



Ingeniería en Robótica y Telecomunicaciones
Departamento de computación, electrónica y mecatrónica.

Course Name:

COMPUTER NETWORK O24-LRT4042-1

Members & ID:

Jonathan Eliasib Rosas Tlaczani - 168399

Malcolm Thompson - 161044

HOMEWORK 4

04/11/24

Abstract

This experiment investigates the behavior of the ICMP, IP, and DHCP protocols in a network by analyzing the structure and function of each through packet captures and specific commands. Using tools like Wireshark and PingPlotter, we capture and analyze messages generated by the Ping and Traceroute applications, as well as DHCP packets. Key aspects, such as IP datagram fragmentation and the nature of ICMP error messages, are explored to provide a practical insight into packet transmission and IP management in a network. This study aids in understanding network diagnostics and troubleshooting.

Introduction

In modern digital communications, networks rely on a range of protocols to facilitate connectivity, manage data transfer, and provide diagnostic tools to troubleshoot issues. Among these, the **Internet Control Message Protocol (ICMP)**, **Internet Protocol (IP)**, and **Dynamic Host Configuration Protocol (DHCP)** serve as foundational elements that ensure devices on a network can communicate efficiently, identify connectivity issues, and dynamically allocate IP addresses to manage network resources. Each protocol addresses a different aspect of network functionality, yet they work in tandem to maintain the stability, reliability, and performance of data networks (Kurose & Ross, 2008).

The **Internet Control Message Protocol (ICMP)** is essential for network diagnostics and error reporting. Developed as part of the Internet Protocol Suite, ICMP enables devices to send control messages and error notifications between hosts. It is most commonly known for its use in the Ping and Traceroute commands, which help verify network connectivity and trace the path that data packets take to reach their destination. Ping, for instance, sends ICMP echo request messages to a target host and waits for echo replies to determine if the host is reachable and to measure round-trip time. Traceroute, on the other hand, sends ICMP messages with increasing time-to-live (TTL) values to map each "hop" a packet takes through routers until it reaches its final destination (Postel, 1981). This protocol helps network administrators quickly identify the points of failure in a network path, enabling efficient troubleshooting and optimization of network performance. The ICMP packet structure, which includes fields like Type, Code, and Checksum, is specifically designed to deliver feedback on network issues without overwhelming the network with additional data (Wikipedia contributors, 2024a).

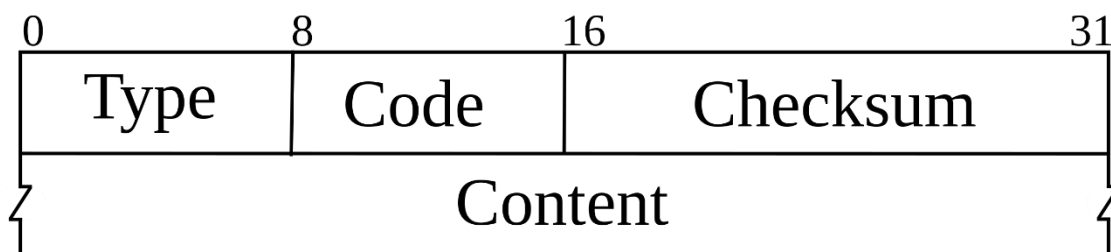


Figure 1. Header for ICMPV4

Meanwhile, **Internet Protocol (IP)** serves as the backbone of network communication by handling the addressing and routing of data packets. IP assigns a unique address to each device on a network, ensuring that data packets sent from one device are correctly routed to their intended destination. IP also plays a crucial role in **packet fragmentation**, a process necessary for transmitting data across networks with different **Maximum Transmission Unit (MTU)** sizes. When data packets must traverse networks with varying MTU restrictions, IP fragments these packets to ensure compatibility with the smallest MTU encountered along the path. Each fragment is labeled with sequence information so that the destination host can reassemble them into the original packet upon arrival. IP fragmentation is particularly important in heterogeneous networks, where devices with different capabilities and requirements must communicate seamlessly (Wikipedia contributors, 2024b).

The fragmentation process begins when a data packet exceeds the MTU of a link in the network path. IP divides the packet into smaller fragments, each of which carries the same IP header information along with additional fragmentation details, such as the offset value indicating the position of each fragment relative to the original packet. The "More Fragments" flag is set for all fragments except the last one, signaling to the receiving device that it should wait for additional fragments before reassembling the data. Although fragmentation enables data to traverse networks with varying MTU sizes, it can also increase the chances of packet loss, as all fragments must arrive for reassembly to succeed. Understanding how IP handles fragmentation is essential for network optimization and for avoiding bottlenecks, particularly when large data transfers are involved (Kurose & Ross, 2008).

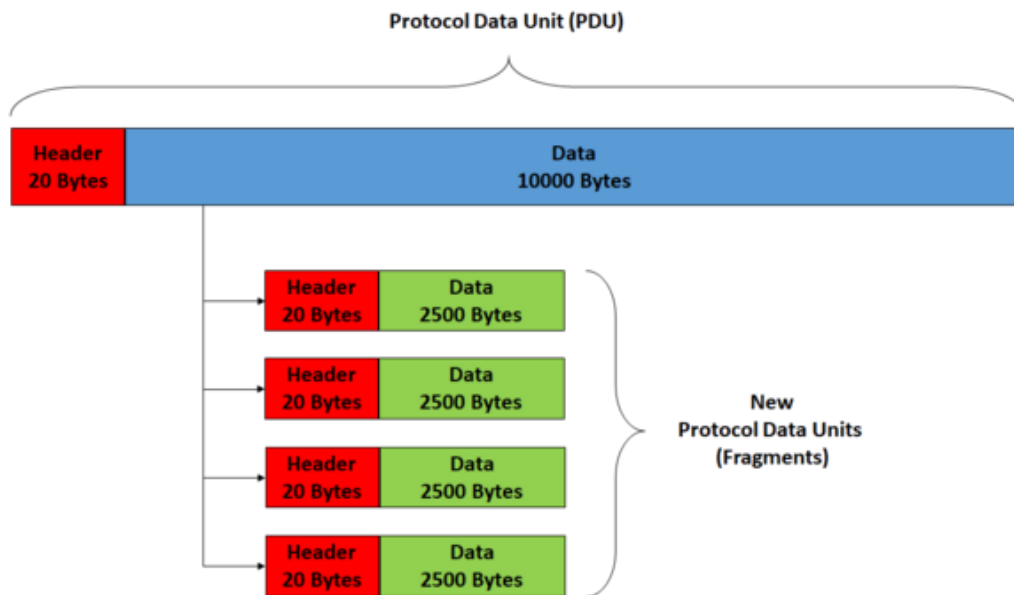


Figure 2. Fragmentation of a Protocol Data Unit

The **Dynamic Host Configuration Protocol (DHCP)** automates the assignment of IP addresses to devices within a network. This protocol is especially important in large networks where manually configuring IP addresses for each device would be impractical and error-prone. DHCP operates through a sequence known as DORA (Discover, Offer, Request, Acknowledge). When a device connects to the

network, it sends a DHCP Discover message to indicate it needs an IP address. In response, the DHCP server sends an Offer message, providing an available IP address and additional network configuration details. The device then sends a Request to confirm the IP address, and the server concludes the process with an Acknowledge message (Droms, 1997). This four-step handshake allows each device to obtain a unique IP address and necessary network configuration details, such as subnet mask, default gateway, and DNS server addresses. DHCP also helps manage IP address leases, ensuring that addresses are reused and available for new devices when no longer needed by the original device (Wikipedia contributors, 2024c).

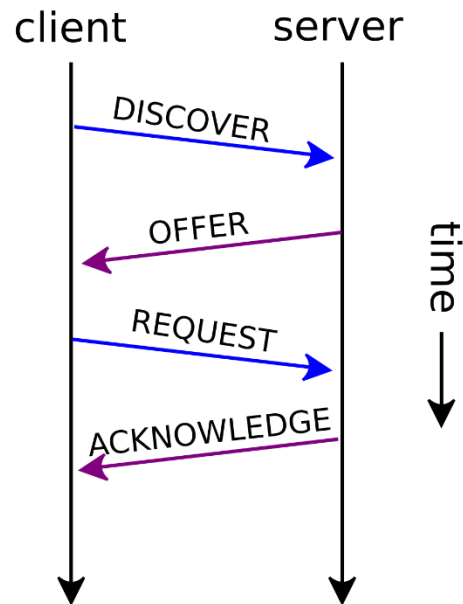


Figure 3. Non-renewing DHCP session

In this study, we use tools like Wireshark and PingPlotter to capture and analyze packets generated by ICMP, IP, and DHCP protocols, allowing for a practical exploration of these protocols' roles in network operations. Wireshark, a packet-sniffing tool, provides insights into packet structures and allows users to filter by protocol type, enabling detailed analysis of ICMP messages generated by Ping and Traceroute as well as DHCP transactions (Wireshark, n.d.). PingPlotter offers additional visualization capabilities for tracing packet routes, making it a valuable tool for observing how TTL values change at each network hop (PingPlotter, n.d.). Through these tools, we aim to understand how ICMP messages facilitate network troubleshooting, how IP manages packet fragmentation, and how DHCP dynamically assigns IP addresses to devices, supporting seamless connectivity.

Understanding these protocols provides insight into how networks handle a wide variety of challenges, from ensuring efficient data routing to detecting connectivity issues and managing IP address allocations. Each protocol contributes uniquely to the network's overall functionality and performance, ensuring that devices can connect, communicate, and troubleshoot issues with minimal human intervention. By dissecting the mechanisms of ICMP, IP, and DHCP, this study highlights the protocols' interdependent

roles in maintaining network integrity and supporting the global infrastructure that powers the Internet (Kurose & Ross, 2008).

Methodology

Capturing ICMP Packets

Ping Program

1. Initiated packet capture in Wireshark.
2. Executed the Ping command in the Command Prompt using `ping -n 10 hostname` to send 10 ICMP requests.
3. Concluded the capture in Wireshark after the ping test completed.
4. Applied an ICMP filter to focus solely on ICMP packets.

Traceroute Program

1. Launched the Windows Command Prompt.
2. Started packet capture in Wireshark.
3. Ran the Traceroute command using `tracert hostname` to map the route to the target.
4. Ended the capture in Wireshark upon completion of the Traceroute.

Capturing IP Packets

1. Set up and initiated a packet capture session in Wireshark.
2. Sent ICMP Echo Request messages of varying sizes to the target host using PingPlotter.
3. Stopped the capture in Wireshark after completing the tests.

Capturing DHCP Packets

1. Opened the Windows Command Prompt.
2. Used the command `ipconfig /release` to release the current IP address.
3. Began packet capture in Wireshark.
4. Requested a new IP address with `ipconfig /renew`.
5. Repeated the renewal process to capture additional DHCP traffic.
6. Released and renewed the IP address once more.
7. Stopped the capture in Wireshark at the end of the DHCP sequence.

Results and Discussions

1. Capturing ICMP packets

Let's begin the 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 to verify whether a host is alive 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, both of these Ping packets are ICMP packets. Do the following:

- Start up the Wireshark packet sniffer [2], and begin Wireshark packet capture.
- The ping command is in c:\windows\system, so type “ping -n 10 hostname” in the MS-DOS command line, where hostname is any host. The argument “-n 10” indicates that 10 ping messages should be sent. Then run the Ping program by typing return.
- When the Ping program terminates, stop the packet capture in Wireshark.
- Note that “icmp” must be entered into the filter display window.

```
C:\Windows\System32>ping -n 10 facebook.com

Haciendo ping a facebook.com [157.240.25.35] con 32 bytes de datos:
Respuesta desde 157.240.25.35: bytes=32 tiempo=11ms TTL=57
Respuesta desde 157.240.25.35: bytes=32 tiempo=10ms TTL=57
Respuesta desde 157.240.25.35: bytes=32 tiempo=11ms TTL=57
Respuesta desde 157.240.25.35: bytes=32 tiempo=10ms TTL=57
Respuesta desde 157.240.25.35: bytes=32 tiempo=11ms TTL=57
Respuesta desde 157.240.25.35: bytes=32 tiempo=11ms TTL=57
Respuesta desde 157.240.25.35: bytes=32 tiempo=15ms TTL=57
Respuesta desde 157.240.25.35: bytes=32 tiempo=11ms TTL=57
Respuesta desde 157.240.25.35: bytes=32 tiempo=11ms TTL=57
Respuesta desde 157.240.25.35: bytes=32 tiempo=71ms TTL=57

Estadísticas de ping para 157.240.25.35:
    Paquetes: enviados = 10, recibidos = 10, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 10ms, Máximo = 71ms, Media = 17ms
```

Figure 4. Ping Window

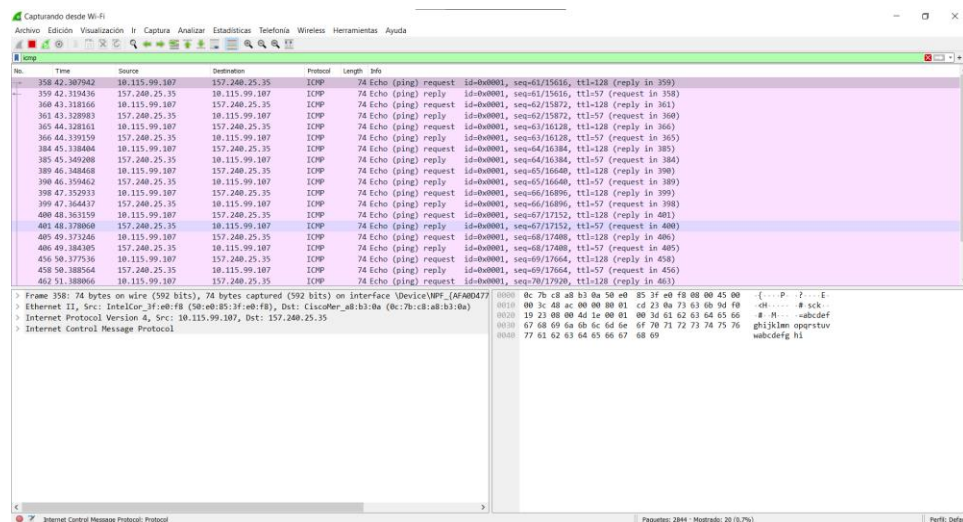
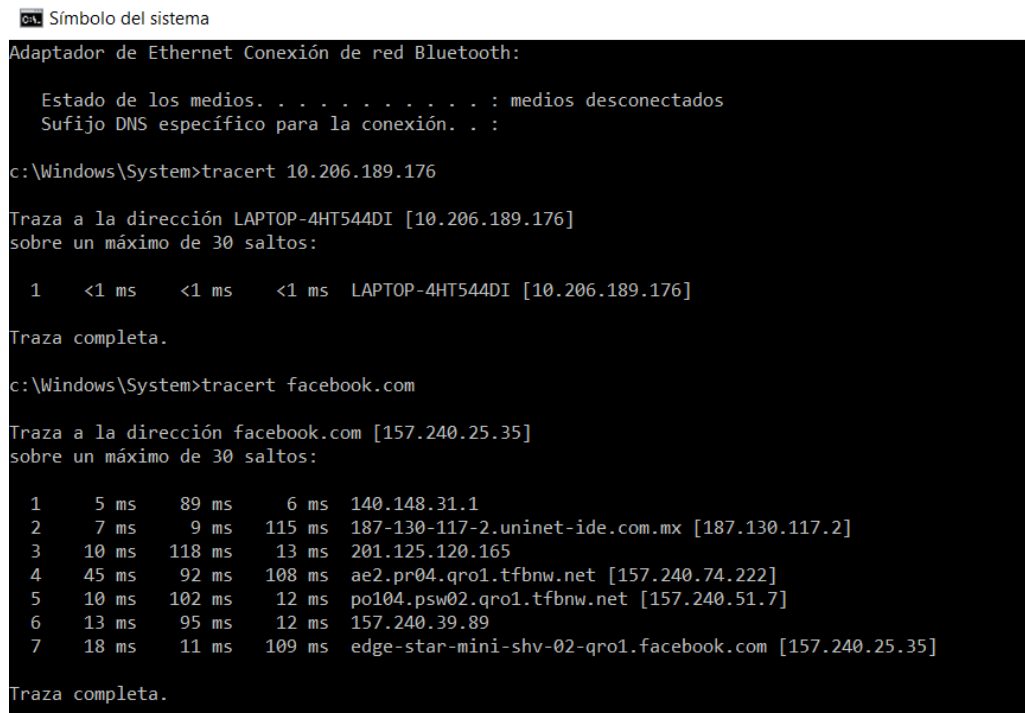


Figure 5. Entering ICMP

Let's now continue the ICMP adventure by capturing the packets generated by the Traceroute (tracert) program. You may recall that the *tracert* program can be used to figure out the path a packet takes from source to destination. Do the following:

- Let's begin by opening the Windows Command Prompt application.
- Start up the Wireshark packet sniffer, and begin Wireshark packet capture. The *tracert* command is in c:\windows\system, so type "*tracert* hostname" in the MS-DOS command line, where hostname is any host.
- Then run the Traceroute program by typing return. When the Traceroute program terminates, stop packet capture in Wireshark.



```

C:\Windows\System>tracert 10.206.189.176

Traza a la dirección LAPTOP-4HT544DI [10.206.189.176]
sobre un máximo de 30 saltos:

 1    <1 ms    <1 ms    <1 ms    LAPTOP-4HT544DI [10.206.189.176]

Traza completa.

C:\Windows\System>tracert facebook.com

Traza a la dirección facebook.com [157.240.25.35]
sobre un máximo de 30 saltos:

 1      5 ms     89 ms     6 ms    140.148.31.1
 2      7 ms     9 ms     115 ms   187-130-117-2.uninet-ide.com.mx [187.130.117.2]
 3     10 ms    118 ms    13 ms    201.125.120.165
 4     45 ms     92 ms    108 ms   ae2.pr04.qro1.tfbnw.net [157.240.74.222]
 5     10 ms    102 ms    12 ms    po104.psw02.qro1.tfbnw.net [157.240.51.7]
 6     13 ms     95 ms    12 ms    157.240.39.89
 7     18 ms     11 ms    109 ms   edge-star-mini-shv-02-qro1.facebook.com [157.240.25.35]

Traza completa.

```

Figure 6. Trace Route

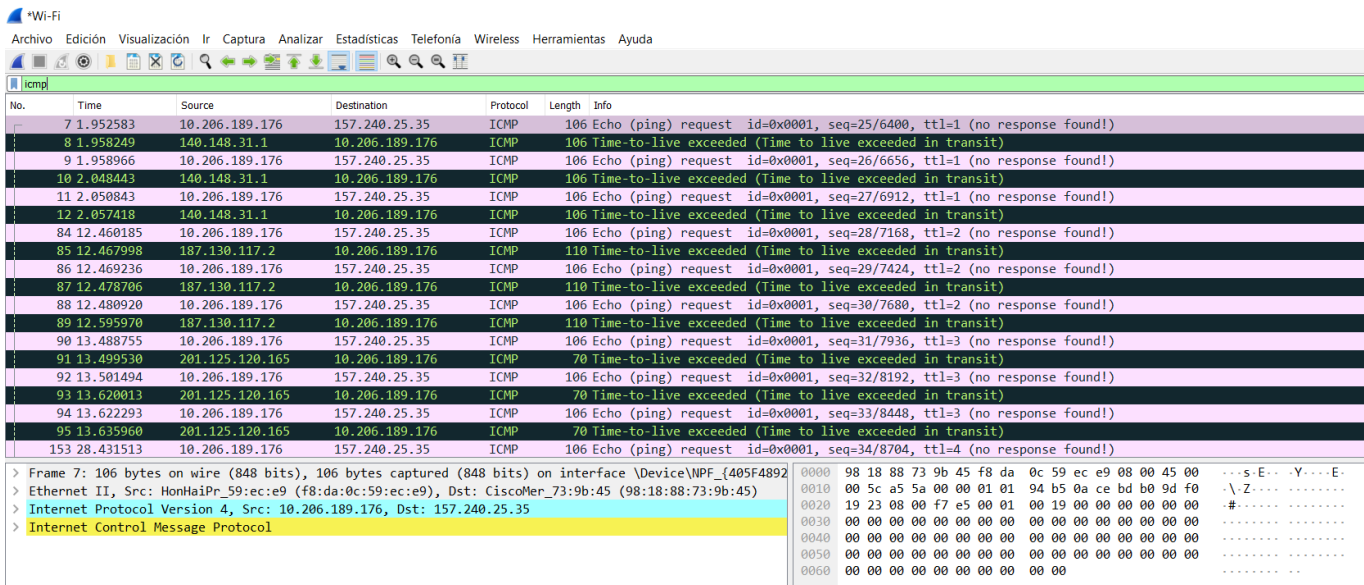


Figure 7. Packet Capture of Trace Route

For this part of the lab, you should hand in a screen shot of the Command Prompt window. Whenever possible, when answering a question below, you should hand in a printout of the packet(s) within the trace that you used to answer the question asked. Annotate the printout to explain your answer. To print a packet, use *File*→*Print*, choose Selected packet only, choose Packet summary line, and select the minimum amount of packet detail that you need to answer the question. Answer the following questions for the Ping application:

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

Source Address: 10.115.99.107

Destination Address: 157.240.25.35

2. Why is it that an ICMP packet does not have source and destination port numbers?

The ICMP packet does not include source and destination port numbers because its purpose is to transmit network-layer data between hosts and routers, rather than supporting communication between application layer functions.

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?

Type: 8

Code: 0

Other Fields: Checksum, Sequence Number, Identifier

Checksum: [0x4cea]

Sequence Number: [113 (0x0072)]

Identifier: [256 (0x0001)]

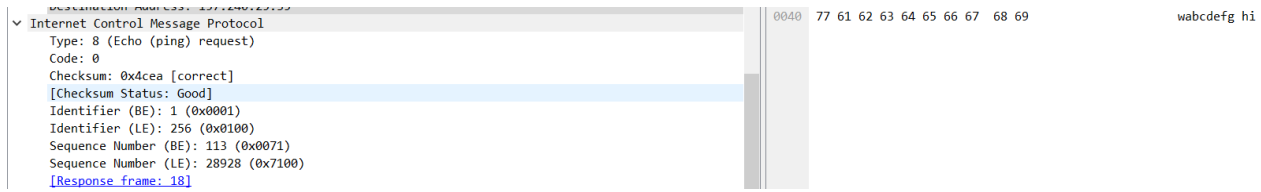


Figure 8. Type and Code Numbers

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?

ICMP Type: 8 (Echo Reply)

ICMP Code: 0

Other Fields: Checksum, Sequence Number, Identifier

Checksum: [0x338a]

Sequence Number: [691 (0x02b3)]

Identifier: [1 (0x0001)]

Now for the traceroute application:

5. What is the IP address of your host? What is the IP address of the target destination host?

The IP address of the host: 192.168.1.3

6. Examine the ICMP echo packet in your screenshot. Is this different from the ICMP ping query packets? If yes, how so?

The ICMP Echo packet in the screenshot is different from the ICMP Ping Query packets.

7. Examine the ICMP error packet in your screenshot. It has more fields than the ICMP echo packet. What is included in those fields?

The ICMP Echo packet in the screenshot is different from the ICMP Ping Query packets.

8. 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?

The last three ICMP packets are different from ICMP error packets because they indicate successful hops.

9. Within the *tracert* measurements, 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?

It's not clear.

2. Capturing IP packets

In order to generate a trace of IP datagrams for this task, we will use the traceroute program to send datagrams of different sizes towards some destination, X. Recall that traceroute operates by first sending one or more datagrams with the time-to-live (TTL) field in the IP header set to 1; it then sends a series of one or more datagrams towards the same destination with a TTL value of 2; it then sends a series of datagrams towards the same destination with a TTL value of 3; and so on. Recall that a router must decrement the TTL in each received datagram by 1. If the TTL reaches 0, the router returns an ICMP message (type 11 – TTL-exceeded) to the sending host. We will want to run traceroute and have it send datagrams of various lengths. The *tracert* program provided with Windows does not allow one to change the size of the ICMP echo request (ping) message sent by the tracert program. A nice Windows traceroute program is **pingplotter** [3]. Download and install pingplotter, and test it out by performing a few traceroutes to your favorite sites. The size of the ICMP echo request message can be explicitly set in pingplotter by selecting the menu item *Edit→Advanced Options→Packet Options* and then filling in the Packet Size field. The default packet size is 56 bytes. Once pingplotter has sent a series of packets with the increasing TTL values, it restarts the sending process again with a TTL of 1, after waiting Trace Interval amount of time. The value of Trace Interval and the number of intervals can be explicitly set in pingplotter [1]. Do the following:

- Start up Wireshark [2] and begin packet capture and then press OK on the Wireshark Packet Capture Options screen.
- If you are using a Windows platform, start up *pingplotter* and enter the name of a target destination in the “Address to Trace Window.” Enter 3 in the “# of times to Trace” field, so you do not gather too much data. Select the menu item *Edit→Advanced Options→Packet Options* and enter first a value of 56 in the Packet Size field and then press OK. Then press the Trace button. Next, send a set of datagrams with a longer length, by selecting *Edit→Advanced Options→Packet Options* and enter a value of 2000 in the Packet Size field and then press OK. Then press the Resume button. Finally, send a set of datagrams with a longer length, by selecting *Edit→Advanced Options→Packet Options* and enter a value of 3500 in the Packet Size field and then press OK. Then press the Resume button.
- Stop Wireshark tracing.

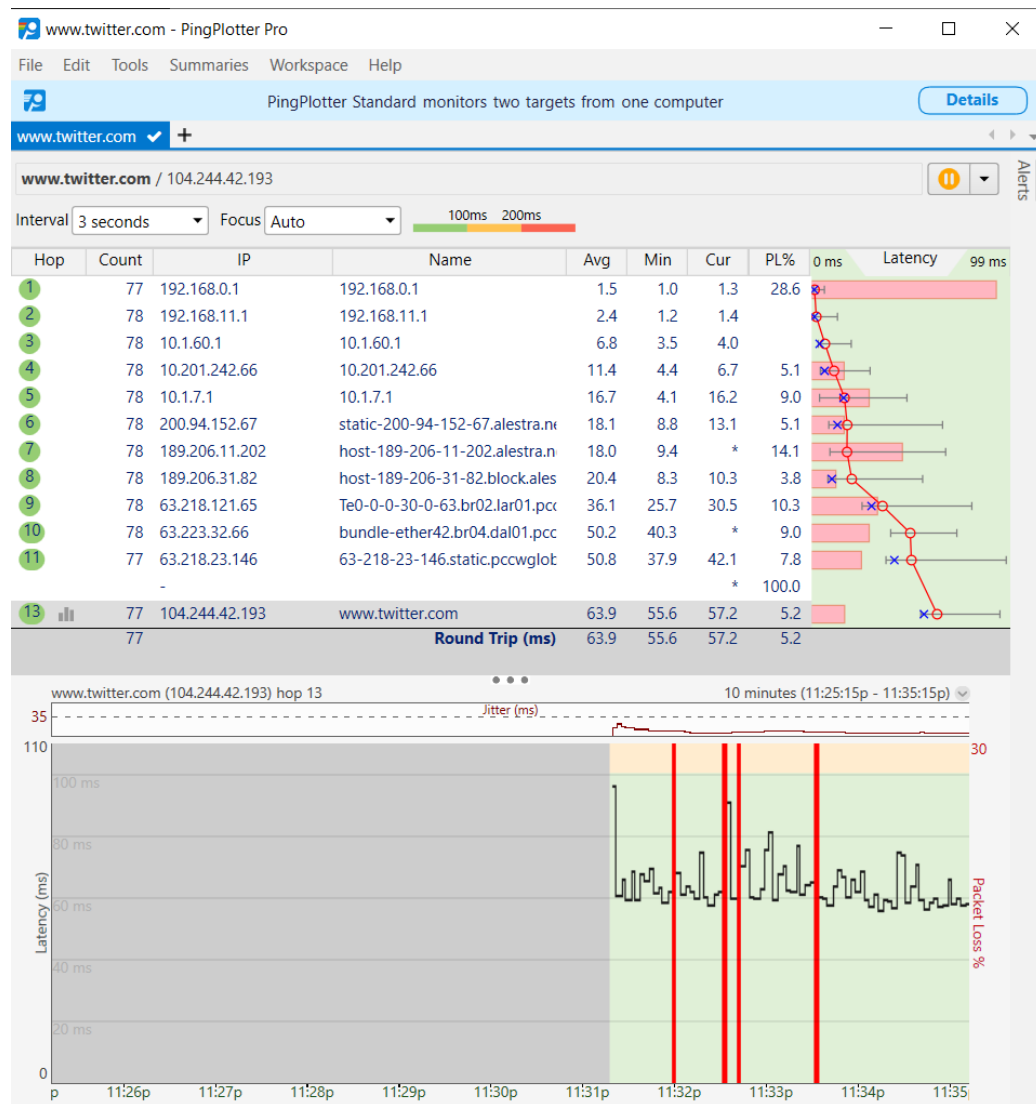


Figure 9. Pingplotter Test 1

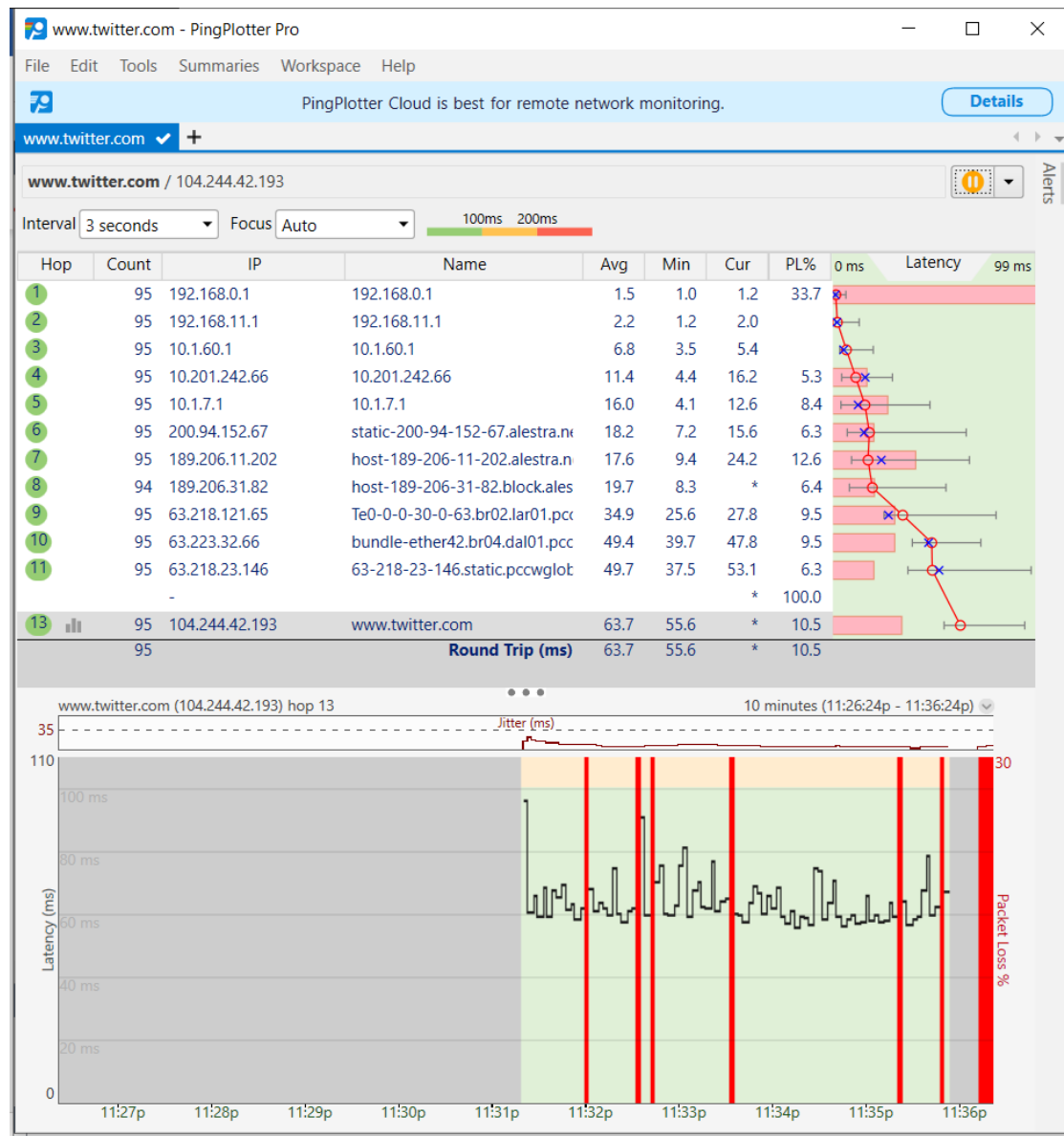


Figure 10. Pingplotter Test 2

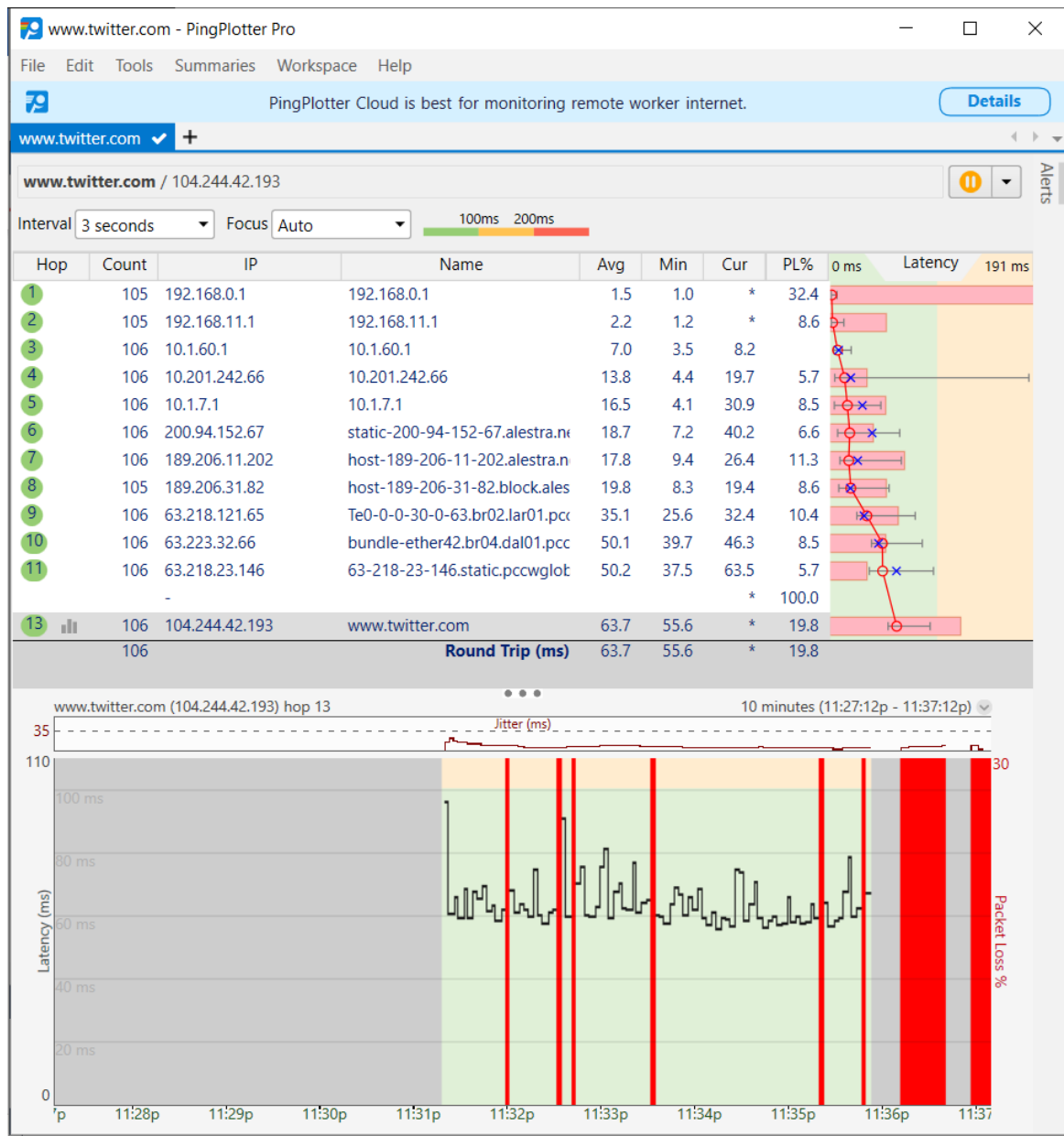


Figure 11. Pingplotter Test 2

In your trace, you should be able to see the series of ICMP Echo Request sent by your computer and the ICMP TTL-exceeded messages returned to your computer by the intermediate routers. Whenever possible, when answering a question you should hand in a printout of the packet(s) within the trace that you used to answer the question asked. Annotate the printout to explain your answer. To print a packet,

use *File*→*Print*, choose Selected packet only, choose Packet summary line, and select the minimum amount of packet detail that you need to answer the question.

1. Select the first ICMP Echo Request message sent by your computer, and expand the Internet Protocol part of the packet in the packet details window. What is the IP address of your computer?

The IP address of the computer is 192.168.0.119

2. Within the IP packet header, what is the value in the upper layer protocol field?

The value in the upper layer protocol field is ICMP (1)

3. How many bytes are in the IP header? How many bytes are in the payload of the IP datagram? Explain how you determined the number of payload bytes.

The Header length are 20 bytes and 56 in the total length, this give us 36 bytes of IP datagram payload.

4. Has this IP datagram been fragmented? Explain how you determined whether or not the datagram has been fragmented.

No, since in the more fragments flag bit = 0, this means that it is not fragmented.

```

-----
v 000. .... = Flags: 0x0
    0... .... = Reserved bit: Not set
    .0.. .... = Don't fragment: Not set
    ..0. .... = More fragments: Not set
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 13
Protocol: ICMP (1)
Header Checksum: 0x8849 [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.0.119
Destination Address: 104.244.42.129
  
```

Figure 12. Datagram

Next, sort the traced packets according to IP source address. Select the first ICMP Echo Request message sent by your computer, and expand the Internet Protocol portion in the “details of selected packet header” window. In the “*listing of captured packets*” window, you should see all of the subsequent ICMP messages (perhaps with additional interspersed packets sent by other protocols running on your

computer) below this first ICMP. Use the down arrow to move through the ICMP messages sent by your computer [1, 2].

5. Which fields in the IP datagram always change from one datagram to the next within this series of ICMP messages sent by your computer?

The TTL and the checksum change between each packet.

6. Which fields stay constant? Which of the fields must stay constant? Which fields must change? Why?

Since the header length and time to live are fixed, they remain unchanged. However, each segment varies in terms of its fragment number, sequence number, flags, total length, and checksum, meaning these values will differ.

7. Describe the pattern you see in the values in the Identification field of the IP datagram. Next (with the packets still sorted by source address) find the series of ICMP TTL-exceeded replies sent to your computer by the nearest (first hop) router.

Every new message that is sent out increases the value of the identification field by 1.

8. What is the value in the Identification field and the TTL field?

Identification: 0xd0e7 (53479) Time to Live: 13

9. Do these values remain unchanged for all of the ICMP TTL-exceeded replies sent to your computer by the nearest (first hop) router? Why?

The ID field must be unique, so it is modified in all TTL Exceeded ICMP responses. If two or more IP datagrams share the same ID, it means they are fragments of a larger, single IP datagram. Because the TTL value at the router's first hop remains constant, the TTL field also stays unchanged.

Fragmentation

Sort the packet listing according to time again by clicking on the Time column [1, 2].

10. Find the first ICMP Echo Request message that was sent by your computer after you changed the Packet Size in pingplotter to be 2000. Has that message been fragmented across more than one IP datagram?

The first ICMP Echo Request message sent after changing the packet size to 2000 has been fragmented.

11. Print out the first fragment of the fragmented IP datagram. What information in the IP header indicates that the datagram been fragmented? What information in the IP header indicates whether this is the first fragment versus a latter fragment? How long is this IP datagram?

Identification field indicates fragmentation.

Flags indicate the More Fragments bit is set.

Total Length: [1500]

12. Print out the second fragment of the fragmented IP datagram. What information in the IP header indicates that this is not the first datagram fragment? Are there more fragments? How can you tell?

More Fragments bit set.

Flags indicate this is not the first fragment.

The Identification field is the same.

Total Length: [1480]

13. What fields change in the IP header between the first and second fragment? Now find the first ICMP Echo Request message that was sent by your computer after you changed the Packet Size in *pingplotter* to be 3500.

Fields that change in the IP header between the first and second fragments are Offset and More Fragments bit.

14. How many fragments were created from the original datagram?

Extracted from the original datagram were three fragments.

15. What fields change in the IP header among the fragments?

Upon analyzing these three packets, it was found that the fragment offset and checksum were the IP header fields that changed for each packet. The total length and flags differ between the last packet and the first two. The first two packets, with the "More Fragments" flag set, have a total length of 1500, while the final packet has a total length of 540 and the "More Fragments" bit set to 0. The fields that vary include the fragment offset, total length, "More Fragments" bit, TTL, and checksum.

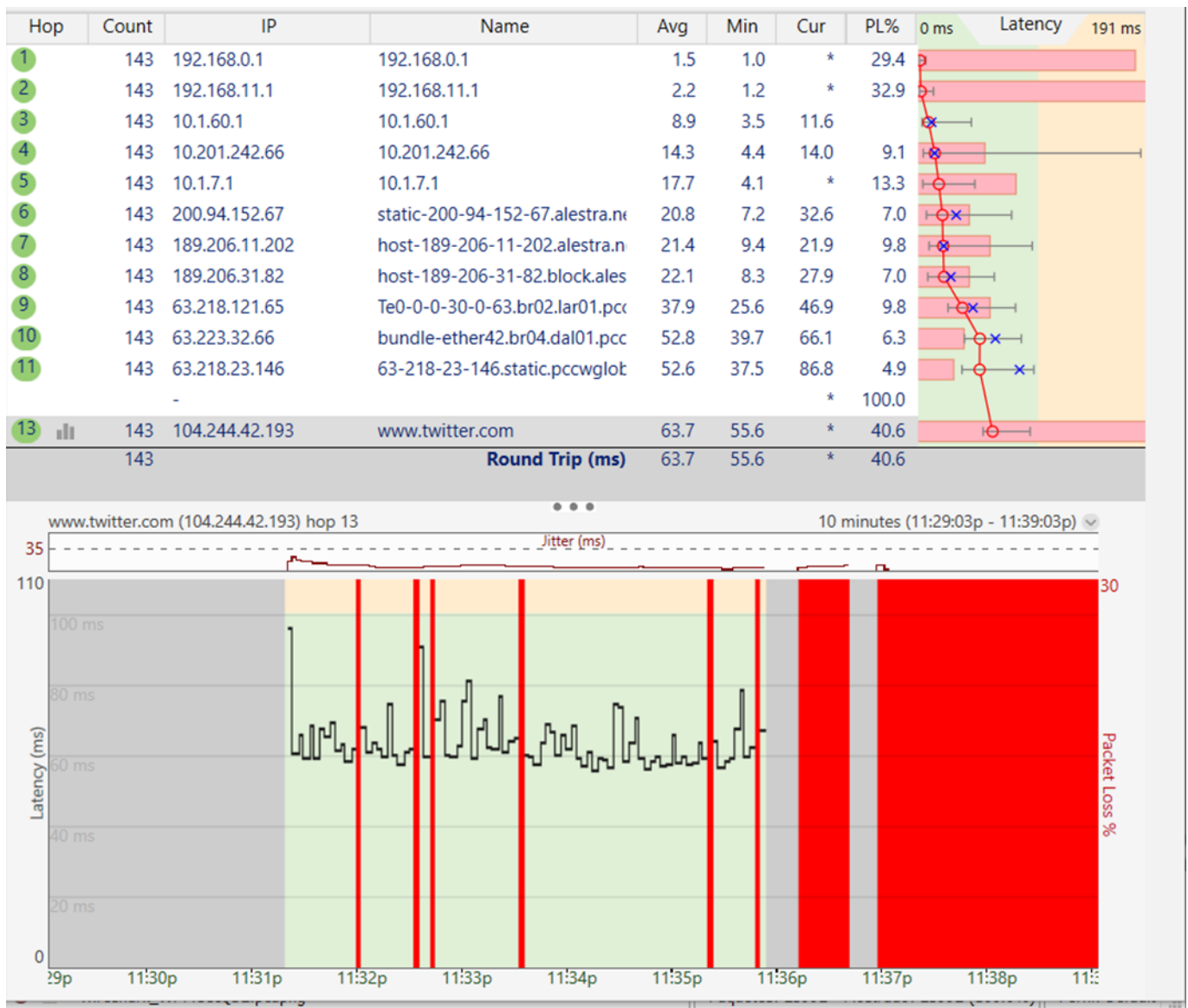


Figure 13. Ping Plotter

3. Capturing DHCP packets

In order to observe DHCP in action, we will perform several DHCP-related commands and capture the DHCP messages exchanged as a result of executing these commands. Do the following [1]:

- Begin by opening the Windows Command Prompt application, enter `ipconfig /release`. This command releases your current IP address, so that your host's IP address becomes 0.0.0.0.

```
C:\Windows\System>ipconfig/release

Configuración IP de Windows

No se puede realizar ninguna operación en Ethernet mientras los medios
estén desconectados.
No se puede realizar ninguna operación en Conexión de área local* 1 mientras los medios
estén desconectados.
No se puede realizar ninguna operación en Conexión de área local* 2 mientras los medios
estén desconectados.
No se puede realizar ninguna operación en Conexión de red Bluetooth mientras los medios
estén desconectados.

Adaptador de Ethernet Ethernet:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

Adaptador de LAN inalámbrica Conexión de área local* 1:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

Adaptador de LAN inalámbrica Conexión de área local* 2:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

Adaptador de LAN inalámbrica Wi-Fi:

    Sufijo DNS específico para la conexión. . :
    Dirección IPv6 . . . . . : 2002:c0a8:b02:10:3519:6273:d295:3e18
    Dirección IPv6 temporal. . . . . : 2002:c0a8:b02:10:c858:904d:6391:6d58
    Vínculo: dirección IPv6 local. . . : fe80::e3e7:355a:18e2:8736%17
    Puerta de enlace predeterminada . . . . : fe80::9e53:22ff:fe7b:ff78%17

Adaptador de Ethernet Conexión de red Bluetooth:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :
```

Figure 14. ipconfig /release

- Start up the Wireshark packet sniffer [2], and begin Wireshark packet capture.

- Now go back to the Windows Command Prompt and enter “`ipconfig /renew`”. This instructs your host to obtain a network configuration, including a new IP address.

```
C:\Windows\System>ipconfig/renew

Configuración IP de Windows

No se puede realizar ninguna operación en Ethernet mientras los medios
estén desconectados.
No se puede realizar ninguna operación en Conexión de área local* 1 mientras los medios
estén desconectados.
No se puede realizar ninguna operación en Conexión de área local* 2 mientras los medios
estén desconectados.
No se puede realizar ninguna operación en Conexión de red Bluetooth mientras los medios
estén desconectados.

Adaptador de Ethernet Ethernet:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

Adaptador de LAN inalámbrica Conexión de área local* 1:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

Adaptador de LAN inalámbrica Conexión de área local* 2:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

Adaptador de LAN inalámbrica Wi-Fi:

    Sufijo DNS específico para la conexión. . :
    Dirección IPv6 . . . . . : 2002:c0a8:b02:10:3519:6273:d295:3e18
    Dirección IPv6 temporal. . . . . : 2002:c0a8:b02:10:c858:904d:6391:6d58
    Vínculo: dirección IPv6 local. . . : fe80::e3e7:355a:18e2:8736%17
    Dirección IPv4. . . . . : 192.168.0.119
    Máscara de subred . . . . . : 255.255.255.0
    Puerta de enlace predeterminada . . . . : fe80::9e53:22ff:feff:ff78%17
                                              192.168.0.1

Adaptador de Ethernet Conexión de red Bluetooth:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :
```

Figure 15. `ipconfig /renew`

- Wait until the “`ipconfig /renew`” has terminated. Then enter the same command “`ipconfig /renew`” again.

```
C:\Windows\System>ipconfig/renew

Configuración IP de Windows

No se puede realizar ninguna operación en Ethernet mientras los medios
estén desconectados.
No se puede realizar ninguna operación en Conexión de área local* 1 mientras los medios
estén desconectados.
No se puede realizar ninguna operación en Conexión de área local* 2 mientras los medios
estén desconectados.
No se puede realizar ninguna operación en Conexión de red Bluetooth mientras los medios
estén desconectados.

Adaptador de Ethernet Ethernet:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

Adaptador de LAN inalámbrica Conexión de área local* 1:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

Adaptador de LAN inalámbrica Conexión de área local* 2:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

Adaptador de LAN inalámbrica Wi-Fi:

    Sufijo DNS específico para la conexión. . :
    Dirección IPv6 . . . . . : 2002:c0a8:b02:10:3519:6273:d295:3e18
    Dirección IPv6 temporal. . . . . : 2002:c0a8:b02:10:c858:904d:6391:6d58
    Vínculo: dirección IPv6 local. . . : fe80::e3e7:355a:18e2:8736%17
    Dirección IPv4. . . . . : 192.168.0.119
    Máscara de subred . . . . . : 255.255.255.0
    Puerta de enlace predeterminada . . . . : fe80::9e53:22ff:feff:ff78%17
                                              192.168.0.1

Adaptador de Ethernet Conexión de red Bluetooth:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :
```

Figure 16. `ipconfig /renew` again

- When the second “`ipconfig /renew`” terminates, enter the command “`ipconfig /release`” to release the previously- allocated IP address to your computer.

```
C:\Windows\System>ipconfig/release

Configuración IP de Windows

No se puede realizar ninguna operación en Ethernet mientras los medios
estén desconectados.
No se puede realizar ninguna operación en Conexión de área local* 1 mientras los medios
estén desconectados.
No se puede realizar ninguna operación en Conexión de área local* 2 mientras los medios
estén desconectados.
No se puede realizar ninguna operación en Conexión de red Bluetooth mientras los medios
estén desconectados.

Adaptador de Ethernet Ethernet:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

Adaptador de LAN inalámbrica Conexión de área local* 1:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

Adaptador de LAN inalámbrica Conexión de área local* 2:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

Adaptador de LAN inalámbrica Wi-Fi:

    Sufijo DNS específico para la conexión. . :
    Dirección IPv6 . . . . . : 2002:c0a8:b02:10:3519:6273:d295:3e18
    Dirección IPv6 temporal. . . . . : 2002:c0a8:b02:10:c858:904d:6391:6d58
    Vínculo: dirección IPv6 local. . . : fe80::e3e7:355a:18e2:8736%17
    Puerta de enlace predeterminada . . . . : fe80::9e53:22ff:feff:ff78%17

Adaptador de Ethernet Conexión de red Bluetooth:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :
```

Figure 17. ipconfig/release again

- Finally, enter “*ipconfig /renew*” to again be allocated an IP address for your computer.

```

C:\Windows\System>ipconfig/renew

Configuración IP de Windows

No se puede realizar ninguna operación en Ethernet mientras los medios
estén desconectados.
No se puede realizar ninguna operación en Conexión de área local* 1 mientras los medios
estén desconectados.
No se puede realizar ninguna operación en Conexión de área local* 2 mientras los medios
estén desconectados.
No se puede realizar ninguna operación en Conexión de red Bluetooth mientras los medios
estén desconectados.

Adaptador de Ethernet Ethernet:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

Adaptador de LAN inalámbrica Conexión de área local* 1:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

Adaptador de LAN inalámbrica Conexión de área local* 2:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

Adaptador de LAN inalámbrica Wi-Fi:

    Sufijo DNS específico para la conexión. . :
    Dirección IPv6 . . . . . : 2002:c0a8:b02:10:3519:6273:d295:3e18
    Dirección IPv6 temporal. . . . . : 2002:c0a8:b02:10:c858:904d:6391:6d58
    Vínculo: dirección IPv6 local. . . : fe80::e3e7:355a:18e2:8736%17
    Dirección IPv4. . . . . : 192.168.0.119
    Máscara de subred . . . . . : 255.255.255.0
    Puerta de enlace predeterminada . . . . : fe80::9e53:22ff:fefb:ff78%17
                                              192.168.0.1

Adaptador de Ethernet Conexión de red Bluetooth:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

```

Figure 18. Last ipconfig /renew

- Stop Wireshark packet capture.

You should hand in a screen shot of the Command Prompt window. Whenever possible, when answering a question below, you should hand in a printout of the packet(s) within the trace that you used to answer the question asked. Annotate the printout to explain your answer. To print a packet, use *File→Print*, choose Selected packet only, choose Packet summary line, and select the minimum amount of packet detail that you need to answer the question. Answer the following questions [1, 2]:

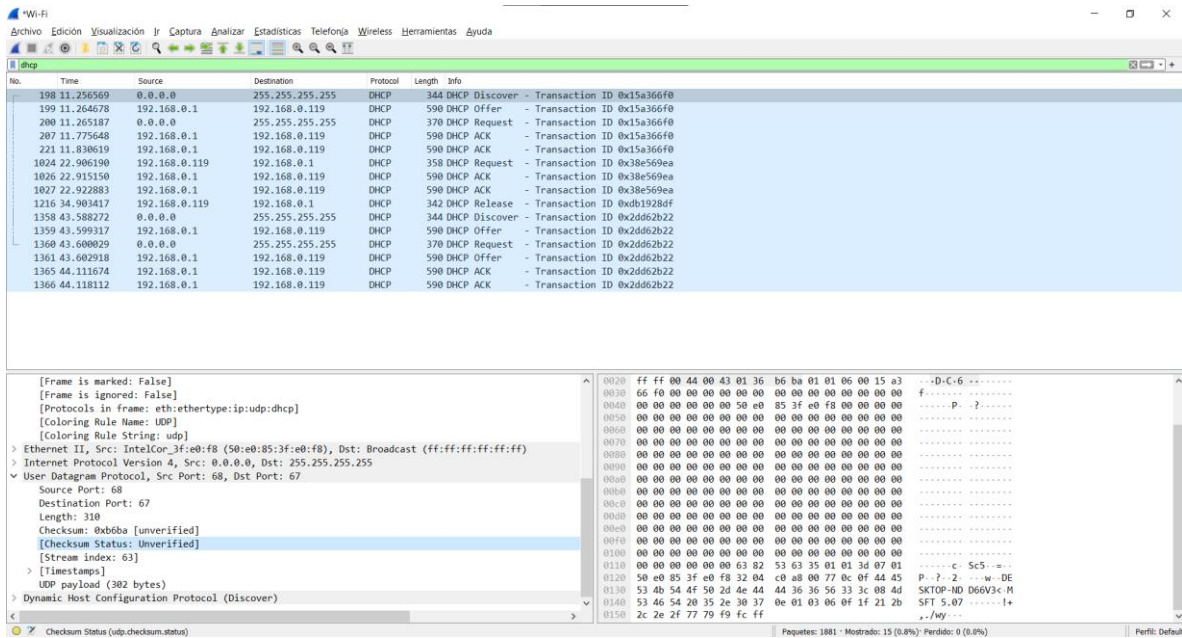


Figure 19. Wireshark after last ipconfig /renew

1. Are DHCP messages sent over UDP or TCP?

UDP

2. Draw a timing datagram illustrating the sequence of the first four-packet *Discover/Offer/Request/ACK* DHCP exchange between the client and server. For each packet, indicated the source and destination port numbers. Are the port numbers the same as in the example given in this lab assignment?

Discover (Source Port: 68, Destination Port: 67)

Offer (Source Port: 67, Destination Port: 68)

Request (Source Port: 68, Destination Port: 67)

Acknowledgment (Source Port: 67, Destination Port: 68)

3. What is the link-layer (e.g., Ethernet) address of your host?

The host is 50:e0:85:3f:e0:f8

4. What values in the DHCP discover message differentiate this message from the DHCP request message?

▾ Option: (53) DHCP Message Type (Discover) ▾ Length: 1 DHCP: Discover (1)	▾ Option: (53) DHCP Message Type (Offer) ▾ Length: 1 DHCP: Offer (2)
--	--

Figure 20. Discover/Offer Values

These messages differ in that one is intended for offer, while the other is meant for discovery.

5. What is the value of the Transaction-ID in each of the first four (*Discover/Offer/Request/ACK*) DHCP messages? What are the values of the Transaction-ID in the second set (*Request/ACK*) set of DHCP messages? What is the purpose of the Transaction-ID field?

For the first four packets the number of transactions is 0x15a366f0, the second number of transactions is 0x2dd3b22, this helps the server and client to secure that they are reaching the same thing shown in figure 19.

6. A host uses DHCP to obtain an IP address, among other things. But a host's IP address is not confirmed until the end of the four-message exchange! If the IP address is not set until the end of the four-message exchange, then what values are used in the IP datagrams in the four-message exchange? For each of the four DHCP messages (*Discover/Offer/Request/ACK* DHCP), indicate the source and destination IP addresses that are carried in the encapsulating IP datagram.

The IP datagrams in the four DHCP messages carry the source and destination IP addresses for the client and server.

7. What is the IP address of your DHCP server?

With the help of the ACK packet, we can know the IP address of the DHCP server which is 192.168.0.1

8. What IP address is the DHCP server offering to your host in the DHCP Offer message? Indicate which DHCP message contains the offered DHCP address.

The host address is 192.168.0.19 and the DHCP message contains option 53 with a length of 1 and option 2 in the DHCP.

9. Explain the purpose of the lease time. How long is the lease time in your experiment?

By default, the router line notifies the client of the message's destination, and the subnet mask line provides the client with the mask that should be used.

Conclusions

The experimentation with ICMP, IP, and DHCP protocols demonstrated their significance in network management and diagnostics. ICMP proved valuable for identifying connectivity issues, such as lost packets or unreachable destinations, and Traceroute provided insight into network routing paths. IP fragmentation illustrated the process of breaking down data for transmission across networks with varying MTU sizes. Lastly, the DHCP analysis highlighted its role in dynamic IP address allocation, simplifying the management of large networks. This study contributes to a better understanding of network mechanisms and underscores the importance of these technologies in data communication.

Bibliography

Droms, R. (1997). RFC 2131 - Dynamic Host Configuration Protocol. Internet Engineering Task Force. Retrieved from <https://tools.ietf.org/html/rfc2131>

Kurose, J. F., & Ross, K. W. (2008). "Computer Networking: A Top-Down Approach." 4th edition. Addison-Wesley.

PingPlotter. (n.d.). PingPlotter Documentation. Retrieved from <https://www.pingplotter.com>

Postel, J. (1981). RFC 792 - Internet Control Message Protocol. Internet Engineering Task Force. Retrieved from <https://tools.ietf.org/html/rfc792>

Wikipedia contributors. (2024). Dynamic Host Configuration Protocol. In "Wikipedia, The Free Encyclopedia". Retrieved from https://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol

Wikipedia contributors. (2024). Internet Control Message Protocol. In "Wikipedia, The Free Encyclopedia". Retrieved from https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol

Wikipedia contributors. (2024). IP fragmentation. In "Wikipedia, The Free Encyclopedia". Retrieved from https://en.wikipedia.org/wiki/IP_fragmentation

Wireshark. (n.d.). Wireshark Documentation. Retrieved from <https://www.wireshark.org>