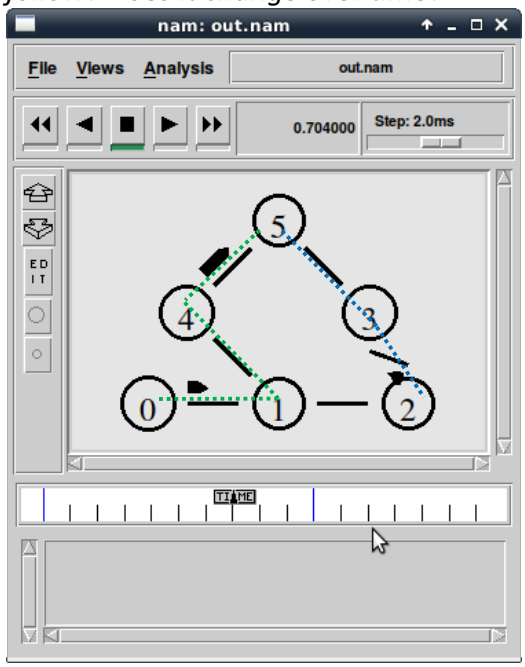
Lab Exercise 6: Routing, Throughput and IP Fragmentation

Z5145114
Xiaodan Wang

Exercise 1: Understanding the Impact of Network Dynamics on Routing

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

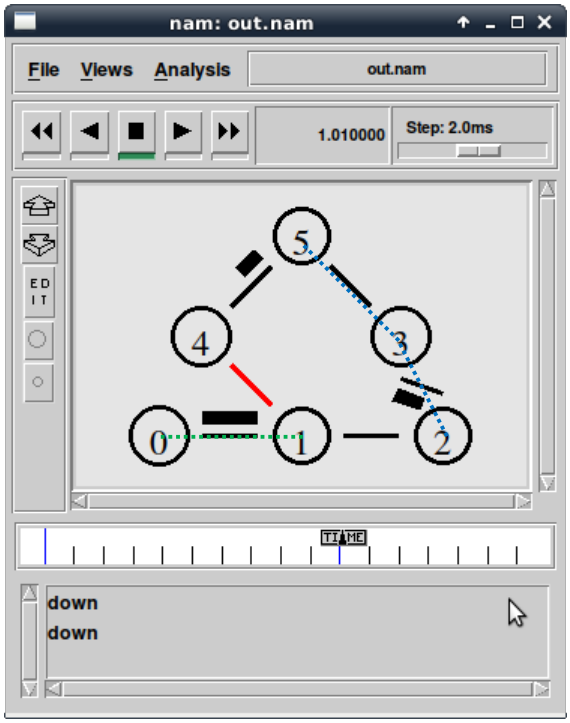


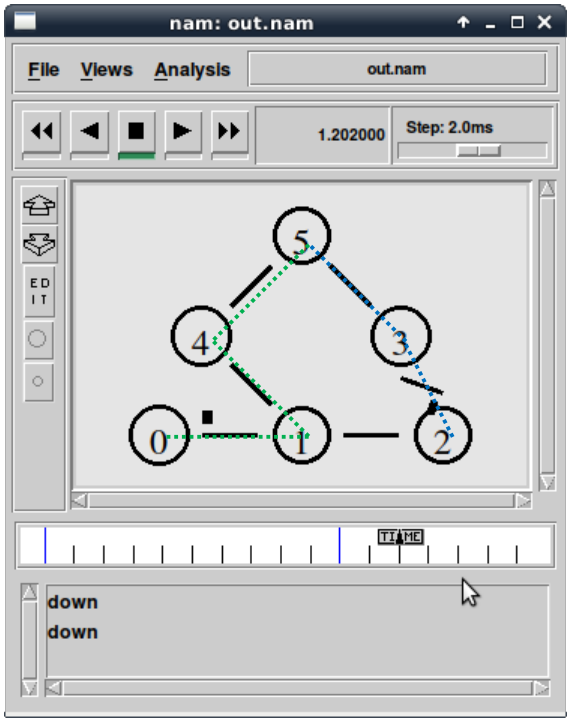
Observe the NAM window output, it is easy to find that the communication between nodes.
The following nodes pairs are communicating:
0-1, 1-4, 4-5, 2-3, 3-5.
The routes packets following are:
0-1-4-5 and 2-3-5.
Since the connection established, it doesn't change over time.

Examining the simulation setting in the script file, same answers can be found.
From line 56 to line 82 in tp_routing.tcl, 2 UDP connections are established. Udp0 is a UDP agent attached to node0 and connects with a null agent node 5. Udp1 is a UDP agent attached to node2 and connects with a null agent node 5.

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?

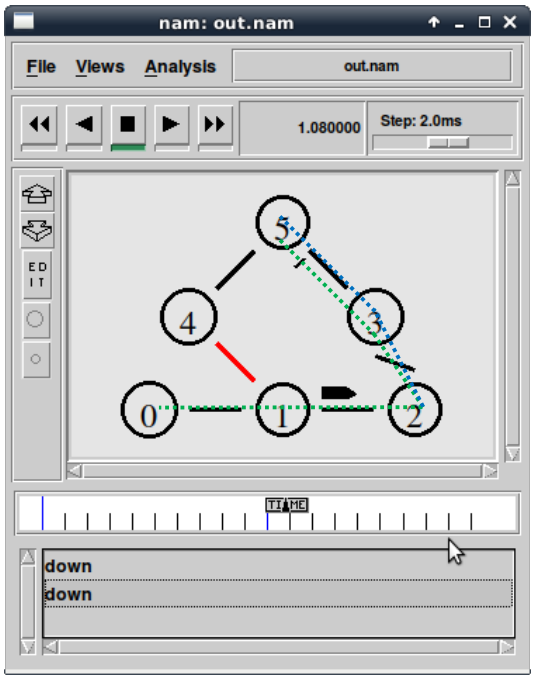
At time 1.0, the route model between node 1 and node4 is down.
At time 1.0, the route model between node 1 and node4 is up.
The route between the communicating nodes doesn't change as it using a static routing protocol.

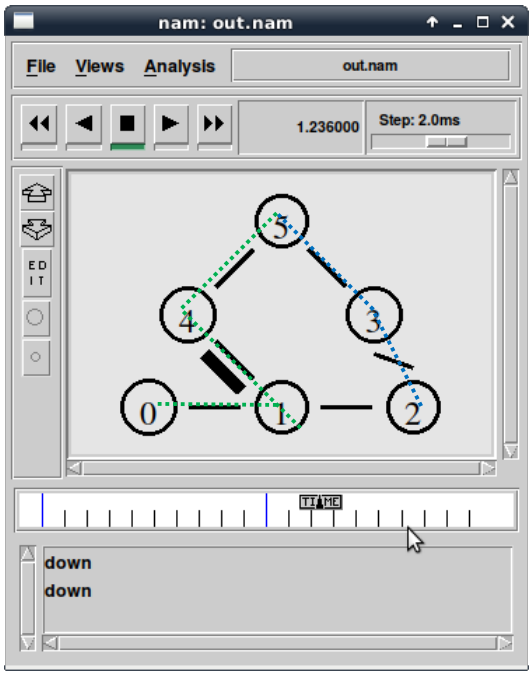




Question 3: Did you observe any additional traffic as compared to Step 3 above? How does the network react to the changes that take place at time 1.0 and time 1.2 now?

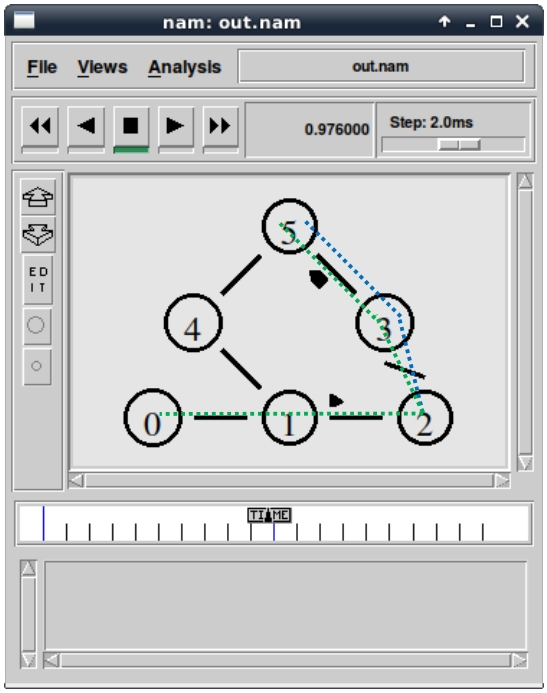
Yes. There is a communication between node1 and node2.
When the route model between node 1 and node4 down at time 1.0, routes changes from 0-1-4-5 to 0-1-2-3-5. The reason is we use a Distance-Vector routing protocol instead of a static routing protocol. In Distance-Vector routing protocol, route 0-1-2-3-5 is better than route 0-1-4-5 as route model 1-4 is down.
However, when the route model between node 1 and node4 up again at time 1.2, routes changes from 0-1-2-3-5 to 0-1-4-5.





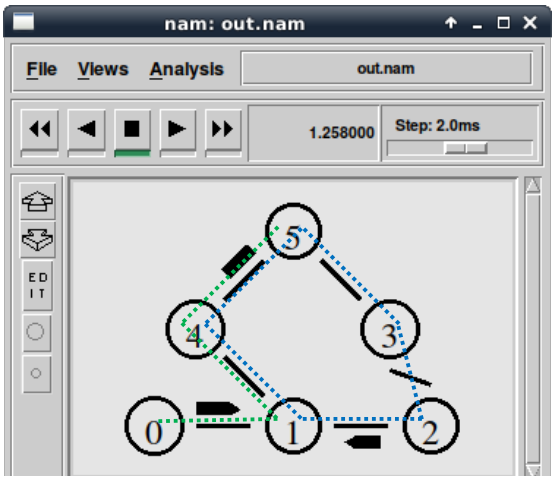
Question 4: How does this change affect the routing? Explain why.

The route from 0-1-4-5 changes to 0-1-2-3-5.
The reason is this line *($ns cost $n1 $n4 3*) indicates that the cost of communicating between node1 and node4 is 3 which makes the cost of 0-1-4-5 higher than cost of 0-1-2-3-5. While we use a Distance-Vector routing protocol, the connection prefers the routing with lower cost.



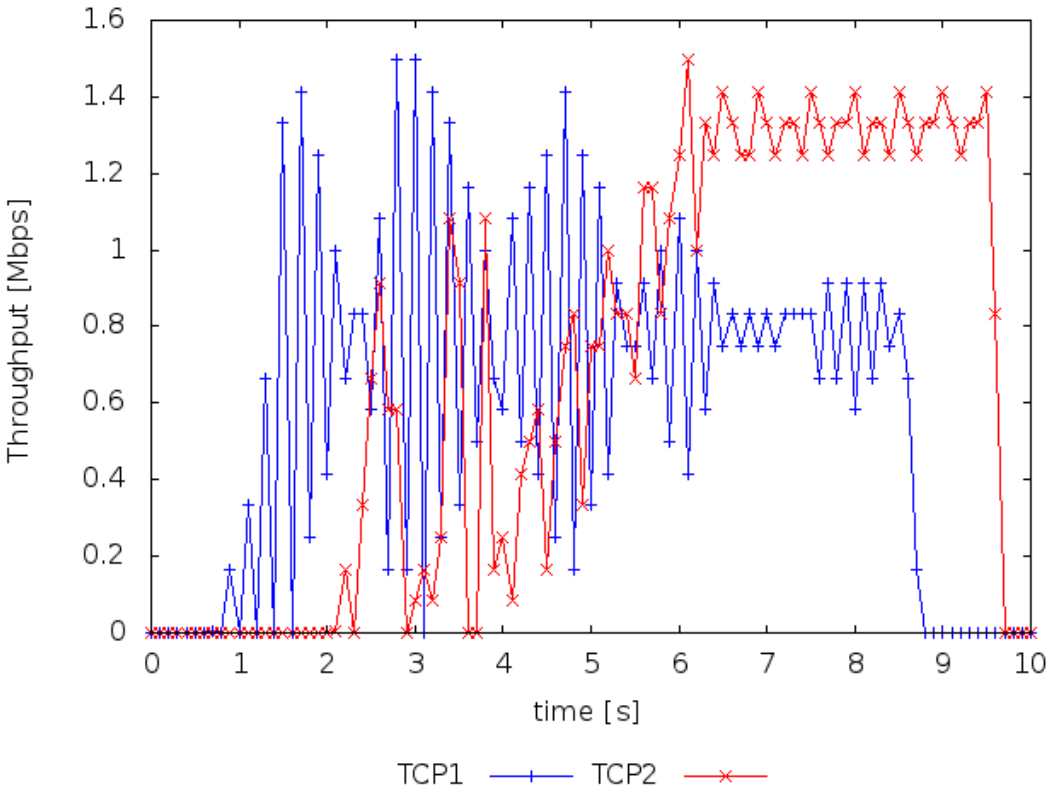
Question 5: Describe what happens and deduce the effect of the line you just uncommented.

The udp1 has multiple path which routing 2-3-5 and 2-1-4-5.
$Node set multiPath_ 1 is setting all nodes for multiple path nodes.
In this way, for udp0 which from node0 to node5, the cost of 0-1-4-5 is 4 while the cost of 0-1-2-3-5 is 6. And the route will be 0-1-4-5.
For udp1 which from node2 to node5, the cost of 2-3-5 is 4 while the cost of 2-1-4-5 is also 4. As all nodes is multiple path nodes, udp1 will communicating in both routes.



Exercise 2: Setting up NS2 simulation for measuring TCP throughput

The completed tcl file (exercise2.tcl) and the script (throughput.plot) for producing the throughput plot are attached. The throughput plot is below.



In the plot above, we can see the slow start of tcp connections. At time 2.8 and 3.6, there is a loss and congestions appear. At time 6.4, tcps become stable. And tcp1 ends at time 8.5 while tcp2 ends at time 9.5.

Exercise 3: Understanding IP Fragmentation

Step 1: Ping with default packet size to the target destination as 8.8.8.8
ping -c 10 8.8.8.8

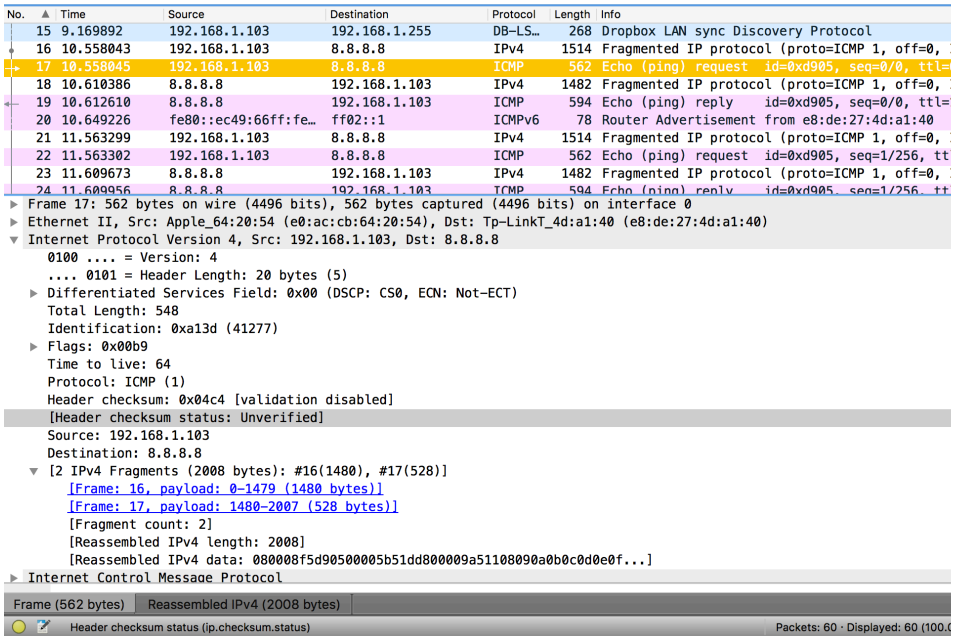
Step 2: Repeat by sending a set of ICMP requests with data of 2000.
ping -s 2000 -c 10 8.8.8.8

Step 3: Repeat again with data size set as 3500
ping -s 3500 -c 10 8.8.8.8

Load this trace file in Wireshark, filter on protocol field ICMP (you may need to clear the filter to see the fragments) and answer the following questions.

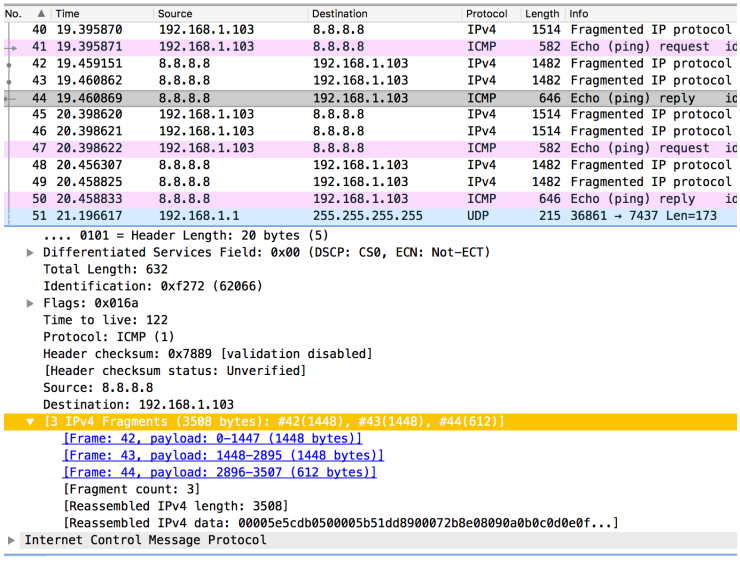
Question 1: Which data size has caused fragmentation and why? Which host/router has fragmented the original datagram? How many fragments have been created when data size is specified as 2000?

Data size 2000 and 3500 both cause fragmentation as the max transfer size is 1500 bytes.
192.168.1.103 has fragmented the original datagram.
2 fragments have been created when data size is specified as 2000.



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, the reply from the destination 8.8.8.8 for 3500 bytes data size also get fragmented. The reason is that when 8.8.8.8 receives the request from 192.168.1.103, it will give a reply base on the request. In this case, it has a 3500 bytes data size reply which needs a fragmentation.



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?

ID	0x7a7b (31355)	0x7a7b (31355)	0x7a7b (31355)
Length	1500	1500	568
Flag	0x2000, More fragments	0x20b9, More fragments	0x0172
Offset values	0	185	370

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

There is no fragmentation of fragments.
It can be observed by the length of fragments. When data size is 3500 bytes, there are 3 fragments and their length are 1500, 1500 and 568. As the max transfer size is 1500 bytes, there is no need to has fragmentation of fragments.

Question 5: What will happen if for our example one fragment of the original datagram from 192.168.1.103 is lost?

If one fragment of the original datagram is lost, the receiver (8.8.8.8) cannot reassemble the datagram. It will ask the sender resend the whole datagram again.