

CS3205 - Introduction to Computer Networks

Even Sem. 2019, Dr. Manikantan Srinivasan

Assignment 4: DNS and ICMP

Individual Assignment

Due date: Apr 17, 2019, 11PM, On Moodle

Extension: 15 % penalty for each 24-hr period; Max. of 48-hrs past the original deadline

April 7, 2019

The purpose of this study/assignment is to understand the behavior of : 1) Domain Name Service and 2) Internet Control Message Protocol. This assignment is derived from the Supplemental Wireshark lab projects designed for the Book Computer Networking: A Top-Down Approach, 6th ed., by Authors J.F. Kurose and K.W. Ross.

1 Domain Name Service (DNS)

1.1 Description

As described in Section 2.5 of the text book, the Domain Name System (DNS) translates hostnames to IP addresses, fulfilling a critical role in the Internet infrastructure. In this assignment, we'll take a closer look at the client side of DNS. Recall that the client's role in the DNS is relatively simple – a client sends a query to its local DNS server, and receives a response back. As shown in Figures 2.21 and 2.22 in the textbook, much can go on “under the covers,” invisible to the DNS clients, as the hierarchical DNS servers communicate with each other to either recursively or iteratively resolve the client's DNS query. From the DNS client's standpoint, however, the protocol is quite simple – a query is formulated to the local DNS server and a response is received from that server.

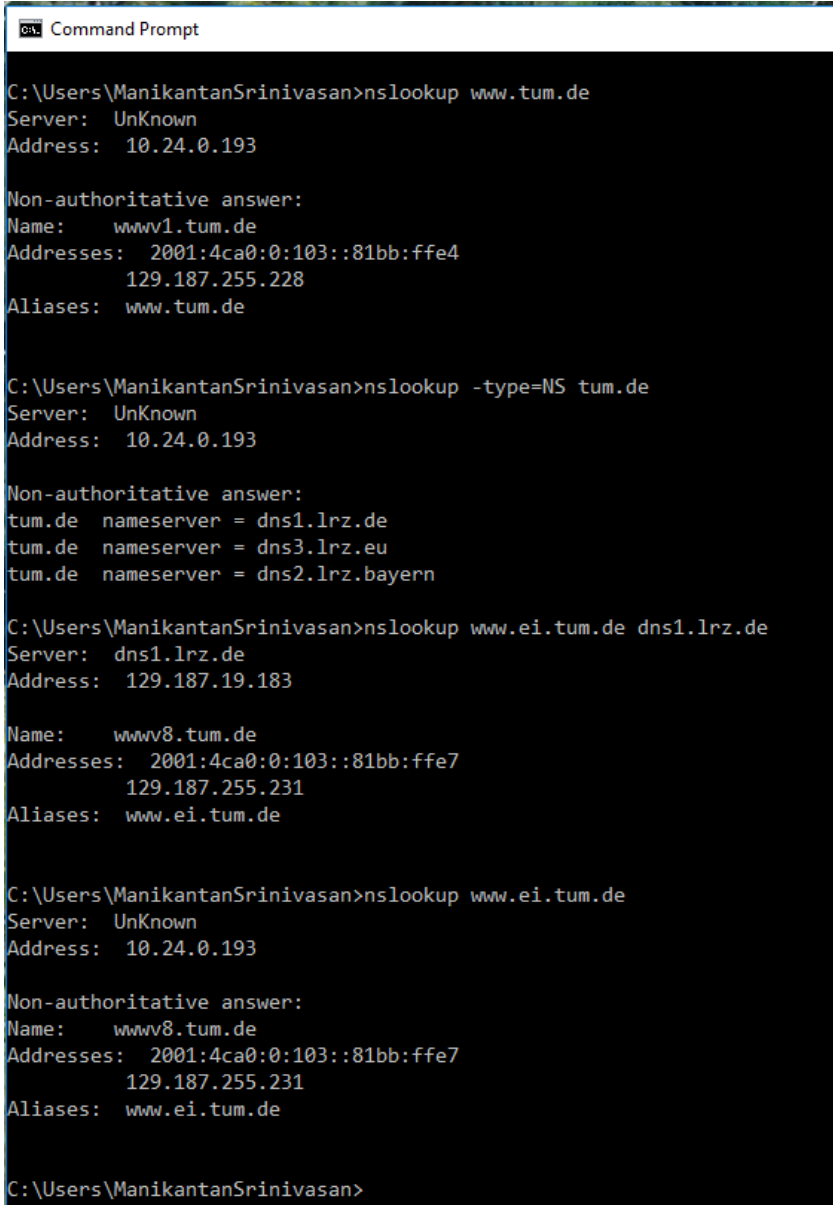
Before beginning this assignment, you'll probably want to review DNS by reading Section 2.5 of the text. In particular, you may want to review the material on local DNS servers, DNS caching, DNS records and messages, and the TYPE field in the DNS record.

1.1.1 nslookup

In this assignment, we'll make extensive use of the nslookup tool, which is available in most Linux/Unix and Microsoft platforms today. To run nslookup in Linux/Unix, you just type the nslookup command on the command line. To run it in Windows, open the Command Prompt and run nslookup on the command line.

In its most basic operation, nslookup tool allows the host running the tool to query any specified DNS server for a DNS record. The queried DNS server can be a root DNS server, a top-level-domain DNS server, an authoritative DNS server, or an intermediate DNS server (see the textbook for definitions of these terms).

To accomplish this task, nslookup sends a DNS query to the specified DNS server, receives a DNS reply from that same DNS server, and displays the result.



```
Command Prompt

C:\Users\ManikantanSrinivasan>nslookup www.tum.de
Server: UnKnown
Address: 10.24.0.193

Non-authoritative answer:
Name: www1.tum.de
Addresses: 2001:4ca0:0:103::81bb:ffe4
          129.187.255.228
Aliases: www.tum.de

C:\Users\ManikantanSrinivasan>nslookup -type=NS tum.de
Server: UnKnown
Address: 10.24.0.193

Non-authoritative answer:
tum.de nameserver = dns1.lrz.de
tum.de nameserver = dns3.lrz.eu
tum.de nameserver = dns2.lrz.bayern

C:\Users\ManikantanSrinivasan>nslookup www.ei.tum.de dns1.lrz.de
Server: dns1.lrz.de
Address: 129.187.19.183

Name: www8.tum.de
Addresses: 2001:4ca0:0:103::81bb:ffe7
          129.187.255.231
Aliases: www.ei.tum.de

C:\Users\ManikantanSrinivasan>nslookup www.ei.tum.de
Server: UnKnown
Address: 10.24.0.193

Non-authoritative answer:
Name: www8.tum.de
Addresses: 2001:4ca0:0:103::81bb:ffe7
          129.187.255.231
Aliases: www.ei.tum.de

C:\Users\ManikantanSrinivasan>
```

Figure 1: Sample NS Look up for Technical University of Munich.

The above Figure 1 shows the results of three independent nslookup commands (displayed in the Windows Command Prompt). In this example, the client host is located on the campus of IIT-Madras - HPCN Lab, where the default local DNS server is 10.24.0.193. When running nslookup, if no DNS server is specified, then nslookup sends the query to the default DNS server, which in this case is 10.24.0.193. Consider the first command:

nslookup www.tum.de

In words, this command is saying “please send me the IP address for the host www.tum.de”. As shown in the screenshot, the response from this command provides two pieces of information: (1) the name and IP address of the DNS server that provides the answer; and (2) the answer itself, which is the host name and

IP address of `www.tum.de`. Although the response came from the local DNS server at IIT-Madras, it is quite possible that this local DNS server iteratively contacted several other DNS servers to get the answer, as described in Section 2.5 of the textbook.

Now consider the second command:

nslookup -type=NS tum.de

In this example, we have provided the option “-type=NS” and the domain “tum.de”. This causes nslookup to send a query for a type-NS record to the default local DNS server. In words, the query is saying, “please send me the host names of the authoritative DNS for tum.de”. (When the -type option is not used, nslookup uses the default, which is to query for type A records.) The answer, displayed in the above screenshot, first indicates the DNS server that is providing the answer (which is the default local DNS server) along with three TUM nameservers. Each of these servers is indeed an authoritative DNS server for the hosts on the TUM campus. However, nslookup also indicates that the answer is “non-authoritative,” meaning that this answer came from the cache of some server rather than from an authoritative TUM DNS server.

Now consider the third command:

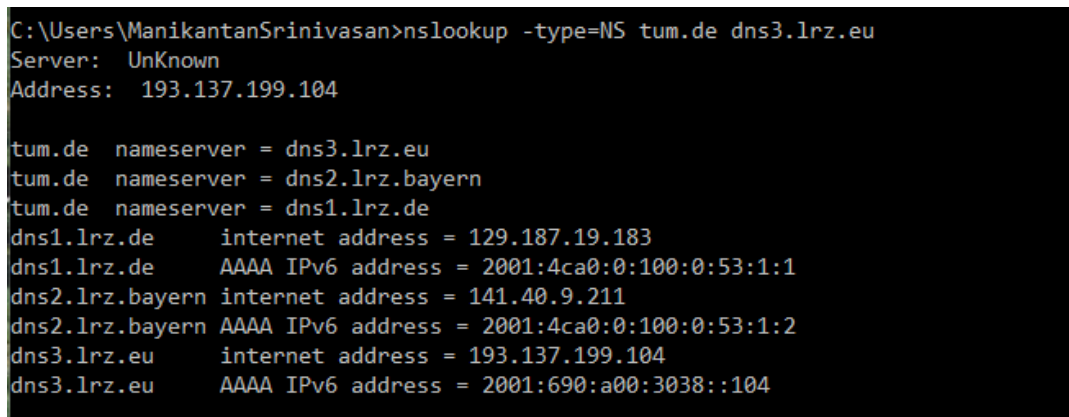
nslookup www.ei.tum.de dns1.lrz.de

In this example, we indicate that we want the query sent to the DNS server `dns1.lrz.de` rather than to the default DNS server (10.24.0.193). Thus, the query and reply transaction takes place directly between our querying host and `dns1.lrz.de`. In this example, the DNS server `dns1.lrz.de` provides the IP address of the host `www.ei.tum.de`, which is a web server at the Electrical and Computer Engineering department of Technical University of Munich.

Now consider the last command: (Figure 2)

nslookup -type=NS tum.de dns3.lrz.eu

In this example the DNS server `dns1.lrz.de` is contacted and the both IPv4 and IPv6 address associated with the authoritative DNS servers are given.



```
C:\Users\ManikantanSrinivasan>nslookup -type=NS tum.de dns3.lrz.eu
Server: UnKnown
Address: 193.137.199.104

tum.de nameserver = dns3.lrz.eu
tum.de nameserver = dns2.lrz.bayern
tum.de nameserver = dns1.lrz.de
dns1.lrz.de internet address = 129.187.19.183
dns1.lrz.de AAAA IPv6 address = 2001:4ca0:0:100:0:53:1:1
dns2.lrz.bayern internet address = 141.40.9.211
dns2.lrz.bayern AAAA IPv6 address = 2001:4ca0:0:100:0:53:1:2
dns3.lrz.eu internet address = 193.137.199.104
dns3.lrz.eu AAAA IPv6 address = 2001:690:a00:3038::104
```

Figure 2: Sample NS Look up for Technical University of Munich-2, Authoritative DNS servers for TUM.

Now that we have gone through a few illustrative examples, you are perhaps wondering about the general syntax of nslookup commands. The syntax is:

nslookup -option1 -option2 host-to-find dns-server

In general, nslookup can be run with zero, one, two or more options. And as we have seen in the above examples, the dns-server is optional as well; if it is not supplied, the query is sent to the default local DNS server.

Now that we have provided an overview of nslookup, it is time for you to test drive it yourself. Do the

following (and write down the results):

1. Run nslookup to obtain the IP address of the web server of your favorite technical university in Asia, in Europe and in America. What is the IP address of that server?
2. Run nslookup to determine the authoritative DNS servers for the same technical universities.
3. Run nslookup so that one of the DNS servers obtained in Question 2 is queried for the web and mail server for the Computer Science Department your favorite technical universities. What are the IP addresses?
4. What are the IP addresses of the authoritative DNS servers at these universities?

1.1.2 Tracing DNS with Wireshark

Let's first capture the DNS packets that are generated by ordinary Web-surfing activity.

- Empty the DNS cache in your host. (Please find by internet search the option to clean the DNS cache)
- Open your browser and empty your browser cache.
- Open Wireshark and enter "ip.addr == IP address of your system" into the filter, where you obtain "IP address of your system" using suitable command such as ipconfig/ifconfig. This filter removes all packets that neither originate nor are destined to your host.
- Start packet capture in Wireshark.
- With your browser, visit the Web page: <http://www.ietf.org>
- Stop packet capture.

Answer the following questions:

5. Locate the DNS query and response messages. Are then sent over UDP or TCP?
6. What is the destination port for the DNS query message? What is the source port of DNS response message?
7. To what IP address is the DNS query message sent? Use ipconfig to determine the IP address of your local DNS server. Are these two IP addresses the same?
8. Examine the DNS query message. What "Type" of DNS query is it? Does the query message contain any "answers"?
9. Examine the DNS response message. How many "answers" are provided? What do each of these answers contain?
10. Consider the subsequent TCP SYN packet sent by your host. Does the destination IP address of the SYN packet correspond to any of the IP addresses provided in the DNS response message?
11. This web page contains images. Before retrieving each image, does your host issue new DNS queries?

Now let's play with nslookup .

- Start packet capture.
- Do an nslookup on one your favorite web site (here for example we use www.mit.edu).

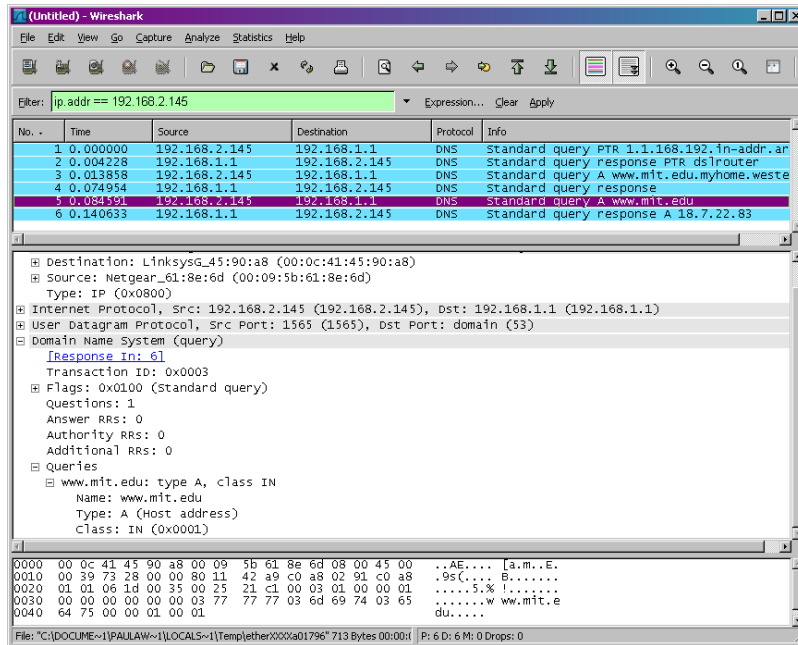


Figure 3: Output with filter - DNS for mit.edu.

- Stop packet capture.

You should get a trace that looks something like the following:

We see from the above screenshot that nslookup actually sent three DNS queries and received three DNS responses. For the purpose of this assignment, in answering the following questions, ignore the first two sets of queries/responses, as they are specific to nslookup and are not normally generated by standard Internet applications. You should instead focus on the last query and response messages.

- What is the destination port for the DNS query message? What is the source port of DNS response message?
- To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server?
- Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?
- Examine the DNS response message. How many “answers” are provided? What do each of these answers contain?
- Provide a screenshot.

Now repeat the previous experiment, but instead issue the command:

For example: (you can do it for the same department, same university as done initially)

nslookup -type=NS tum.de

Answer the following questions :

- To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server?

17. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?
18. Examine the DNS response message. What MIT nameservers does the response message provide? Does this response message also provide the IP addresses of the MIT namesers?
19. Provide a screenshot.

Now repeat the previous experiment, but instead issue the command:

For example: (you can do it for the same department, same university as done initially)

nslookup www.ei.tum.de dns1.lrz.de

Answer the following questions :

20. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server? If not, what does the IP address correspond to?
21. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?
22. Examine the DNS response message. How many “answers” are provided? What does each of these answers contain?
23. Provide a screenshot.

Submit wireshark captures for all the scenarios.

You may refer the following URLs for additional information/understanding.

1. <https://ns1.com/resources/dns-types-records-servers-and-queries>
2. <https://simplifiedns.com/help/dns-record-types>
3. <https://www.farsightsecurity.com/txt-record/2017/12/01/stsauver-dnsrecords/>

1.2 What to Submit

The sub-directory “DNS” should contain the following files:

- Appropriate Wireshark capture files (4 in number), and screen shots
- Text file (doc) that answers the given set of questions

2 Internet Control Message Protocol - ICMP

In this part of assignment, we’ll explore several aspects of the ICMP protocol:

- ICMP messages generating by the Ping program;
- ICMP messages generated by the Traceroute program;
- the format and contents of an ICMP message.

Before working this part of the assignment, you’re encouraged to review the ICMP material in section 4.4.3 of the text book. This assignment can be performed easily in any enviornment - Windows, Linux, MacOS. Example screen shots made here are done using MacOS.

2.1 ICMP and Ping

Let's begin our ICMP adventure by capturing the packets generated by the Ping program. You may recall that the Ping program is a simple tool that allows anyone (for example, a network administrator) to verify if a host is live or not. The Ping program in the source host sends a packet to the target IP address; if the target is live, the Ping program in the target host responds by sending a packet back to the source host. As you might have guessed (given that this assignment is about ICMP), both of these Ping packets are ICMP packets.

Do the following :

- Use suitable console application in your system. i.e., terminal in Linux, Command prompt in Windows, terminal in MacOS.
- Start up the Wireshark packet sniffer, and begin Wireshark packet capture.
- You can type "ping hostname" or "ping *ip address*".
- To specify the number of ping request/responses add the option "-n *count*".
- To specify the size of ping request/response add the option "-s *size*".
- Ping one of your favourite website (i.e., use the name used in the earlier part of the assignment)
- By specifying "-n *count*" the ping program terminates after the specified number of packets are sent and received.
- In Linux, by default the ping continues if count is not specified. In windows by default the ping stops after 3 counts. So suitably specify the needful optional values.
- Ideally choose a name that is quite far from your location.
- When the Ping program terminates, stop the packet capture in Wireshark.

At the end of the experiment, your console Window should look something like Figure 4. In this example, the source ping program is in Chennai and the destination Ping program is in Munich (TUM). From this window we see that the source ping program sent 10 query packets and received 10 responses. Note also that for each response, the source calculates the round-trip time (RTT), which for the 10 packets is on average 387.656 msec.

Figure 5 provides a screenshot of the Wireshark output, after "icmp" has been entered into the filter display window. Note that the packet listing shows 20 packets: the 10 Ping queries sent by the source and the 10 Ping responses received by the source. Also note that the source's IP address is a private address (behind a NAT) of the form 192.168/12; the destination's IP address is that of the Web server at TUM. Now let's zoom in on the first packet (sent by the client); in the figure below, the packet contents area provides information about this packet. We see that the IP datagram within this packet has protocol number 01, which is the protocol number for ICMP. This means that the payload of the IP datagram is an ICMP packet.

Figure 6 focuses on the same ICMP but has expanded the ICMP protocol information in the packet contents window. Observe that this ICMP packet is of Type 8 and Code 0 - a so-called ICMP "echo request" packet. (See Figure 4.23 of text.) Also note that this ICMP packet contains a checksum, an identifier, and a sequence number.

What to Submit:

In a subdirectory "ICMP_Ping", You should submit a screen shot of the Command Prompt window similar to Figure 4 above. Whenever possible, when answering a question below, you should submit a screen shot of the packet(s) within the trace that you used to answer the question asked. You should take a screenshot/snapshot of the packet(s) within the trace that you used to answer the question asked. Provide suitable response along

```

~ — -bash
n:~ Admin$ ping www.tum.de -c 10
PING wwwv1.tum.de (129.187.255.228): 56 data bytes
64 bytes from 129.187.255.228: icmp_seq=0 ttl=242 time=368.183 ms
64 bytes from 129.187.255.228: icmp_seq=1 ttl=242 time=391.659 ms
64 bytes from 129.187.255.228: icmp_seq=2 ttl=242 time=414.748 ms
64 bytes from 129.187.255.228: icmp_seq=3 ttl=242 time=433.963 ms
64 bytes from 129.187.255.228: icmp_seq=4 ttl=242 time=299.295 ms
64 bytes from 129.187.255.228: icmp_seq=5 ttl=242 time=474.341 ms
64 bytes from 129.187.255.228: icmp_seq=6 ttl=242 time=493.417 ms
64 bytes from 129.187.255.228: icmp_seq=7 ttl=242 time=312.135 ms
64 bytes from 129.187.255.228: icmp_seq=8 ttl=242 time=335.048 ms
64 bytes from 129.187.255.228: icmp_seq=9 ttl=242 time=353.770 ms

--- wwwv1.tum.de ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 299.295/387.656/493.417/62.770 ms
Manikantan:~ Admin$

```

Figure 4: Console output after Ping command execution

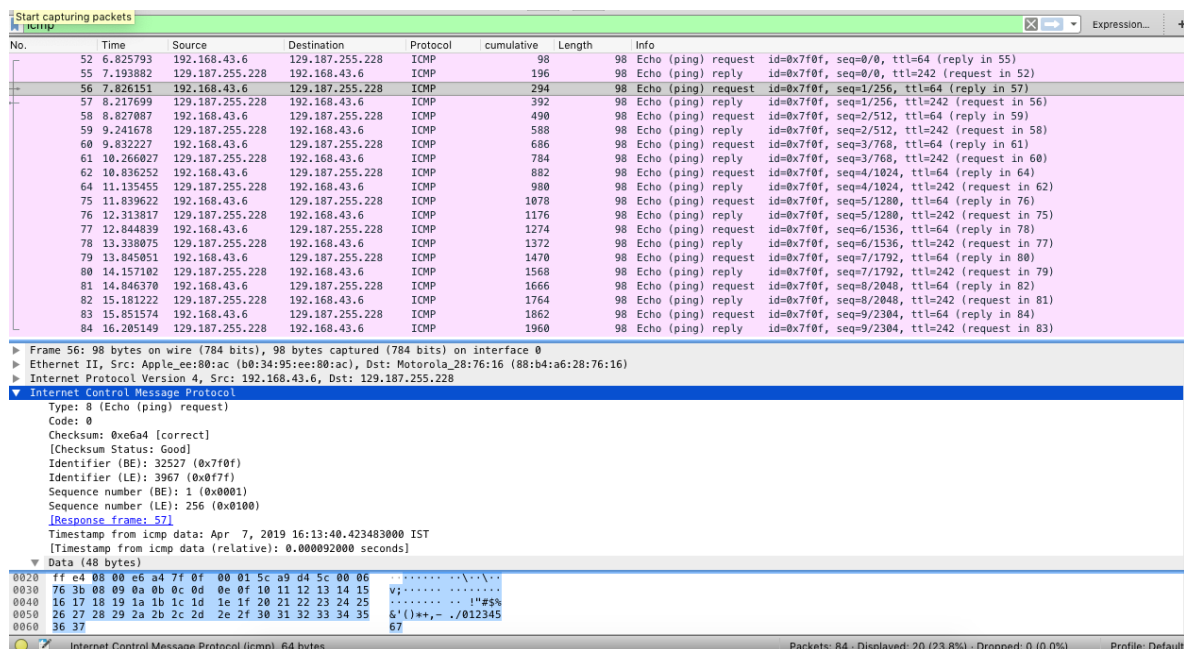


Figure 5: Wireshark capture of the Ping execution

with the snapshot (picture) to explain your answer. Select the needful packet contents and capture suitable screen shots. The Wireshark file submitted must have the packets from which the screen shot has been enabled. The above figures are examples of the screen shots expected.

You should answer the following questions: (A word / PDF document)

1. What is the IP address of your host? What is the IP address of the destination host?

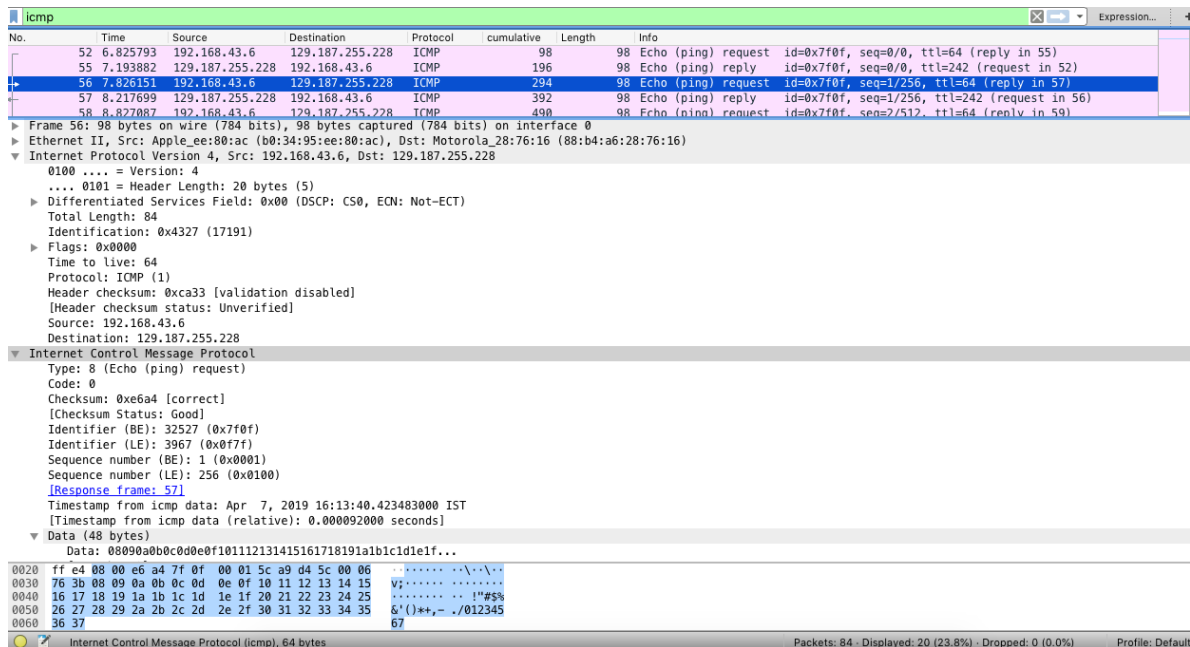


Figure 6: Wireshark capture of ping packet with IP and ICMP packet expanded

2. Why is it that an ICMP packet does not have source and destination port numbers?
3. Examine one of the ping request packets sent by your host. What are the ICMP type and code numbers? What other fields does this ICMP packet have? How many bytes are the checksum, sequence number and identifier fields?
4. Examine the corresponding ping reply packet. What are the ICMP type and code numbers? What other fields does this ICMP packet have? How many bytes are the checksum, sequence number and identifier fields?

2.2 ICMP and Traceroute

Let's now continue our ICMP adventure by capturing the packets generated by the Traceroute program. You may recall that the Traceroute program can be used to figure out the path a packet takes from source to destination. Traceroute is discussed in Section 1.4 and in Section 4.4 of the text book.

Traceroute is implemented in different ways in Unix/Linux/macOS and in Windows. In Unix/Linux, the source sends a series of UDP packets to the target destination using an unlikely destination port number; in Windows, the source sends a series of ICMP packets to the target destination. For both operating systems, the program sends the first packet with TTL=1, the second packet with TTL=2, and so on. Recall that a router will decrement a packet's TTL value as the packet passes through the router. When a packet arrives at a router with TTL=1, the router sends an ICMP error packet back to the source.

Figure shows the execution of "traceroute" in MacOS. The application generates UDP packets. Figure shows the execution of "tracert" in Windows.

You will have to do the assignment with any one implementation. If you are interested, and if you have opportunity test the behavior in other systems.

Do the following :

- Open suitable console as supported in your system.

```
Admin — -bash — 93x28
~ — -bash

Manikantan:~ Admin$ traceroute www.tum.de
traceroute to wwwv1.tum.de (129.187.255.228), 64 hops max, 52 byte packets
 1  192.168.43.1 (192.168.43.1)  6.010 ms  2.039 ms  2.945 ms
 2  * * *
 3  10.50.122.73 (10.50.122.73)  38.375 ms  23.019 ms  46.962 ms
 4  10.0.66.181 (10.0.66.181)  35.710 ms *  43.488 ms
 5  125.17.180.201 (125.17.180.201)  38.019 ms  29.569 ms  29.673 ms
 6  182.79.224.181 (182.79.224.181)  73.176 ms
    182.79.152.162 (182.79.152.162)  86.312 ms
    182.79.237.16 (182.79.237.16)  78.632 ms
 7  63-218-107-193.static.pccwglobal.net (63.218.107.193)  68.164 ms  69.075 ms  70.074 ms
 8  tenge0-0-0-14.br01.frf08.pccwbtn.net (63.223.13.94)  291.595 ms  335.930 ms
    tenge0-1-0-12.br01.frf08.pccwbtn.net (63.223.13.138)  482.794 ms
 9  cr-fra2-be1.x-win.dfn.de (80.81.192.222)  304.333 ms  513.347 ms  409.684 ms
10  cr-gar1-be6.x-win.dfn.de (188.1.145.230)  409.221 ms  403.354 ms  320.949 ms
11  kr-gar188-0.x-win.dfn.de (188.1.37.90)  286.344 ms  287.316 ms  338.568 ms
12  vl-3001.cvr2-2wr.lrz.de (129.187.0.168)  293.343 ms  308.050 ms  281.269 ms
13  wwwv1.tum.de (129.187.255.228)  290.045 ms  298.263 ms  295.886 ms
14  * f5slb4.lrz.de (129.187.255.244)  1423.389 ms !H *
15  f5slb4.lrz.de (129.187.255.244)  1297.177 ms !H *  1392.781 ms !H
Manikantan:~ Admin$
```

Figure 7: Traceroute to www.tum.de from a MacOS system

```
C:\Users\ManikantanSrinivasan>tracert www.tum.de
'tracert' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\ManikantanSrinivasan>tracert www.tum.de

Tracing route to wwwv1.tum.de [129.187.255.228]
over a maximum of 30 hops:

  0  <1 ms  <1 ms  <1 ms  10.21.239.254
  1  <1 ms  <1 ms  <1 ms  10.25.0.14
  2  1 ms  1 ms  1 ms  10.119.232.138
  3  1 ms  1 ms  <1 ms  10.119.232.137
  4  25 ms  25 ms  25 ms  10.163.255.201
  5  25 ms  25 ms  25 ms  10.255.232.217
  6  25 ms  25 ms  25 ms  180.149.48.18
  7  143 ms  143 ms  143 ms  180.149.48.6
  8  143 ms  143 ms  143 ms  180.149.48.20
  9  143 ms  143 ms  143 ms  180.149.48.10
10  150 ms  150 ms  150 ms  ae0.mx1.fra.de.geant.net [62.40.98.128]
11  150 ms  150 ms  150 ms  cr-fra1.x-win.dfn.de [62.40.124.218]
12  159 ms  159 ms  158 ms  cr-gar1-be6.x-win.dfn.de [188.1.145.230]
13  159 ms  159 ms  158 ms  kr-gar188-0.x-win.dfn.de [188.1.37.90]
14  158 ms  158 ms  158 ms  vl-3001.cvr2-2wr.lrz.de [129.187.0.168]
15  158 ms  158 ms  158 ms  wwwv1.tum.de [129.187.255.228]

Trace complete.

C:\Users\ManikantanSrinivasan>
```

Figure 8: Traceroute to www.tum.de from a Windows system

- Start up the Wireshark packet sniffer, and begin Wireshark packet capture.
- Execute the traceroute functionality supported in your system i.e. either “tracert” or “traceroute”. The example show is the route traced to www.tum.de.
- When the Traceroute program terminates, stop packet capture in Wireshark.

At the end of the experiment, your Command Prompt Window should look something like Figure 7 in

MacOS, like Figure 8 in Windows. In Figure 7, the client was a random location in Chennai, and the target destination is in Munich. In Figure 8, the client was in IIT-M CSE department and the target destination is in Munich. From the figures we see that for each TTL value, the source program sends three probe packets. Traceroute displays the RTTs for each of the probe packets, as well as the IP address (and possibly the name) of the router that returned the ICMP TTL-exceeded message.

No.	Time	Source	Destination	Protocol	cumulative	Length	Info
97	6.504229	192.168.43.6	129.187.255.228	UDP	66	66	36938 → 33435 Len=24
98	6.505784	192.168.43.1	192.168.43.6	ICMP	160	94	Time-to-live exceeded (Time to live exceeded in transit)
99	6.507844	192.168.43.6	129.187.255.228	UDP	226	66	36938 → 33436 Len=24
100	6.509798	192.168.43.1	192.168.43.6	ICMP	320	94	Time-to-live exceeded (Time to live exceeded in transit)
101	6.509993	192.168.43.6	129.187.255.228	UDP	386	66	36938 → 33437 Len=24
102	6.511233	192.168.43.1	192.168.43.6	ICMP	480	94	Time-to-live exceeded (Time to live exceeded in transit)
103	6.511368	192.168.43.6	129.187.255.228	UDP	546	66	36938 → 33438 Len=24
104	11.516354	192.168.43.6	129.187.255.228	UDP	612	66	36938 → 33439 Len=24
106	16.520396	192.168.43.6	129.187.255.228	UDP	678	66	36938 → 33440 Len=24
258	21.520989	192.168.43.6	129.187.255.228	UDP	744	66	36938 → 33441 Len=24
259	21.561377	10.50.122.73	192.168.43.6	ICMP	814	70	Time-to-live exceeded (Time to live exceeded in transit)
260	21.562802	192.168.43.6	129.187.255.228	UDP	880	66	36938 → 33442 Len=24
261	21.600701	10.50.122.73	192.168.43.6	ICMP	950	70	Time-to-live exceeded (Time to live exceeded in transit)
262	21.600892	192.168.43.6	129.187.255.228	UDP	1016	66	36938 → 33443 Len=24
263	21.642459	10.50.122.73	192.168.43.6	ICMP	1086	70	Time-to-live exceeded (Time to live exceeded in transit)
264	21.642683	192.168.43.6	129.187.255.228	UDP	1152	66	36938 → 33444 Len=24
265	21.682581	10.0.66.181	192.168.43.6	ICMP	1222	70	Time-to-live exceeded (Time to live exceeded in transit)
266	21.683813	192.168.43.6	129.187.255.228	UDP	1288	66	36938 → 33445 Len=24
267	21.717586	10.0.66.181	192.168.43.6	ICMP	1358	70	Time-to-live exceeded (Time to live exceeded in transit)
268	21.717745	192.168.43.6	129.187.255.228	UDP	1424	66	36938 → 33446 Len=24
269	21.738410	10.0.66.181	192.168.43.6	ICMP	1494	70	Time-to-live exceeded (Time to live exceeded in transit)
270	21.738591	192.168.43.6	129.187.255.228	UDP	1560	66	36938 → 33447 Len=24
271	21.757459	125.17.180.201	192.168.43.6	ICMP	1630	70	Time-to-live exceeded (Time to live exceeded in transit)
272	21.758691	192.168.43.6	129.187.255.228	UDP	1696	66	36938 → 33448 Len=24

Internet Protocol Version 4, Src: 192.168.43.1, Dst: 192.168.43.6

Internet Control Message Protocol

Type: 11 (Time-to-live exceeded)

Code: 0 (Time to live exceeded in transit)

Checksum: 0x6280 [correct]

[Checksum Status: Good]

Internet Protocol Version 4, Src: 192.168.43.6, Dst: 129.187.255.228

User Datagram Protocol, Src Port: 36938, Dst Port: 33435

Source Port: 36938

Destination Port: 33435

Length: 32

Checksum: 0x7f79 [unverified]

0000 b0 34 95 ee 80 ac 88 b4 a6 28 76 16 08 00 45 c0 ...4.....(v...E...
 0010 00 50 93 98 00 00 40 01 0e fd c0 a8 2b 01 c0 a8 ...P...@...+...+...
 0020 2b 06 0b 00 62 00 00 00 00 45 00 00 34 90 4b ...+...b...E...4...K...
 0030 00 00 01 11 bc 1f c0 a8 2b 06 01 bb ff e4 90 4a+...J...
 0040 82 9b 00 20 7f 79 00 00 00 00 00 00 00 00 00 ...y.....

Figure 9: Wireshark packet trace for Traceroute to www.tum.de from a MacOS system

No.	Time	Source	Destination	Protocol	Length	Info
5738	67.183767	10.21.229.102	129.187.255.228	ICMP	106	Echo (ping) request id=0x0001, seq=87/22272, ttl=13 (no response found!)
5742	67.342319	188.1.145.230	10.21.229.102	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
5743	67.343984	10.21.229.102	129.187.255.228	ICMP	106	Echo (ping) request id=0x0001, seq=88/22528, ttl=13 (no response found!)
5767	67.502483	188.1.145.230	10.21.229.102	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
5768	67.503691	10.21.229.102	129.187.255.228	ICMP	106	Echo (ping) request id=0x0001, seq=89/22784, ttl=13 (no response found!)
5806	67.662188	188.1.145.230	10.21.229.102	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
5863	68.517145	10.21.229.102	129.187.255.228	ICMP	106	Echo (ping) request id=0x0001, seq=90/23040, ttl=14 (no response found!)
5874	68.675642	188.1.145.230	10.21.229.102	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
5875	68.678644	10.21.229.102	129.187.255.228	ICMP	106	Echo (ping) request id=0x0001, seq=91/23296, ttl=14 (no response found!)
5897	68.837040	188.1.145.230	10.21.229.102	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
5898	68.839902	10.21.229.102	129.187.255.228	ICMP	106	Echo (ping) request id=0x0001, seq=92/23552, ttl=14 (no response found!)
5935	68.998361	188.1.145.230	10.21.229.102	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
5998	69.848771	10.21.229.102	129.187.255.228	ICMP	106	Echo (ping) request id=0x0001, seq=93/23808, ttl=15 (no response found!)
6026	70.007195	129.187.0.168	10.21.229.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
6028	70.009489	10.21.229.102	129.187.255.228	ICMP	106	Echo (ping) request id=0x0001, seq=94/24064, ttl=15 (no response found!)

Frame 5767: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface 0

Ethernet II, Src: Cisco_57:ec:40 (00:c8:8b:57:ec:40), Dst: Dell_9e:0c:1e (98:90:96:9e:0c:1e)

Internet Protocol Version 4, Src: 188.1.145.230, Dst: 10.21.229.102

Internet Control Message Protocol

Type: 11 (Time-to-live exceeded)

Code: 0 (Time to live exceeded in transit)

Checksum: 0xf4ee [correct]

[Checksum Status: Good]

Unused: 00

Length: 17

[Length of original datagram: 68]

Unused: 0000

Internet Protocol Version 4, Src: 10.21.229.102, Dst: 129.187.255.228

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 92

Identification: 0x6d00 (27904)

Flags: 0x0000

Figure 10: Wireshark packet trace for Traceroute to www.tum.de from a Windows system

Figure 9 displays the Wireshark window for an ICMP packet returned by a router. This is the output when "traceroute" is executed in MacOS. Note that this ICMP error packet contains many more fields than the

Ping ICMP messages. Figure 10 is the Wireshark capture for the "tracert" execution in Windows. The figures show Wireshark window of ICMP fields expanded for one ICMP error packet.

What to Submit:

For this part of the assignment, you should hand in a screen shot of the Command Prompt /Console window. Whenever possible, when answering a question below, you should take a screenshot/snapshot of the packet(s) within the trace that you used to answer the question asked. Provide suitable response along with the snapshot (picture) to explain your answer. Select the needful packet contents and capture suitable screen shots. The Wireshark file submitted must have the packets from which the screen shot has been enabled. The above figures are examples of the screen shots expected.

Answer the following questions: (A word / PDF document)

5. What is the IP address of your host? What is the IP address of the target destination host?
6. If ICMP sent UDP packets instead (as in Unix/Linux), would the IP protocol number still be 01 for the probe packets? If not, what would it be?
7. Examine the ICMP echo packet in your screenshot. Is this different from the ICMP ping query packets in the first half of this lab? If yes, how so?
8. Examine the ICMP error packet in your screenshot. It has more fields than the ICMP echo packet. What is included in those fields?
9. Examine the last three ICMP packets received by the source host. How are these packets different from the ICMP error packets? Why are they different?
10. Within the tracert measurements, is there a link whose delay is significantly longer than others? Refer to the screenshot in Figure 4, is there a link whose delay is significantly longer than others? On the basis of the router names, can you guess the location of the two routers on the end of this link?

3 Grading

3.1 DNS

- Wireshark and screen captures: 30 points
- Answers to questions: 30 points
- Viva Voce: 5 points

3.2 ICMP

- Wireshark and screen captures: 15 points
- Answers to questions: 15 points
- Viva Voce: 5 points

4 Help

1. Ask questions EARLY and start your work NOW. Take advantage of the help of the TAs and the instructor.

2. Submissions PAST the extended deadline SHOULD NOT be mailed to the TAs. Only submissions approved by the instructor or uploaded to Moodle within the deadline will be graded.
3. Demonstration of command execution, explanation of behavior to the TAs MUST be done using the files uploaded on Moodle.
4. Execute the commands / perform the study using your individual laptops or distinct workstations in the labs.
5. The wireshark captures will need to be distinct. If two submissions have same source IP / MAC, it will be treated as copy and will receive “Zero” credits.
6. Try to be creative as much as possible.