Lab 6

Exercise 1: Understanding the Impact of Network Dynamics on Routing

Question 1:

Answer:

Node 0 sends packets to Node 5, the transmitted packets follow the route 0-1-4-5, and the route does not change over time.

Node 2 sends packets to Node 5, the transmitted packets follow the route 2-3-5, and its route does not change either.

Question 2:

Answer:

At time 1.0, link 1-4 goes down, but the route between Node 1 and Node 5 does not change. And the Node 0 can't reach Node 5 at that time.

At time 1.2, link 1-4 goes up, and the packets can be forwarded again, and Node 0 can reach Node 5 again.

Question 3:

Answer:

Now, when link 1-4 goes down, the DV routing protocol learns the change of the network and changes the route to 0-1-2-3-5. However, when the link 1-4 goes up again, the routing protocol makes the route become 0-1-4-5 again. This is because the original route (0-1-4-5) has a lower cost (the number of hops to destination.)

Question 4:

Answer:

This code changes the cost of link 1-4 to 3 (the initial cost of links is 1). And the route becomes 0-1-2-3-5, but not the 0-1-4-5. This is because the cost (4) of route 0-1-2-3-5 is lower than the cost (5) of route 0-1-4-5.

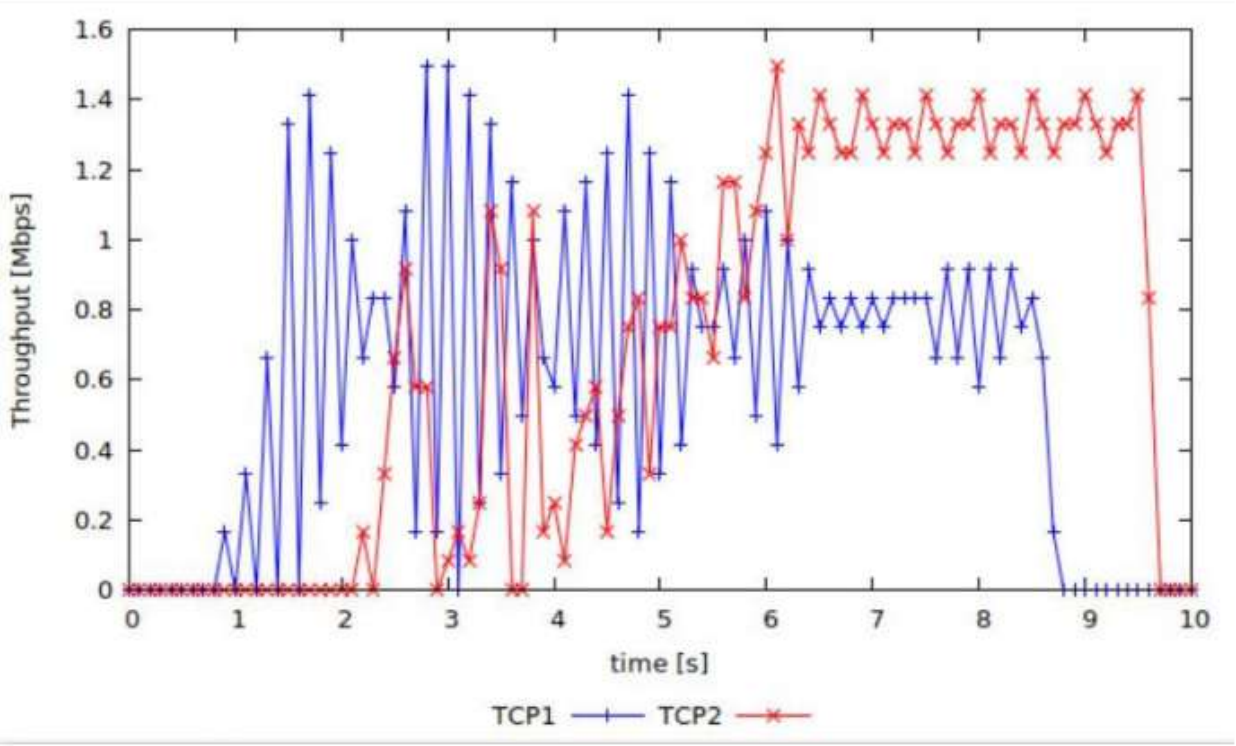
Question 5:

Answer:

The number of route from Node 2 to Node 5 is two, one is 2-3-5 and the other is 2-1-4-5. Since the cost of link 1-4 is changed to 2, and the cost of link 3-5 is changed to 3, the cost of these two routes has equal cost to destination. In addition, the "Node set multiPath_ 1" makes the network use multipath routing, so the data from node 2 will split traffic equally on both these two paths.

Exercise 2: Setting up NS2 simulation for measuring TCP throughput

Here is the graph of TCP throughput,



Question 1:

Answer:

Because flow tcp1 has to traverse more router than flow tcp2 does , and it has a longer delay.

Question 2:

Answer:

Because tcp 1 is using Slow-start mechanism to detect available bandwidth.

Question 3:

Answer:

Because the amount of data that sender can reach is the minimum of rend and the cwnd.

Exercise 3: Understanding IP Fragmentation

Question 1:

Answer:

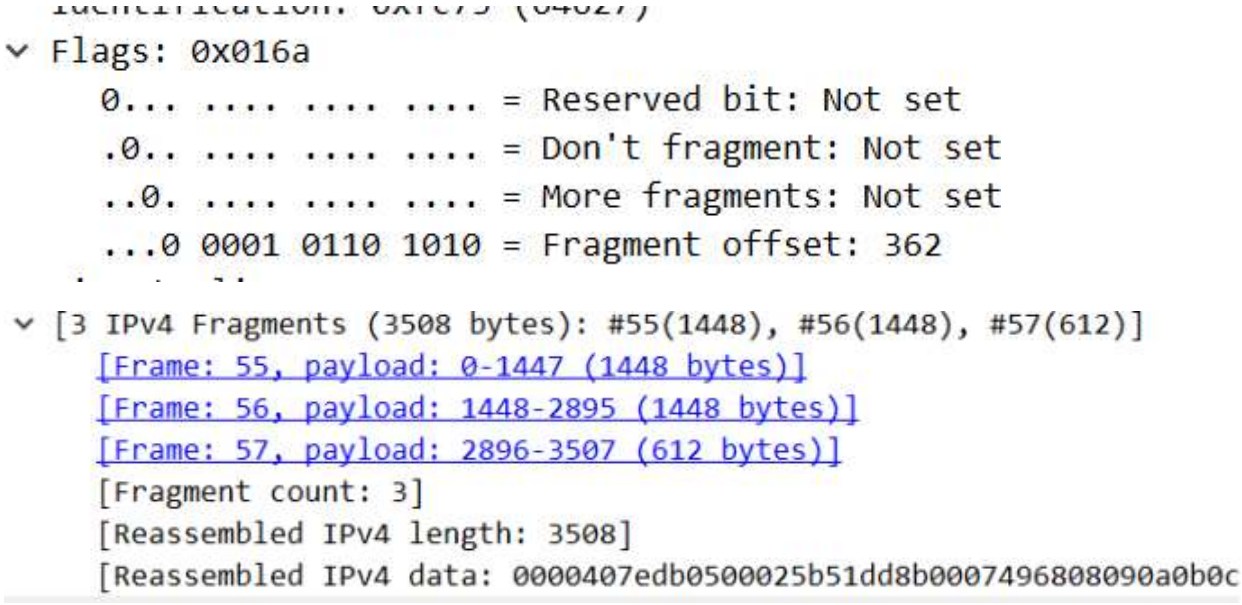
The data size 2000 and 3500 caused fragmentation since their 'offset' is set like below. That means the data is separated and 'offset' is a flag to recombine the data. The 192.168.1.103 fragmented the original datagram. In data size of 2000, there are two fragments created.

```
...0 0000 1011 1001 = Fragment offset: 185

...0 0001 0111 0010 = Fragment offset: 370

▾ [2 IPv4 Fragments (2008 bytes): #16(1480), #17(528)]
    [Frame: 16, payload: 0-1479 (1480 bytes)]
    [Frame: 17, payload: 1480-2007 (528 bytes)]
    [Fragment count: 2]
    [Reassembled IPv4 length: 2008]
    [Reassembled IPv4 data: 080008f5d90500005b51dd800009a5110809
```

Question 2:

Answer:

The reply from 8.8.8.8 for 3500-byte data size also gets fragmented. Like the below, we can find that the ICMP reply has the same data length with the ICMP request. Therefore, the data size is too large to transmit and it will be fragmented.



Question 3:

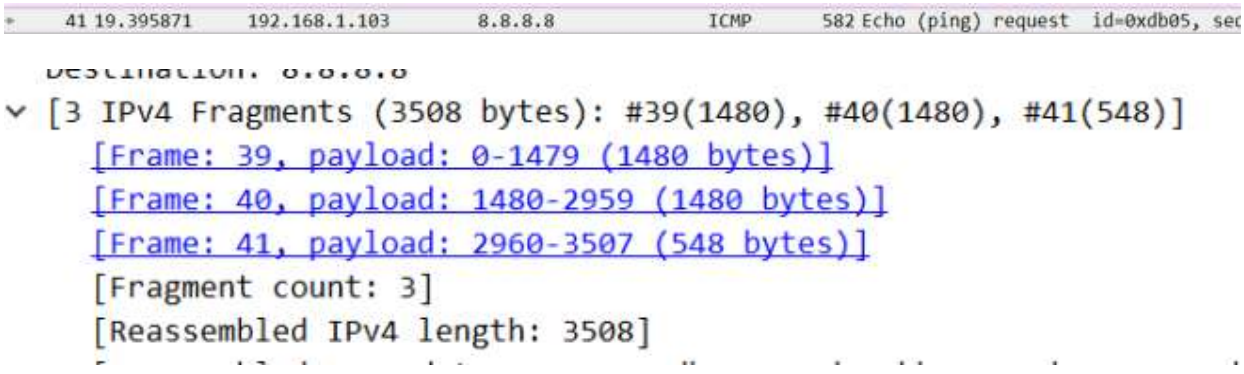
Answer:

The first packet sent by 192.168.1.103 with data size of 3500 bytes has this information respectively:

ID 0xdb05, length 1480, flag 1, offset 000

ID 0xdb05, length 1480, flag 1, offset 185

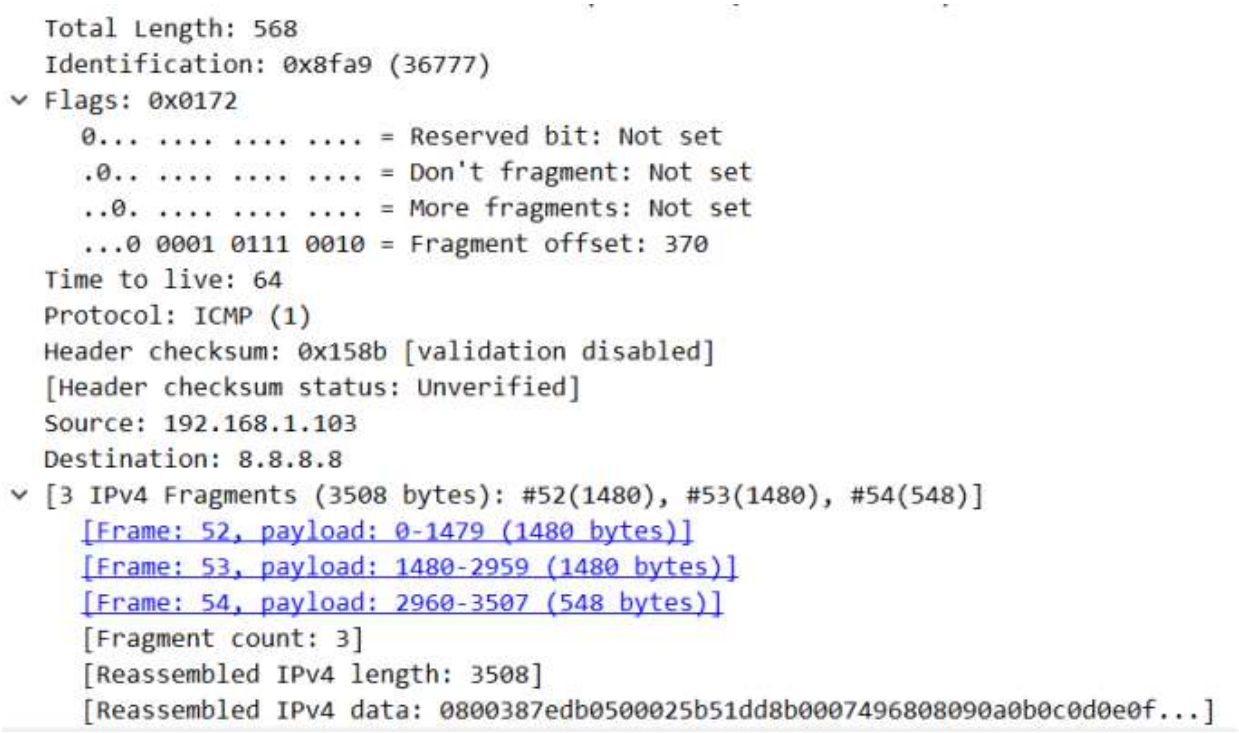
ID 0xdb05, length 548, flag 0, offset 370



Question 4:

Answer:

We can't be sure. From data from Wireshark, we can only observe the situation when offset = 370 and length = 568 (548 + 20 IP header), which means that at least in the last packet of the fragment, no further fragmentation occurs. But we're not sure what happens when offset = 000 and offset = 185, so I don't think we can determine whether it is fragmented. If you want to know whether there is any fragmentation, there should have offset value which is 0<offset<185 or 185<offset<370.



Question 5:

Answer:

This causes the entire IP packet to be discarded, and if retransmission is required, it is implemented by the upper layer protocol of the network layer (IP).

Therefore, we should try to avoid IP fragmentation so as to avoid unnecessary multiple transmission.