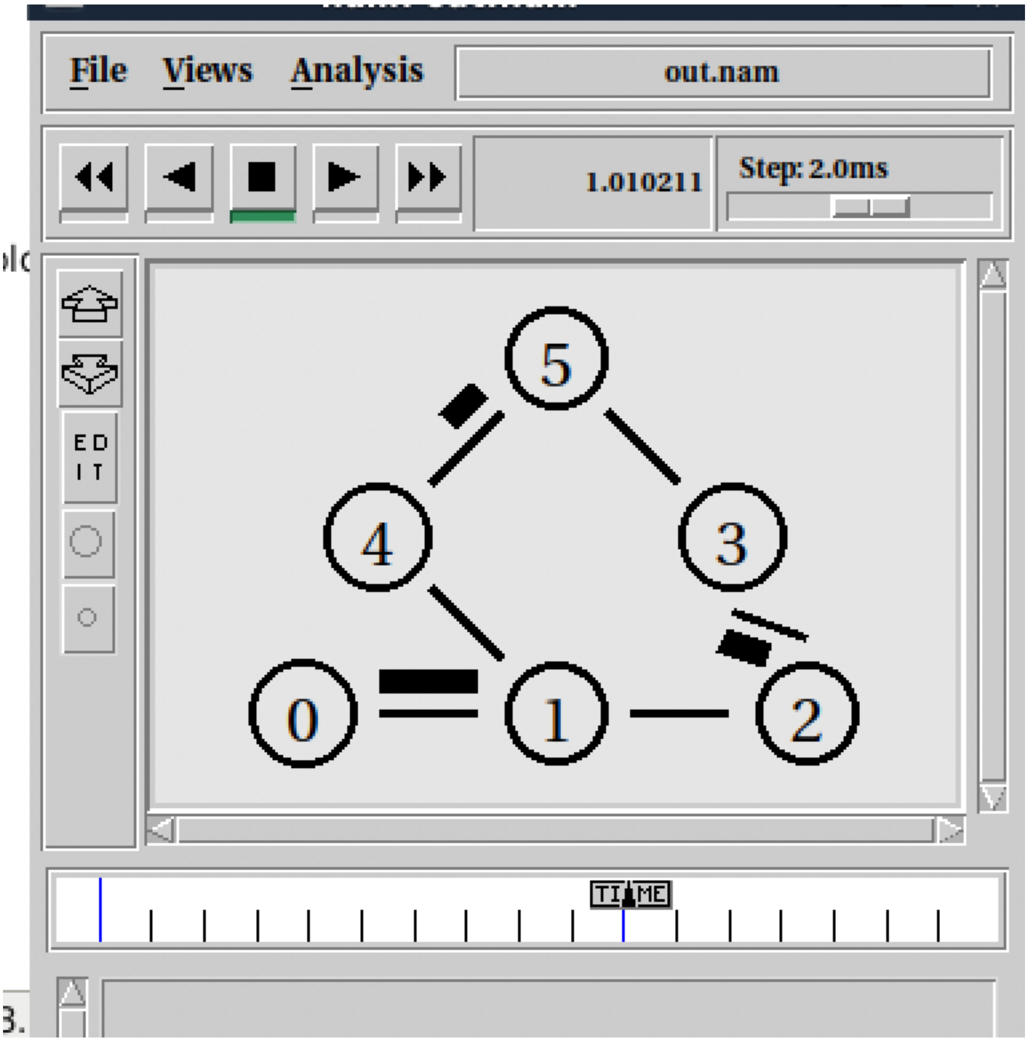
Answer:

There is no fragmentation from 8.8.8.8 to 192.168.1.103, there are three fragments and no fragmentation of fragments. Hence, we have no idea about the fragmentation to both direction and the reassembled only occur the destination.

Question 5: What will happen if for our example one fragment of the original datagram from 192.168.1.103 is lost?
Answer: fragment will be incomplete hence the receiver will discard.

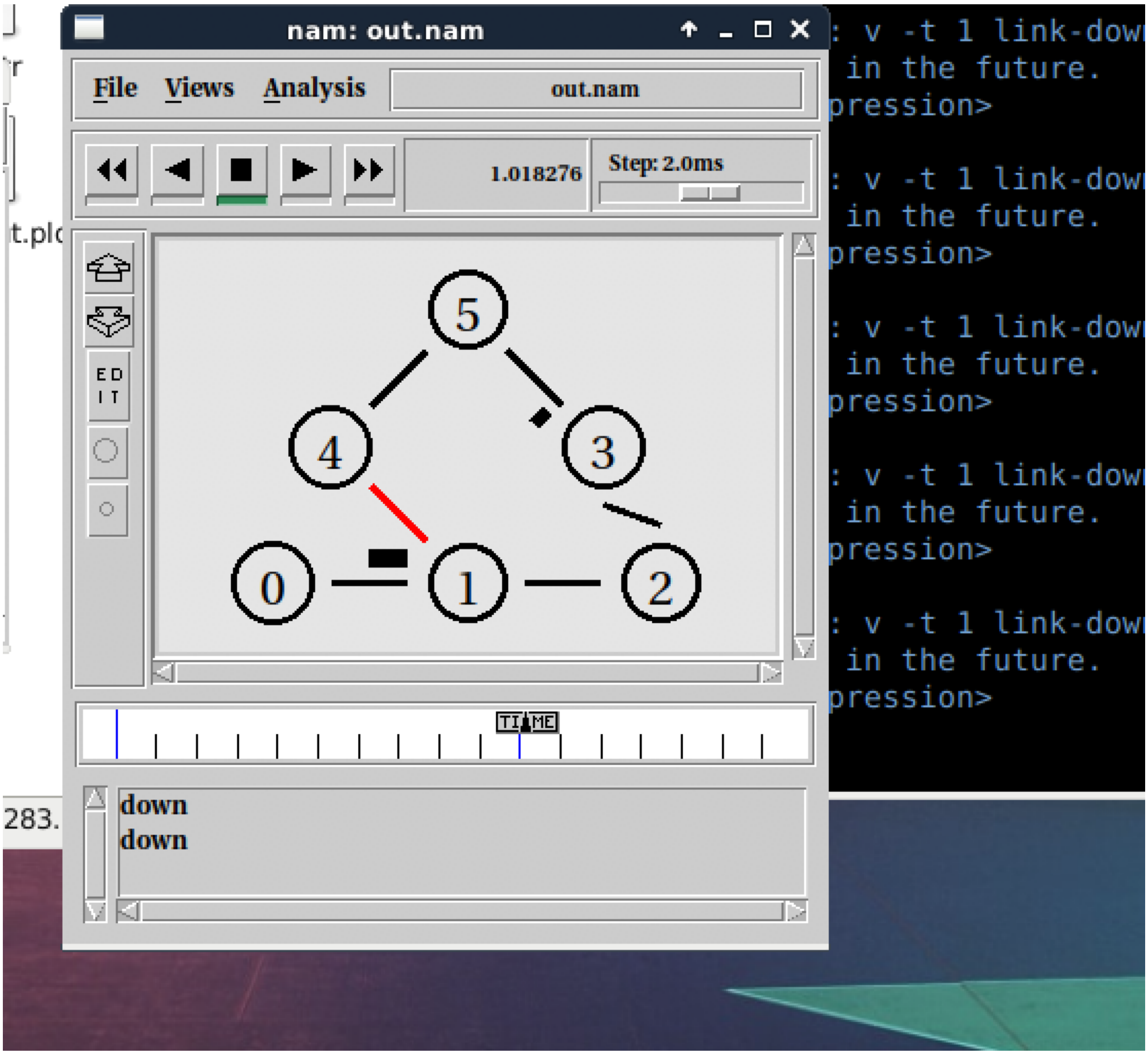
Exercise 3 Answer:

Question 1: Which nodes communicate with which other nodes? Which route do the packets follow? Does it change over time?
Answer:



Node0 -> node1 -> node4 -> node5
Node2 -> node5(2->3->5) doesn't change.

Question 2: What happens at time 1.0 and at time 1.2? Does the route between the communicating nodes change as a result of that?
Answer:



```
: v -t 1 link-dow
 in the future.
pression>
: v -t 1 link-dow
 in the future.
pression>
: v -t 1 link-dow
 in the future.
pression>
: v -t 1 link-dow
 in the future.
pression>
: v -t 1 link-dow
 in the future.
pression>
```

```
down
down
```

At time 1.0, n1 -> n4 goes down. Node0 can’t reach node5 and packet are wait at node1, node2 can still go node5.

At time 1.2, n1 -> n4 goes up. Node0 can reach node5, node1 can go node4.

There is no influence on node2 and 5.

Question 3: Did you observe any additional traffic as compared to Step 3 above? How does the network react to the changes that take place at time 1.0 and time 1.2 now?
Answer:
Yes, when n1 -> n4 is going down, node0 and 5 will use a another path which is 0->1->2->3->5 and when n1->4 is going up, node0 and 5 use 0->1->4->5.

Question 4: How does this change affect the routing? Explain why
Answer:
It will increase the cost of n1->n4 to 3 and the total cost of 0->1->4->5 is 5.
0->1->2->3->5 will be 4, we take the lower cost so that node0 -> 5 will use 0->1->2->3->5.

Question 5: Describe what happens and deduce the effect of the line you just uncommented.

Answer:

Node0 -> node5, the route 0->1->4->5 is still the lower cost but the node2 -> 5, route 2->3->5 and 2->1->4->5 have the same cost 4 and it splits equally on these two.