Sidharth Gilela
ID Number: 1428033
sgilela@ucsc.edu
CMPE 150/150L

Lab 2

HTTP Section

1) The IP address of www.example.com is 93.184.216.34 and the port number that the web server is located on is port 80
2) A request URI is a uniform request identifier of the resource that the request applies to. There are two request URI's that can be seen here. The first one is http://www.example.com/ which is to receive the html file and the second one is http://www.example.com/favicon.icol which is to receive favicol.icol
3) The HTTP version that is being used is HTTP/1.1 and we know this because whenever the get method is called, it can be seen that the HTTP/1.1 version is being used. I have a screen shot below that shows this:

No.	Time	Source	Destination	Protocol	Length	Info
311	24.3489270	128.114.62.51	93.184.216.34	HTTP	336	GET / HTTP/1.1
313	24.3606870	93.184.216.34	128.114.62.51	HTTP	1010	HTTP/1.1 200 OK (text/html)
322	24.4295960	128.114.62.51	93.184.216.34	HTTP	347	GET /favicon.ico HTTP/1.1
324	24.4413390	93.184.216.34	128.114.62.51	HTTP	1043	HTTP/1.1 404 Not Found (text/html)
327	24.4760980	128.114.62.51	93.184.216.34	HTTP	430	GET /favicon.ico HTTP/1.1
328	24.4882620	93.184.216.34	128.114.62.51	HTTP	1043	HTTP/1.1 404 Not Found (text/html)

4) I have opened up Putty and logged into my account. When doing this, I see several protocols being done. These protocols are DNS, TCP, and SSHv2. When the DNS protocol is being done, the unix.ic.ucsc.edu is being translated to an IP address. Then, afterwards the TCP protocol is doing the initial handshake while both SSHv2 and TCP are exchanging messages between the ports of 22 that belongs to ssh and port 53250 of the current computer. Here is a screenshot:

284	27.9105340	128.114.62.51	128.114.142.6	DNS	76	Standard query 0x97fe A unix.ic.ucsc.edu
285	27.9109360	128.114.142.6	128.114.62.51	DNS	358	Standard query response 0x97fe CNAME unix.lt.ucsc.edu A 128.114.104.55 A 128.114.104.5
286	27.9167350	128.114.62.51	128.114.104.55	TCP	66	53250 > ssh [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
287	27.9169880	128.114.104.55	128.114.62.51	TCP	66	ssh > 53250 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
288	27.9170220	128.114.62.51	128.114.104.55	TCP	54	53250 > ssh [ACK] Seq=1 Ack=1 Win=65536 Len=0
289	27.9285690	128.114.104.55	128.114.62.51	SSHv2	77	Server Protocol: SSH-2.0-OpenSSH_6.6.1\r
290	27.9298230	128.114.62.51	128.114.104.55	SSHv2	98	Client Protocol: SSH-1.99-3.2.9 SSH Secure Shell for Windows
291	27.9300420	128.114.104.55	128.114.62.51	TCP	60	ssh > 53250 [ACK] Seq=24 Ack=45 Win=29312 Len=0
292	27.9300570	128.114.62.51	128.114.104.55	SSHv2	390	Client: Key Exchange Init
293	27.9302850	128.114.104.55	128.114.62.51	TCP	60	ssh > 53250 [ACK] Seq=24 Ack=381 Win=30336 Len=0
294	27.9323880	128.114.104.55	128.114.62.51	TCP	1514	[TCP segment of a reassembled PDU]
295	27.9324080	128.114.104.55	128.114.62.51	SSHv2	210	Server: Key Exchange Init
296	27.9324230	128.114.62.51	128.114.104.55	TCP	54	53250 > ssh [ACK] Seq=381 Ack=1640 Win=65536 Len=0
297	27.9399710	128.114.62.51	128.114.104.55	SSHv2	214	Client: Diffie-Hellman Key Exchange Init
298	27.9408150	128.114.104.55	128.114.62.51	SSHv2	710	Server: New Keys
299	28.1408340	128.114.62.51	128.114.104.55	TCP	54	53250 > ssh [ACK] Seq=541 Ack=2296 Win=65024 Len=0
300	28.1410470	128.114.104.55	128.114.62.51	SSHv2	710	[TCP Retransmission] Server: New Keys
301	28.1410780	128.114.62.51	128.114.104.55	TCP	66	[TCP Dup ACK 299#1] 53250 > ssh [ACK] Seq=541 Ack=2296 Win=65024 Len=0 SLE=1640 SRE=229

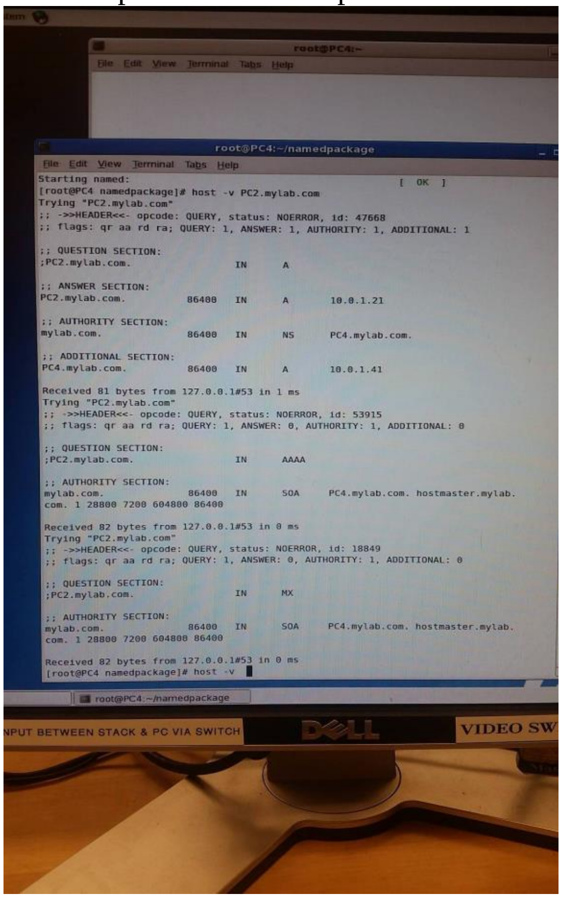
5) The status code for www.example.com is 200 OK and the status code for www.soe.ucsc.edu is 301 Moved Permanently
6) The IP address is 128.114.119.200 and the port number is 443
7) The contents of the packet sent over https is more encrypted because https is more secure than http. Thus, in comparison, when doing http with www.example.com, we were able to decipher the contents of the packet more easily than that of using https on my.ucsc.edu.

DNS Section (Part 1)

1) ON PC 4

   host –v PC2.mylab.com

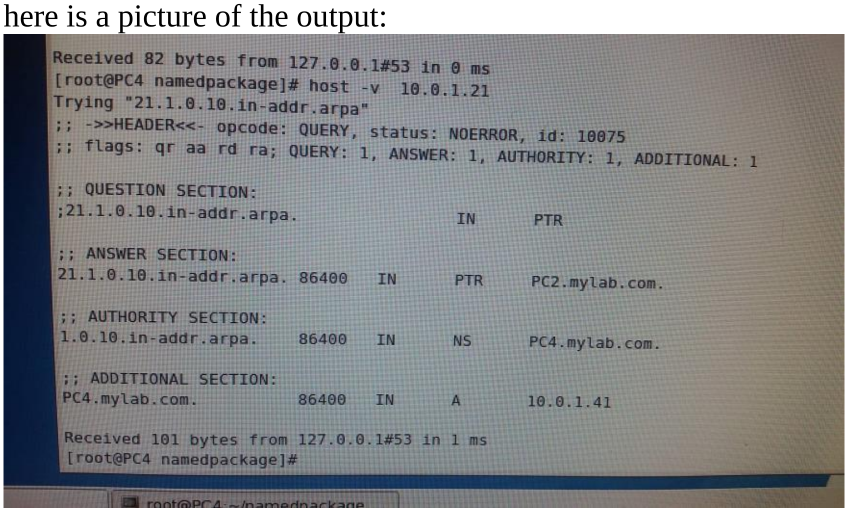
   here is a picture of the output:



- ⇨ As it can be seen in the above output this host command shows that the authoritative server is mylab.com and since there is output correctly being shown, we know that this name was resolved.

   host –v 10.0.1.21

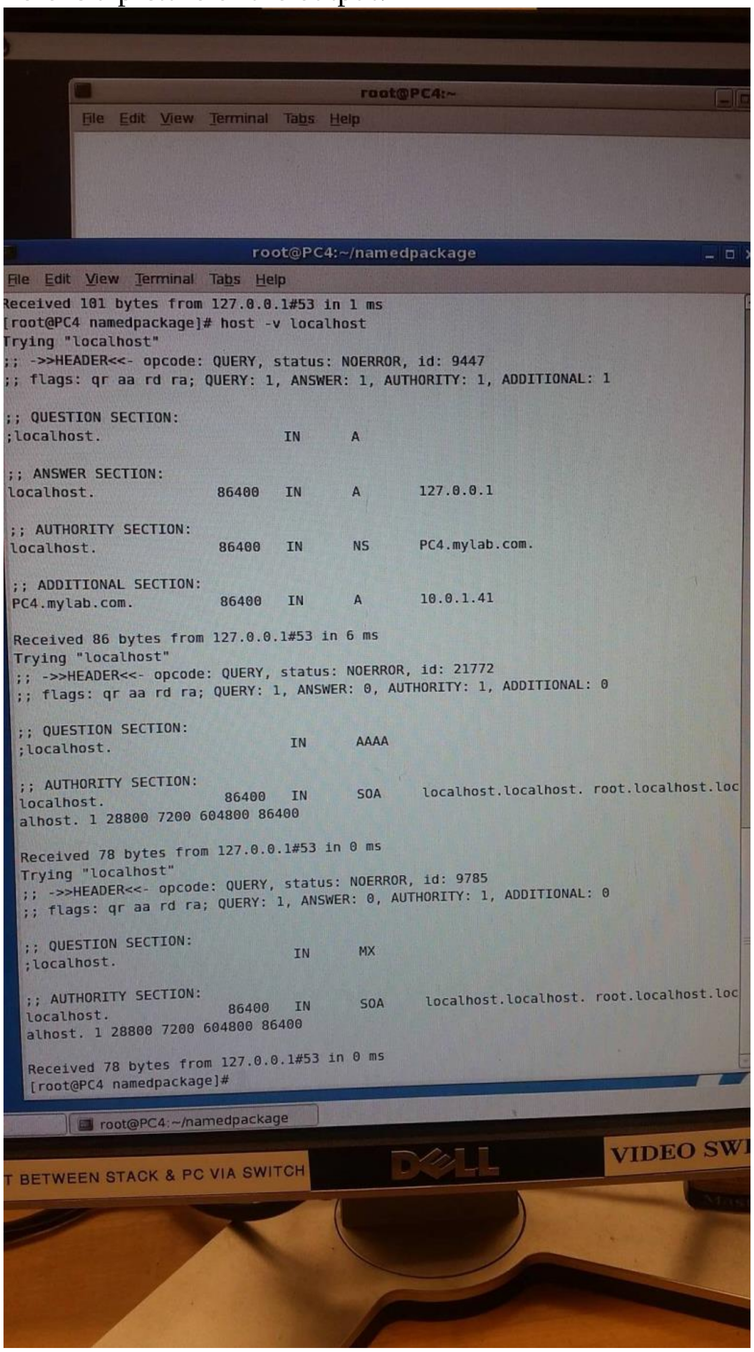
   here is a picture of the output:



⇨ As it can be seen in the above output this host command shows that there is an authority section and a name server present in that section is PC4.mylab.com and since there is output correctly being shown, we know that this name was resolved.

host –v localhost

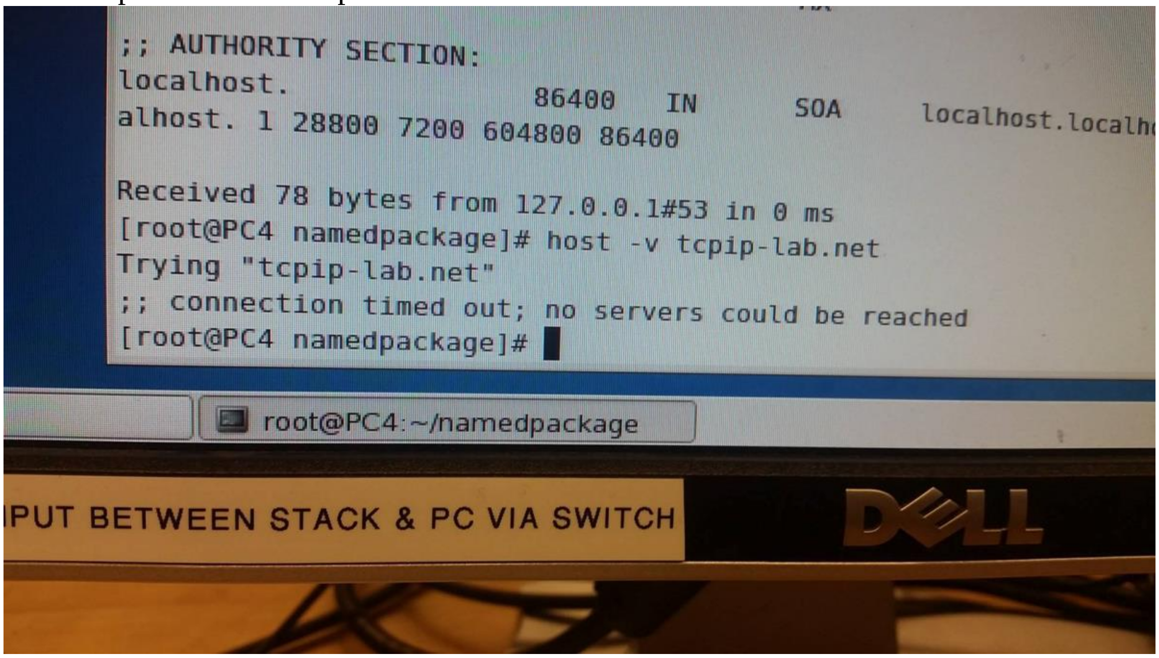
here is a picture of the output:



⇨ As it can be seen in the above output this host command shows that there is an authority section and its authority is itself because localhost is PC4 and since there is output correctly being shown, we know that this name was resolved.

host –v tcpip-lab.net

here is a picture of the output:



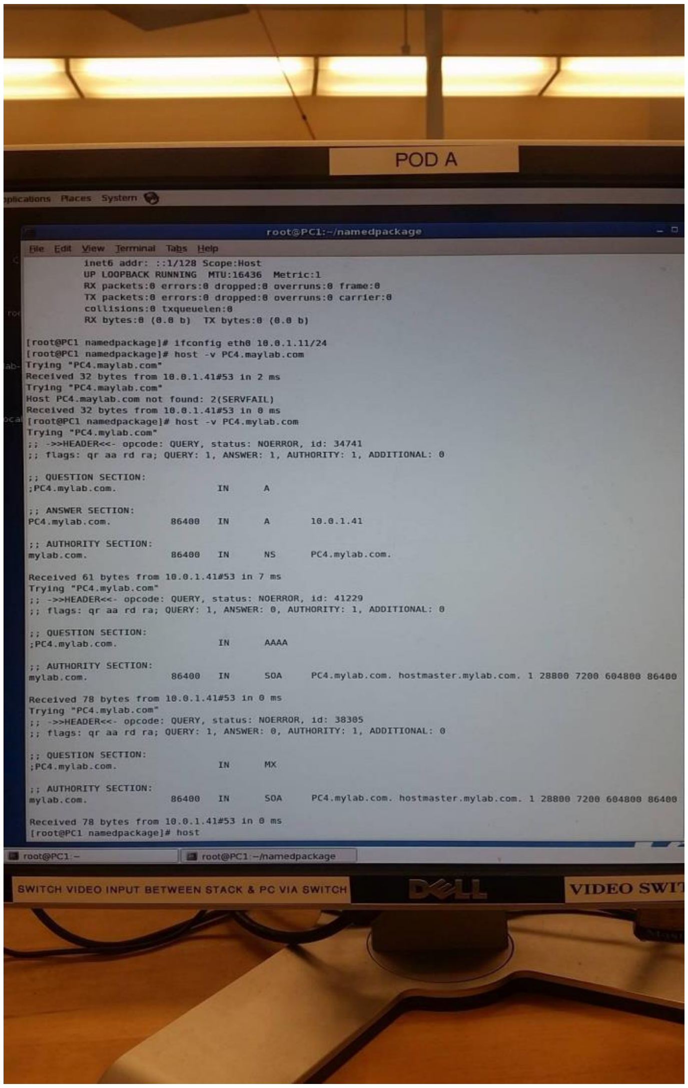
- ⇨ As it can be seen in the above output this host command shows that it cannot look for information on tcpip-lab.net. The reason this is so is because tcpip-lab.net is not part of the topology that PC4 can use to find information about this. As it can be seen in the picture, there is a “connection timed out” if after a certain amount of time, PC4 cannot locate information on this server. This also means that since no output could be generated regarding this server, this name was not resolved.

<u>ON PC 1</u>

host –v PC4.mylab.com

here is a picture of the output:

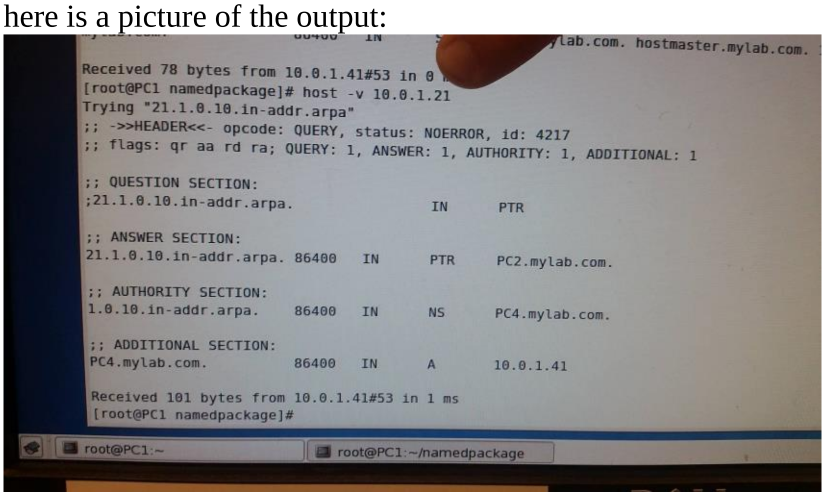
(scroll down for the pic and information)



- ⇨ As it can be seen in the above output this host command shows that the authoritative server is mylab.com and since there is output correctly being shown, we know that this name was resolved.

host –v 10.0.1.21

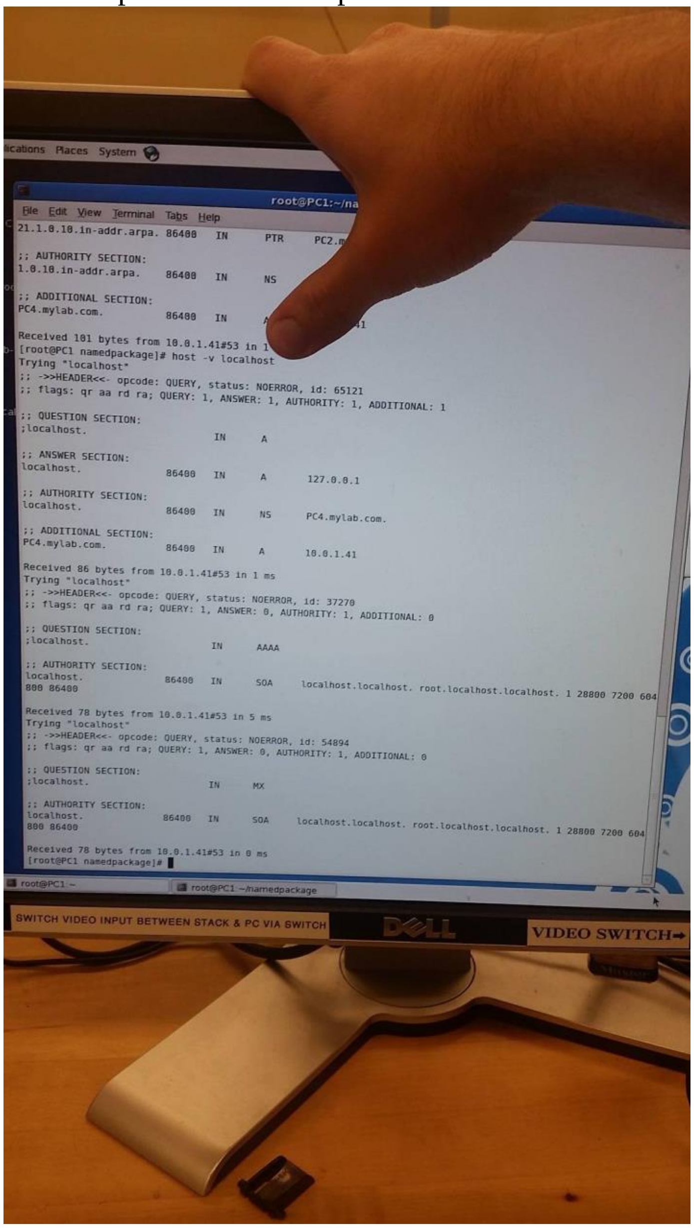
here is a picture of the output:



⇨ As it can be seen in the above output this host command shows that there is an authority section and a name server present in that section is PC2.mylab.com and since there is output correctly being shown, we know that this name was resolved.

host –v localhost

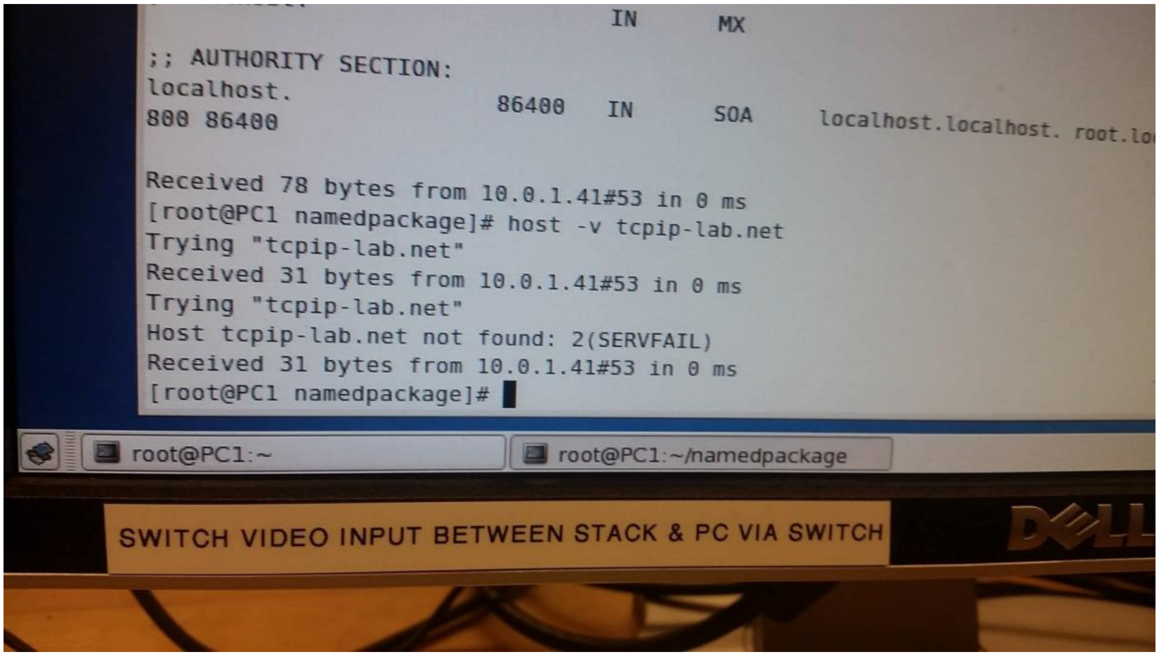
here is a picture of the output:



⇨ As it can be seen in the above output this host command shows that there is an authority section and its authority is itself because localhost is PC1 and since there is output correctly being shown, we know that this name was resolved.

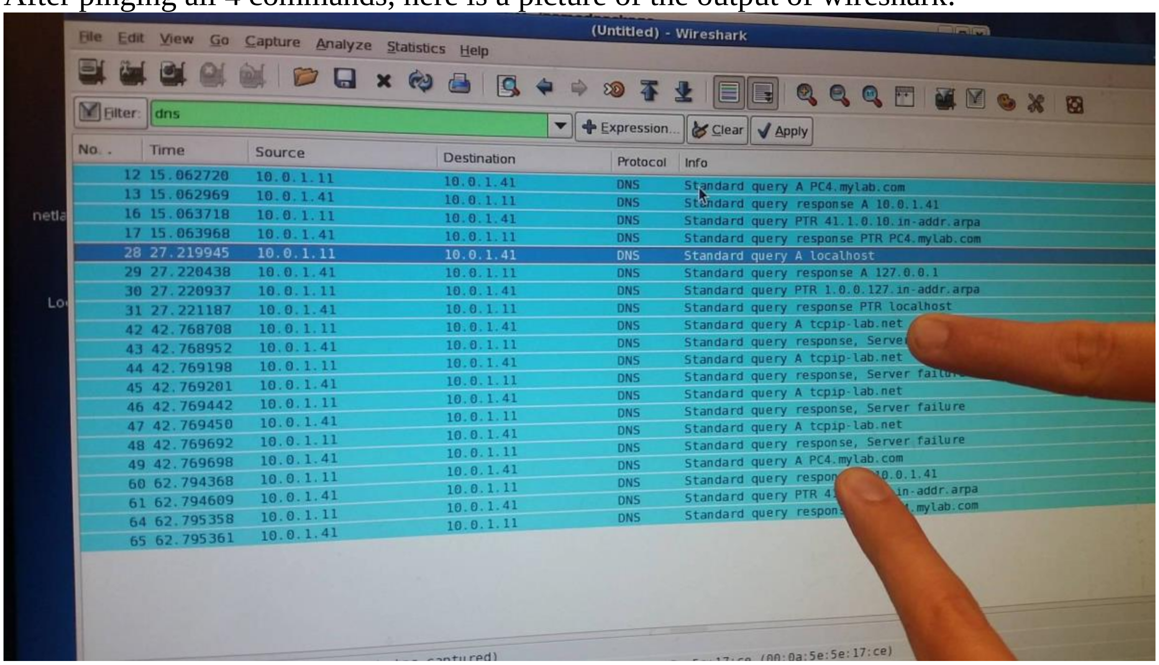
host –v tcpip-lab.net

here is a picture of the output:



=> As it can be seen in the above output this host command shows that it cannot look for information on tcpip-lab.net. The reason this is so is because tcpip-lab.net is not part of the topology that PC1 can use to find information about this. As it can be seen in the picture, it says “SERVEFAIL” meaning PC1 was not able to find out information on the server. This also means that since no output could be generated regarding this server, this name was not resolved.

2) After pinging all 4 commands, here is a picture of the output of wireshark:



In this picture, my partner has separated the four ping DNS messages in a way. The mouse is pointing to the start of the first ping, the highlighted section of wireshark in pointing to the start of the second ping, and both his fingers are pointing to the start of the two other pings. As is it can been seen in this picture, all 4 ping commands have generated DNS messeges.

3) As I have indicated in the pictures, all the queries in the trace were not resolved. When a ping was done to tcpip-lab.net, as it can be seen in the above picture, the response was “Server failure”. Thus, if a DNS query that cannot be resolved is issued, the server gets a response of “Server failure” multiple times. This indicates that the ping to this server cannot be issued.

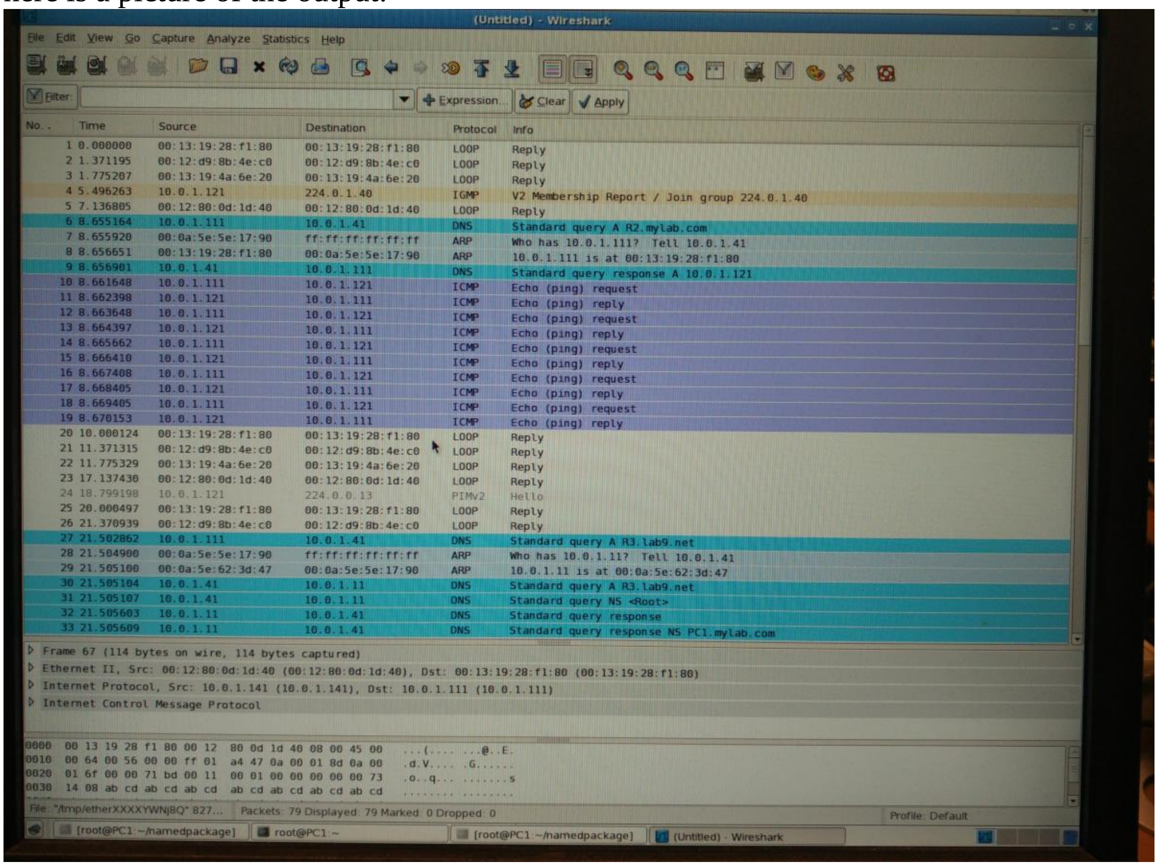
4) When I repeated the ping to PC4, PC1 issued another DNS request. The previous response was not cached. This is probably the case because maybe when we have set up the PCs’ for this lab, we set it so that PC1 could not be allowed to cache.

DNS Section (Part 2)

1) For Router 1:

ping R2.mylab.com

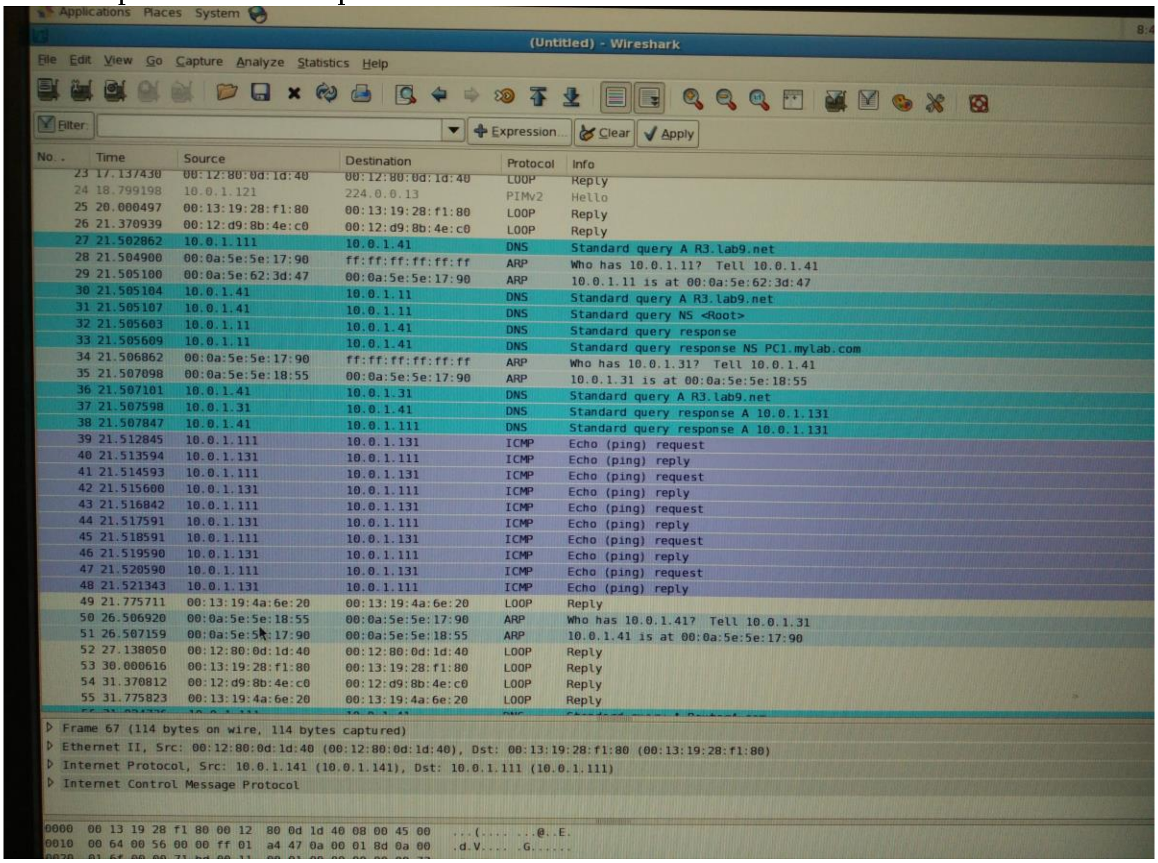
here is a picture of the output:



=> As it can be seen in the picture, router 1 first talks to the server on PC4 and PC4 responds by finding router 2 and linking up router 1 with router 2 so then can exchange messages. Afterwards, router 1 then communicates with router 2 by sending and receiving packets because of the ping command.

ping R3.lab9.net

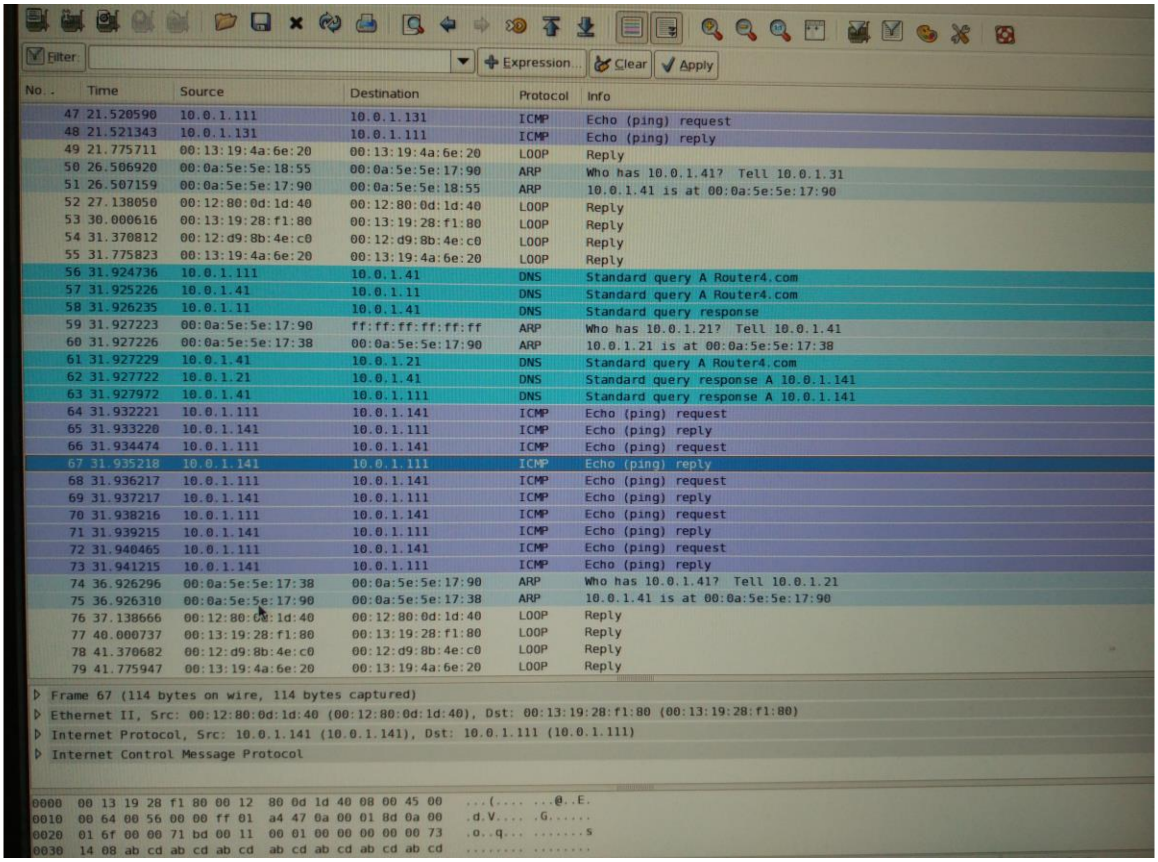
here is a picture of the output:



- As it can be seen in the picture above, router 1 first talks to the PC4 name server to see if this name server has the ip address of router 3. The PC4 server does not, so then it talks to PC1 which is the root server. The reason PC4 does not hold the ip address is because it holds the ip addresses associated with .mylab.com while what we are looking for is .lab9.net. Thus, after PC4 talks with the root server, it finds the address of the nameserver for router3 and then from there is able to locate the ip address of router 3. The connection between router 3 and router 1 is now set and both these routers can communicate with each other using the ping command.

ping Router4.com

here is a picture of the output:



=> As it can be seen in the above picture, router 1 talks to the PC4 nameserver and the PC4 nameserver talks with the root server to receive the location of Router4.com. After receiving the information that it is in the PC2 nameserver, it will communicate with PC2 nameserver to receive the ip address. After receiving the ip address, Router 1, can now talk to Router 4 with ping messages.

<u>For Router 3:</u>

ping R1.mylab.com

here is the explanation to how this DNS query was resolved:

- ⇨ Router 3 first talks to the PC 3 nameserver and this nameserver communicates with the root server on PC1. Then this root server communicates with the PC4 nameserver to attain the ip address of Router 1. After receiving this ip address, Router 3 can communicate with Router 1 by sending and receiving packets.

ping Router4.com

here is the explanation to how this DNS query was resolved:

- ⇨ Router 3 first communicates with the PC 3 nameserver and then this nameserver communicates with the root server on PC1. Then this root server looks for .com and finds it on the PC2 nameserver. Then from this PC2 nameserver, it receives the ip address of router 4 and thus, router 4 and router 3 can now send and receive packets to eachother.

ping root-server.net

here is the explanation to how this DNS query was resolved:

- ⇨ Router 3 first communicates with the PC3 nameserver, and then finds that the root-server.net domain is associated with the root server so thus, PC3 nameserver establishes a connection with the PC1 nameserver. Now, Router 3 can communicate with the root server by sending and receiving packets.

<u>For Router 4:</u>

ping R3.lab9.net

here is the explanation to how this DNS query was resolved:

- ⇨ Router 4 first communicates with the PC2 nameserver and then this nameserver communicates with the PC1 root server. After communicating with the root server, it finds that the root server knows that the PC3 nameserver has this information of the ip address of router 3. Thus, the PC3 nameserver then finds the router 3 ip address and sends this information back to Router 4 and now router 3 and router 4 can communicate with each other by sending and receiving packets.

<u>For Router 2:</u>

ping R3.lab9.net

here is the explanation to how this DNS query was resolved:

=> Since we have killed the nameserver on PC1 which is acting as the root server, if a Router needs to now connect to the root server, there will be a failure. Additionally, we have restarted the nameserver to make sure that it has an empty cache. Now, when we ping to router 3, we will get a failure because Router 2 is not able to connect with the root server that can connect to router 3.

2) The queries that have the recursion-desired flag set are the ones where Router 1 communicates with PC4 nameserver. The other communications that PC4 nameserver has with other servers do not have the recursion-desired flag set. Also when Router 3 communicates with the PC 3 nameserver there is a recursion desired flag and also when Router 4 communicates with the PC nameserver and lastly when Router 2 communicates with the PC4 nameserver. However, when any of this is happening and each of the corresponding nameservers contact other namservers, the DNS queries are iterative.

3) The authoritative servers for the .net and .com domains are PC2 and PC3. This can be seen on the table provided to us on the lab assignment.

4) I observe recursive and iterative DNS queries. As I have mentioned before, whenever Router 1 communicates with the PC4 nameserver, and vice versa, there is a recursive DNS query communication. Whenever PC4 communicates with any other servers, that communication is iterative. This can be seen in other communications as well as I have specified in Number 2.