CSP334: Computer Networks, Lab Assignment No 3,Traceroute ping commands

Rahul Byas Sherwan
Entry No. : 2016UCS0028

1: The First Problem

(a) What if there was no TTL field in the invocation of the traceroute at all?

Traceroute works on the principal of TTL of a packet and loopbacks (that is upto how much time the packet is in the network). If there was no TTL field in the invocation of the traceroute then we will only be able to see the IP address (and other informations such as time etc) of the destination system. We will not be able to see the IPAs of the hops in between the source and destination as no loopbacks is possible .
The other drawback of not having a TTL concept is it will create congestion in the network.

(b) How will the routers in between determine whether the TTL value limit has reached ?

Each router has a to do only two things,i.e.
1. Decrement the TTL by 1.
2. Compare whether the value is zero or not.If it is zero then the value limit has reached , otherwise it will loopback and go to the next hop.
This two steps have to be repeated by each hop.
For example : Suppose a packet has x TTL , then the first hop will decrement its TTL (now after decrementing its TTL is x-1) and forward it to next hop . This process will keep on repeating by each hop unless it has reached to zero . In that case the router will come to know that the TTL value has reached.

(c) Should an intermediate router that receives a traceroute packet always respond with an ICMP TTL exceeded message ? If the answer is a yes, reason why and if the answer is a no, then argue how do we know the address of all the routers/hops in between us and the destination ?

No, not always respond with an ICMP TTL exceeded message.It will respond this message only when the incoming packet has TTL 1.
Since , the IP address of the destination or the last hop which respond with ICMP TTL exceeded message is written on the packet that this IPA has responded with ICMP message. So, in this manner while loopingback each hop just forward it to the nearest hop as the source IP is written on it.

(d) Why does traceroute make use of a destination UDP port number which is

invalid - i.e. it sends a packet to a UDP port in the range h33434 to 33534i ?
If we are using traceroute it means we just want to know the IP and we don't want any service . That's why traceroute uses destination UDP port number which is invalid.

(e) How do we know the address of all the routers/hops in between us and the destination when using the traceroute?
Since traceroute uses loopbacks for acknowledging each hop. So, what it does is, it first go to the nearest hop and loopback and after then to the next hop nearest to the previous hop and again loopbacks. It repeats this process again and again untill TTL limit is reached or untill destination is not found. While looping back it gives the address of the perticular hop. So, in this way , we know the address of all the routers/hops between us and the destination when using the traceroute.

(f) How is traceroute latency calculated?

A roundtrip time of a single packet is its latency . So, When we send a packet ,we recorded that time and when the packet is recieved ,we recorded that time too . Then we subtract the initial time from the final time . That difference is the latency.

2: The Second Problem

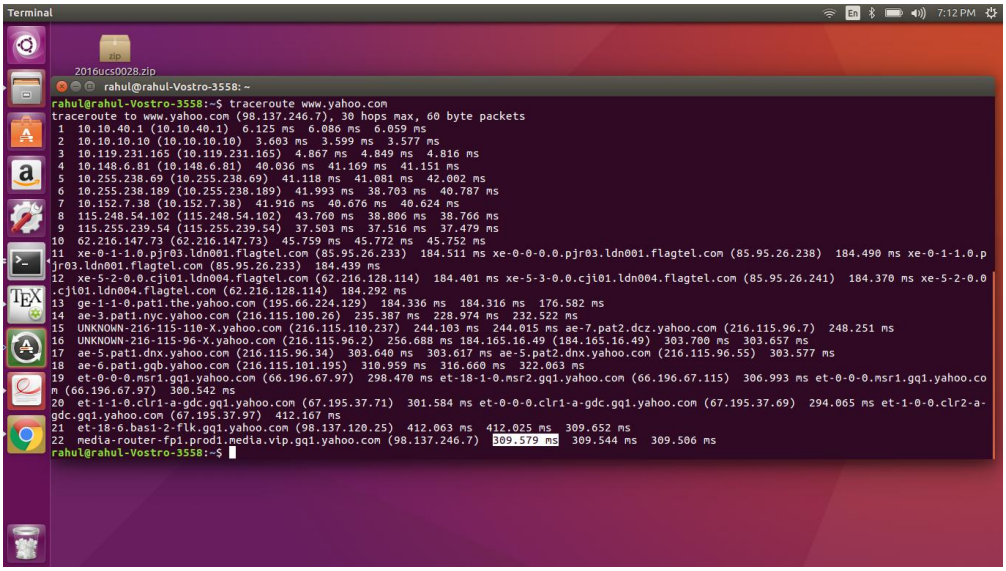

Figure 1: traceroute

As you can see in the image above IP address of yahoo.com that was used for the trace route : 98.137.246.7
The number of iterations required to determine route : 22
The IP addresses of all the machines between the source and the destination : We can see that in the screendump above.
The average round trip time of the packet that reached the destination: 309.543 ms

3: The Third Problem

After running the command screendump :

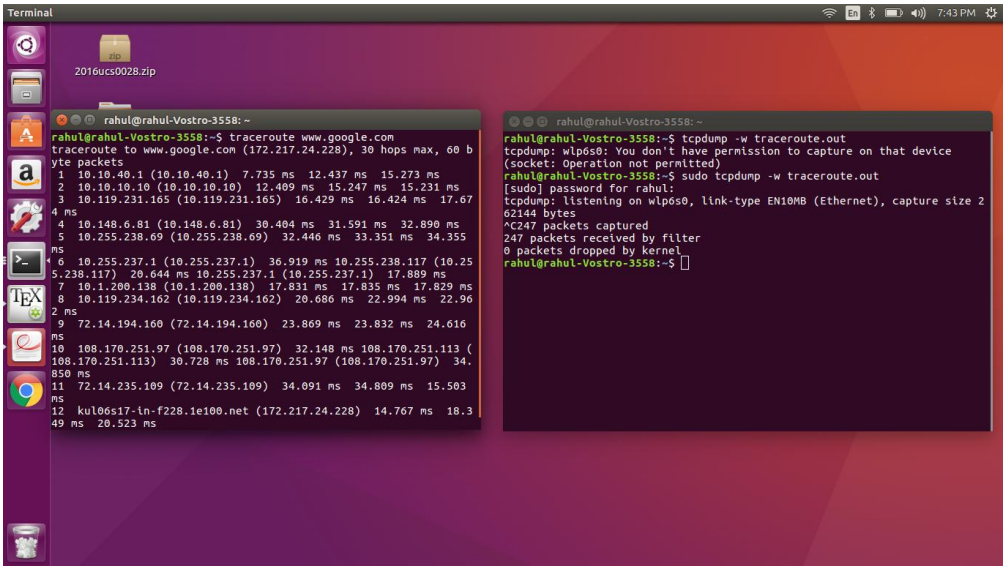

Figure 2: tcpdump listening

(a) How many packets are send by traceroute in each iteration ? How can you prove this using the tcpdump output?

Packets that are send by traceroute in each iteration: 3

As we can see in the tcpdump output that 3 packets have been send to IPA 72.14.194.160 and IPA 10.119.234.162 (screenshot below).

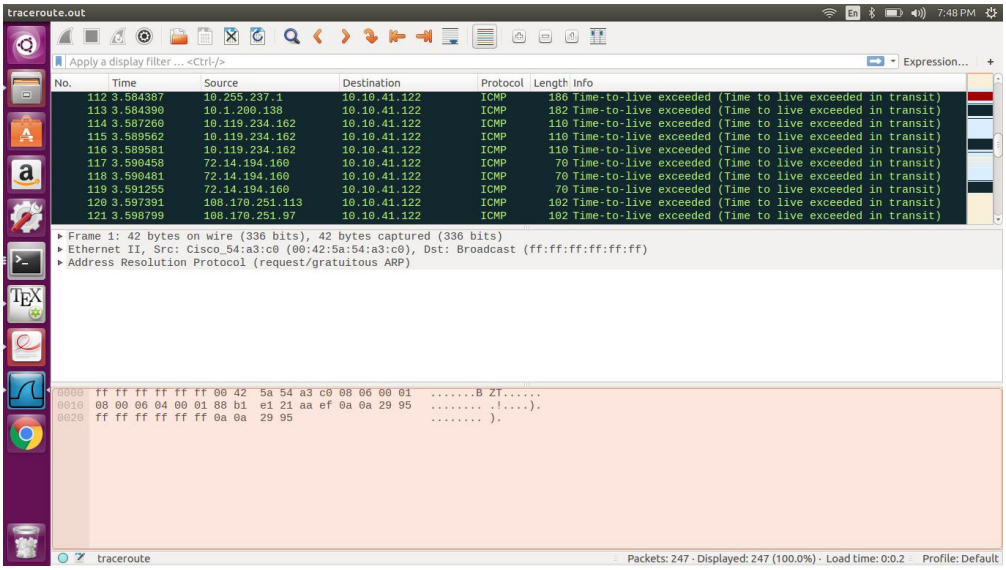

Figure 3: tcpdump output

(b) Consider one specific iteration of traceroute invocation/iteration. For this specific iteration, what are the individual round trip times of each of the three probes sent ? What is the average round trip time ? Does it match with the round trip time returned by traceroute ?

I am considering iteration of 10.10.40.1. The individual round trip time are :

1 . 7.735ms (can see in the screenshot below)

2. 12.437 ms

3. 15.273 ms

Average :11.815 ms

Yes, it approximately matches the average round trip time.

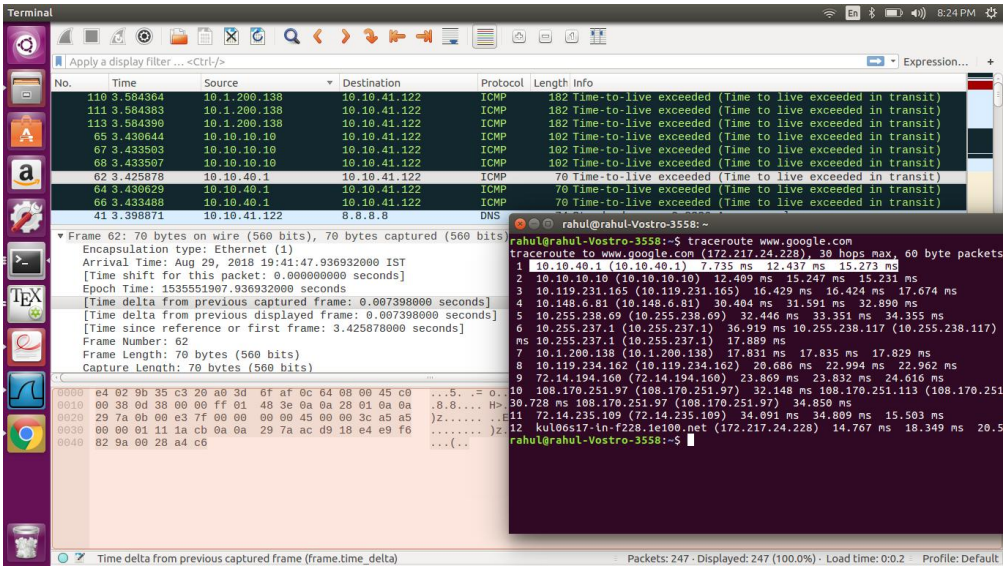

Figure 4: 10.10.40.1 first packet's timestamp

(c) In each iteration of traceroute does it use the same port number for the destination ? IF yes, reason why and if no, then also argue why does it do so.

No, it does not use the same port numbers. This is done basically to check the order in which the packets are sent . Another way to argue about the above condition is that it might be possible that the ports are being utilized by come other application time at the same time , so in such cases it will not respond to our request . And hence we require an another port number to complete our task.

4: The Fourth Problem

open above link and trace the route of google.com

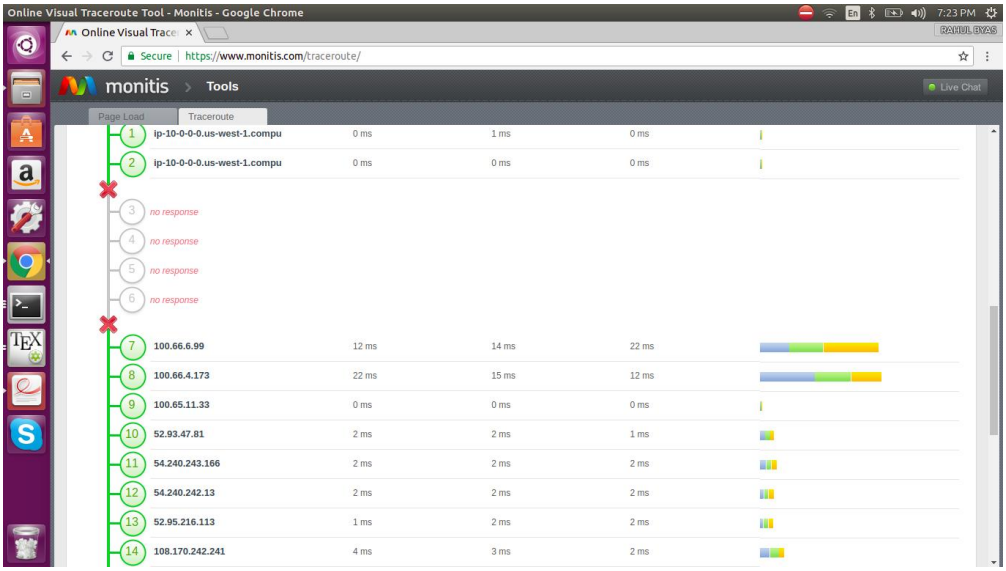

Figure 5: www.monitis.com/traceroute/

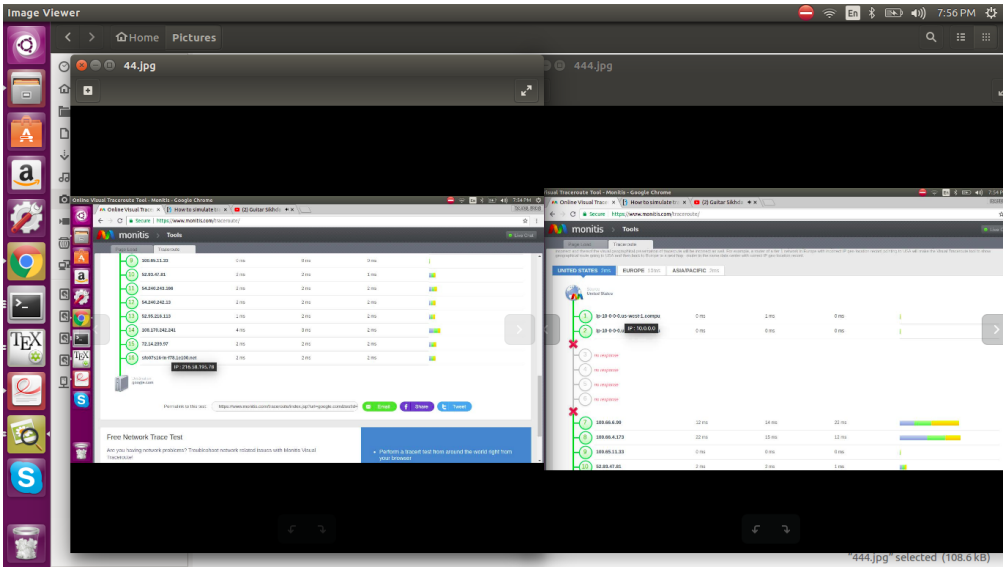

Figure 6: www.monitis.com/traceroute/

source IP:10.0.0.0 destination IP:216.58.195.78

5: The Fifth Problem

Whenever we run a traceroute command for an IP which is protected by firewall then it would not respond any ICMP message.
Instead it will show an *** which has two possibilities ,i.e. either the TTL limit has reached or a firewall has been encountered by the packet in the way to destination.
In order to confirm that it is a firewall , we need to find some IPA related to previous IPA after which *** was acheived.We can use ping sweep to find IPAs related to that IPAs.
If the IPA of the hop just after that IPA is kind of similar to the IPA then it is a firewall.

6: The Sixth Problem

In this case, the last IP appearing indicates the address of the destination.

7: The Seventh Problem

Ping is used to ensure that a host computer the user is trying to reach is actually operating or not. Ping works by sending an Internet Control Message Protocol (ICMP) Echo request to a specified interface on the network and waiting for a reply. Ping can be used for troubleshooting to test connectivity and determine response time. Ping can also be used to gather Information about the maximum frame size on the network.(help : wikipedia)
Usage of ping program :
1. Sends an ICMP message to the argument hostname and wait for the response

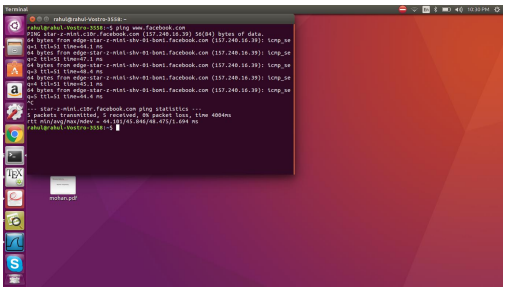

Figure 7: ping

2. Records any loss of packets

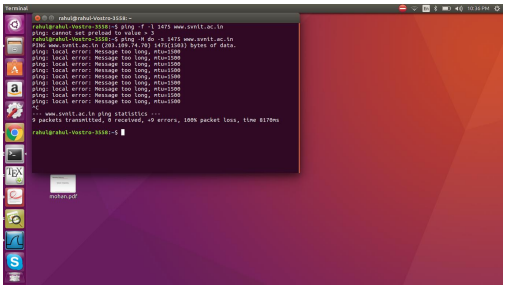

Figure 8: ping

3. The maximum frame size on the network.

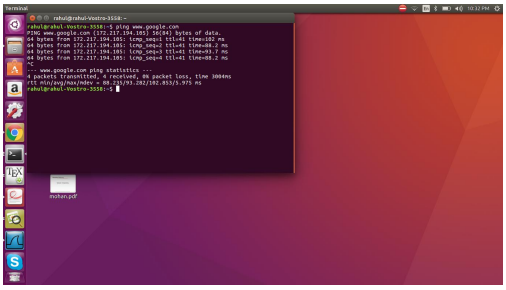

Figure 9: ping

4. Find the IPA of all the routers along the path to the host and the maximum hops to the target

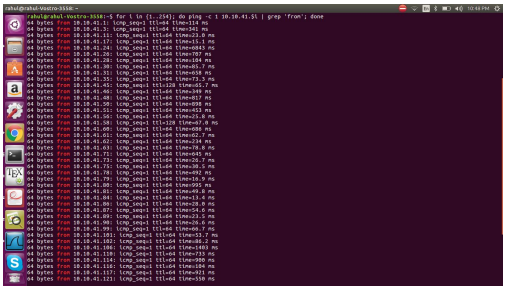

Figure 10: ping

5. Identifying the presence of a firewall
Run : ping -v -T tsandaddr www.svnit.ac.in (No firewall as you can see in the screenshot)
Run : ping -v -T tsandaddr www.du.edu (Firewall detected 100 percent packet lost.)

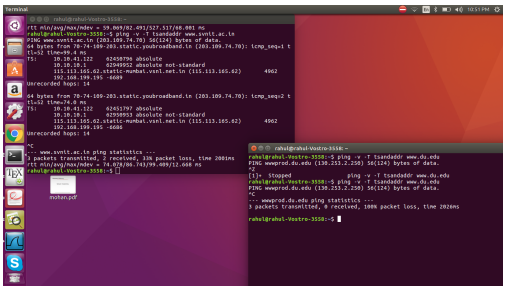

Figure 11: ping

8: The Eighth Problem

Given below is a screenshot in which we have run a small bash script:
EXPLANATION:
Over here it is just a for loop in which i is a variable iterating from 1 to 30 and that i is used for creating different IPAs inside the loop which is used by ping to check its availability . Grep command is given to follow the next IPA from previous IPA.
-c is basically number of counts it is sending to ping request
-t is for TTL of a packet.

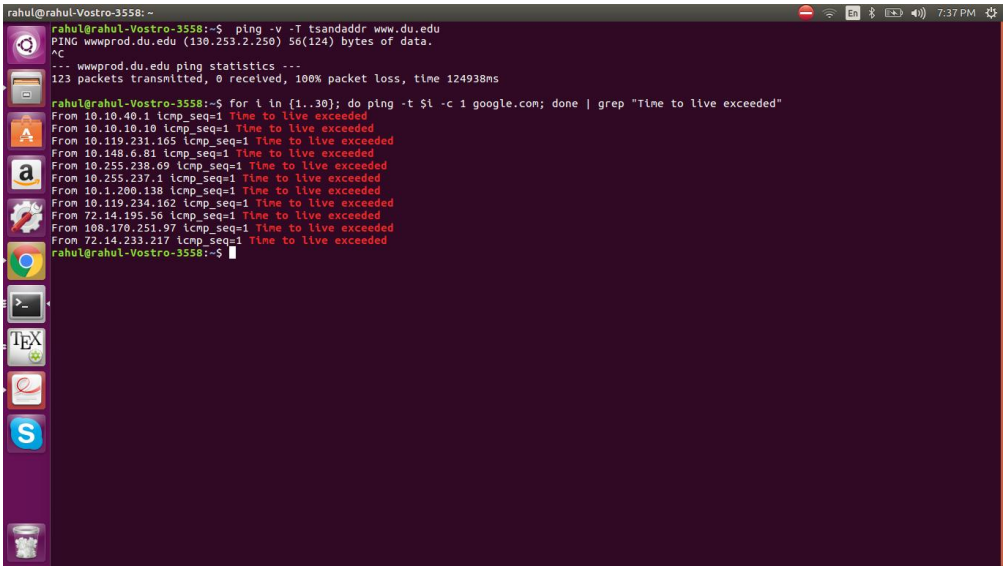

Figure 12: loop command

9: The Nineth Problem

Helping Reference : www.slashroot.in/what-ping-sweep-and-how-do-ping-sweep
Ping sweep is a way to find that various system which are around us are connected to the network or not.There are various approaches for ping sweep and one can use fping , gping , nmap and simple bash for loop for implementing this technique.

(a) Using nmap:

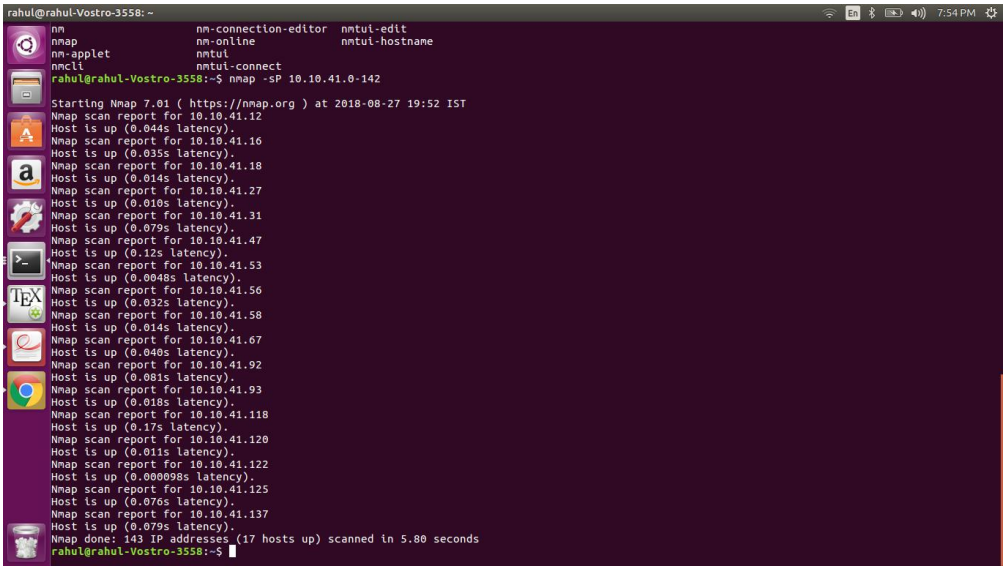

Figure 13: nmap

(b) Using fping:

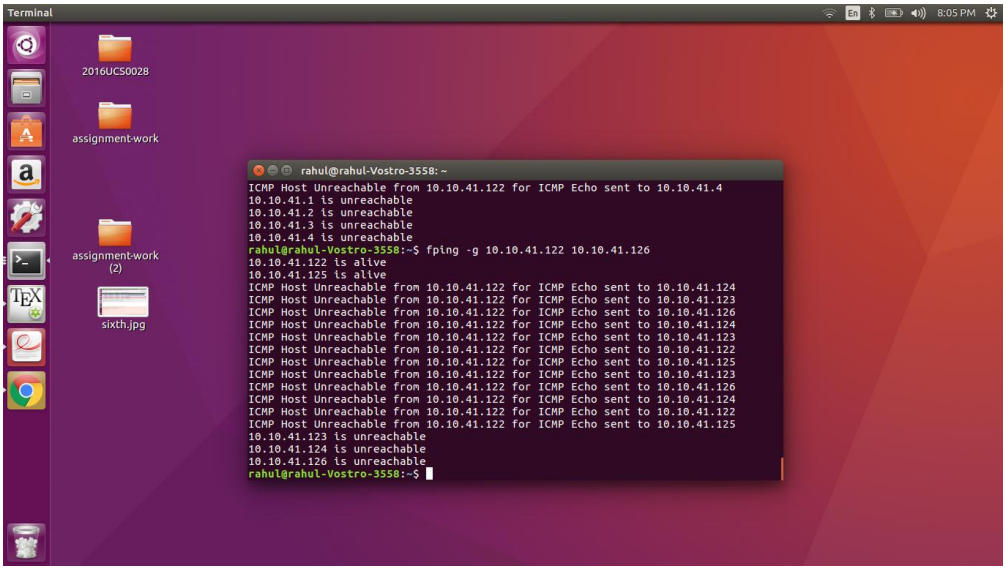

Figure 14: fping

(c) Using simple Bash for loop:

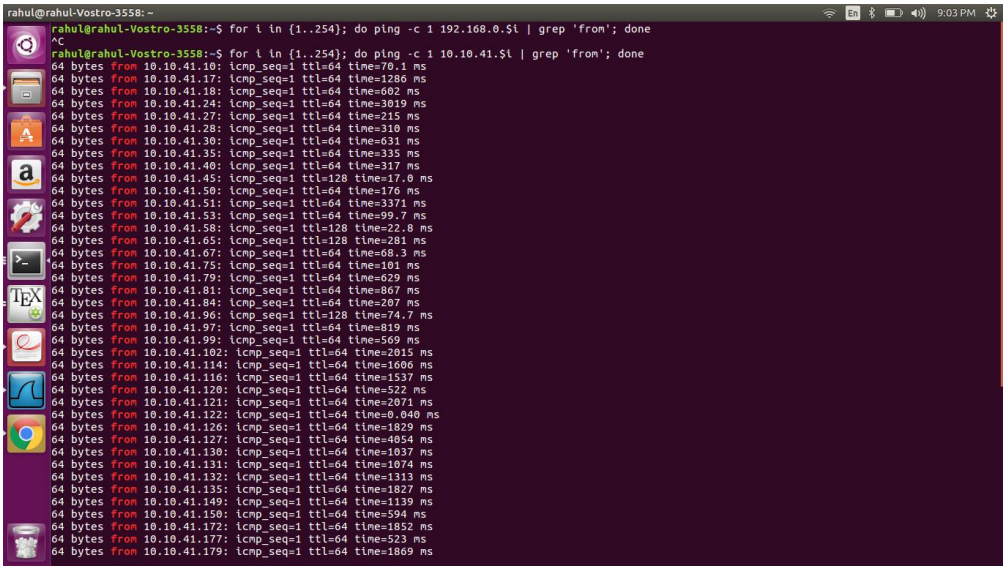

Figure 15: loop command