Lab 5: DNS & HTTP

ECSE 308 Introduction to Communication Systems and Networks

Chenyi Xu 260948311

Yongru Pan 261001758

Abstract: The lab is divided into three parts. The objective of the first part is to use Wireshark to investigate the Domina Name System(DNS) protocol from the DNS client's standpoint. In the second part of the lab, the focus was to use Wireshark to investigate the UDP structure and protocol properties. Last but not least, the third part concentrate on using Wireshark to collect traces and investigates the different aspects of the HTTP protocol operation.

Introduction

This lab explores the core concepts of Domain Name System (DNS), User Datagram Protocol (UDP), and Hypertext Transfer Protocol (HTTP), utilizing tools like nslookup and Wireshark to capture and analyze packet data. By performing a series of queries and analyzing responses, the lab investigates how DNS resolves domain names into IP addresses, the mechanics of UDP as a transport layer protocol for DNS, and the intricacies of HTTP for web communication. Through these exercises, key networking principles such as packet structure, protocol efficiency, and system reliability are examined, providing a comprehensive understanding of how these foundational internet protocols interact to deliver seamless user experiences.

Analysis

Part I: Domain Name System (DNS)

Q1: Use nslookup to determine the IP address of www.cbc.ca. What is the IP address of this web server?

The IP address is 184.25.129.124

Q2: Use nslookup to determine the authoritative DNS servers for McGill University.

The authoritative DNS is pirns1.mcgill.ca, with the IP address 132.206.44.21.

Q3: Run nslookup to obtain the IP address of www.wikipedia.org by sending a query to 8.8.4.4 which is the IP address of the google public DNS server.
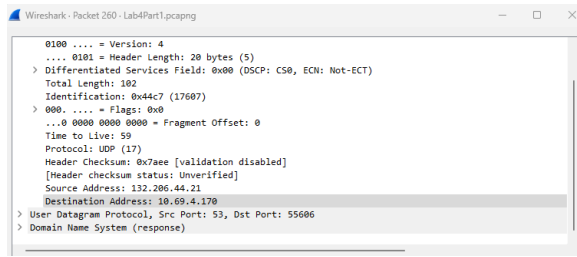

The IPv6 address is 2620:0:861:ed1a::1

The IPv4 address is 208.80.154.224

Q4: What are the destination port number for the DNS query message and the source port number of the DNS response message?
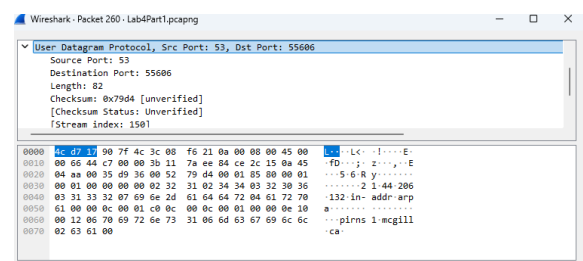
The destination port number is 55606.

Q5: What is the destination IP address of the DNS query? Is this the IP address of your default local DNS server?

The destination IP address is 10.69.4.170

Q6: Examine the DNS query. What is the "Type" of the DNS query? What does this "Type" mean? What are the other values for this field?

The "Type" of the DNS query is A, AAAA, SOA, and CNAME. A signifies the queries for IPv4 address of the host. AAAA signifies the queries for IPv6 address of the host. SOA signifies the query asking for the Start of Authority record. CNAME signifies the query asking for the alias of a domain. Other values include MX, NS, PTR, and TXT.

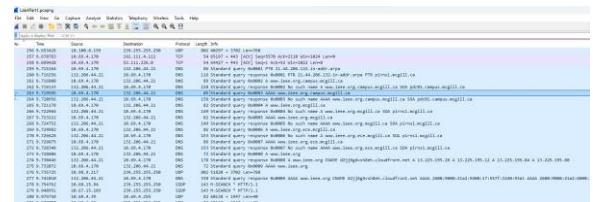*Q7: Which bit in the "Flags" field indicates that the message is a query or a response?*
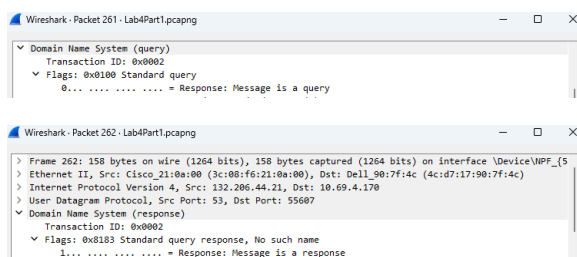
The QR bit indicates this. If QR = 0, the message is a query. If QR = 1: indicates the message is a response.

Q8: Which field of the response message contains the IP address of www.ieee.org?

In the "Answer" field, you can find the IP address of www.ieee.org.

Q9: Provide a screenshot.

Q10: What is the destination IP address of the DNS query? What does this address correspond to?

The destination IP address is "132.206.44.21". This is the DNS server of mcgill.

Q 11: Determine the "Type" of DNS query. What is the authoritative name server of www.wireshark.org. What is the role of an authoritative name server?
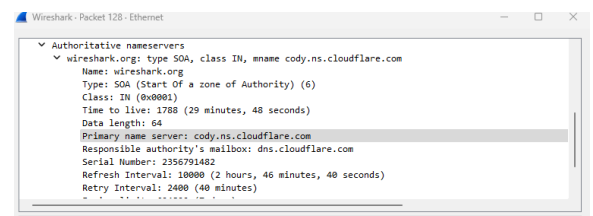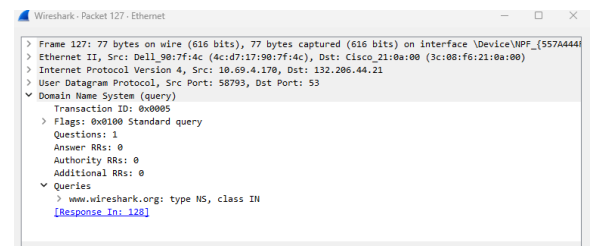
The type of the DNS query is NS.

The authoritative nameservers is cody.ns.cloudflare.com

Role of an Authoritative Name Server is to hold the authoritative DNS records for this domain and provides definitive answers to DNS queries related to it.

Q12: Provide a screenshot.

Q13: What are the destination IP addresses for the two DNS queries? What do these IP addresses correspond to?

It can be seen from the screenshot that the two queries each have the following destination address: 192.168.18.1, which is the local router or DNS server address. The other one is 8.8.8.8, which is Google's public DNS server address.

Q 14: What IP addresses are returned by these two queries? Do they return the same IP addresses for www.google.com? Explain your answer.

I have chosen packet 97 and packet 210 because packet 97's source is 192.168.18.1 and packet 210's source is 8.8.8.8, and they end up having the same IP address for www.google.com. The reason behind this can be that they both access the same cached record, or that they have received the same response from Google's authoritative DNS.

Q15: Provide a screen shot.

Part II: User Datagram Protocol (UDP)

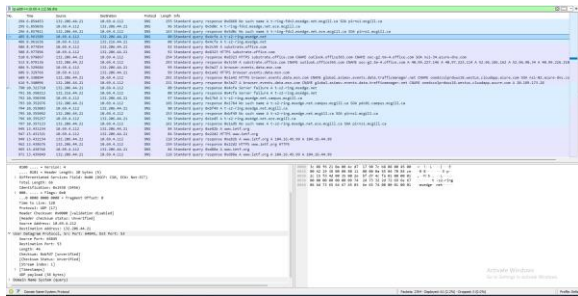Ip address = 10.69.4.112

Q16: What transport layer protocol is used to transfer the DNS query and the response message?

UDP

Q17: To set up the connection, how many UDP datagrams are exchanged between your computer and the server? Explain your answer.

61 UDP datagrams are exchanged between the computer and the server. A filter is used to filter out all the DNS. According to Wireshark, 61 UDP datagrams are exchanged.

Q18: Select the first DNS packet in your trace. From this packet, determine the header fields of UDP.

Destination, Source, Types, Version and header length, Differentiated Service Fields, Total length, Identification , Flags, Fragment offset, Time to live, Protocol, Header Checksum, Source address, Destination address, Source port, destination port, length, checksum, UDP payload
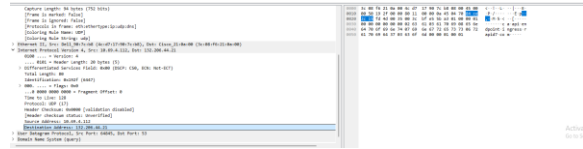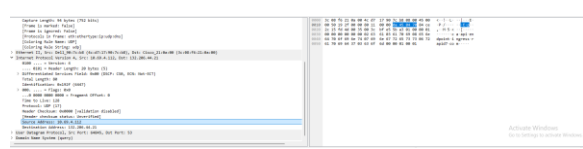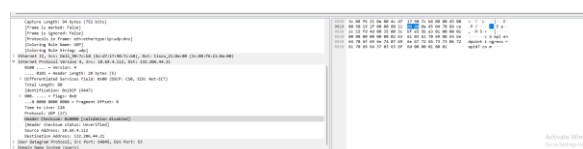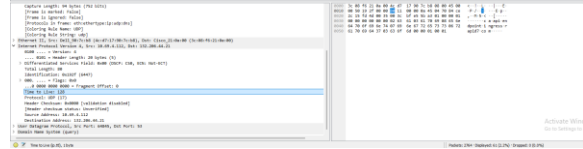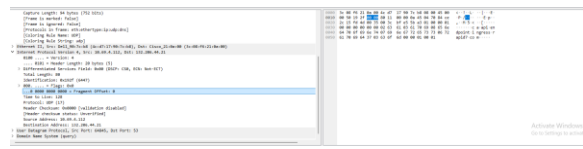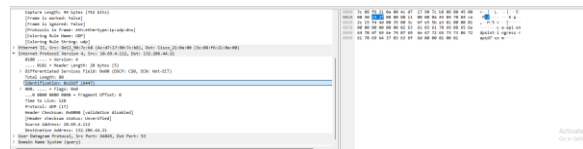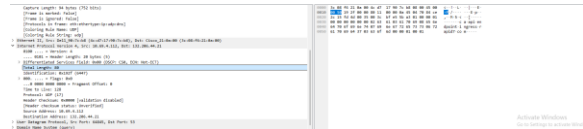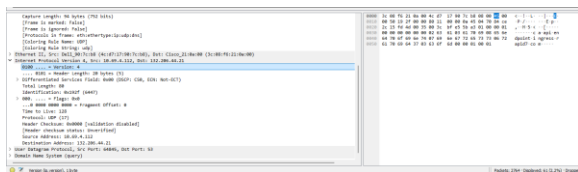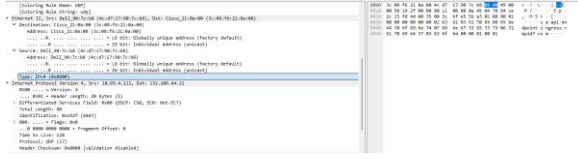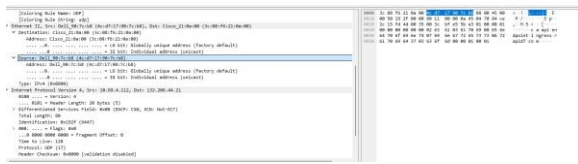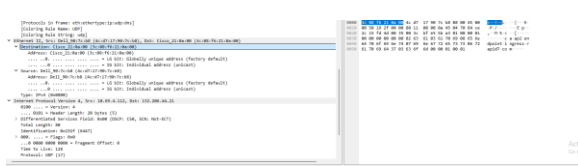
Q19: By consulting the displayed information in Wireshark's packet content field for the first DNS message, determine the length(in bytes) of each of the UDP header fields.

Destination 6 bytes

Source 6 bytes

Types: 2 bytes

IP Version and header length: 1 byte

Differentiated Service Fields: 1 byte

Total length: 2 bytes

Identification: 2 bytes

Fragment offset: 2 byte

Flags: 1 byte

Time to live: 1 byte

Protocol: 1 byte

Header Checksum: 2 bytes

Source address: 4 bytes

Destination adresse: 4 bytes

Source port: 2 bytes

Destination port: 2 bytes

Length: 2 bytes

Checksum: 2 bytes

UDP payload: 52 bytes

Q20: The value in the Length field indicates the length of what? Verify your claim with your captured UDP packet.

It specifies the size of the entire UDP datagram. The length field here shows a value of 60 bytes. 8 bytes are used for the UDP header(source, destination, length, checksum) and 52 bytes are used for the UDP payload.

Q21: What is the maximum number of bytes that can be included in a UDP payload? (Hint: The answer to this question can be determined by your previous answer).

Since length occupies 2 bytes. The largest possible number is ff ff which is 65535. 8 bytes are used for the header field. So the maximum number of bytes that can be included in a UDP pauload is ff ff -

Q22: What is the largest possible source port number?

Source port occupies 2 bytes. The largest possible source port number would be ff ff which is 65535

Q23: Determine whether a checksum is provided for the first DNS message or not. What is the usage of this field?

Yes, it is provided. It ensures that the data in the UDP header and payload arrives without corruption.

Q24: Determine the destination port number for the DNS query message and the source port number of the DNS response. What is the relationship between the two? Which port number is a well-known port number?

Destination port number: 53

Source port number: 64845

Port 53 is a well-known port number, as it is reserved for the DNS protocol. It is used by DNS servers to listen for incoming queries from clients.

The client initiates the query, using the dynamic source port (64845 in this case) and sends the request to port 53 on the DNS server. The server replies using port 53 as its source port (matching the client's query destination port) and sends the response to the dynamic port (64845) that was originally used as the client's source port.

Q25: List two other well-known port numbers used by UDP

Port 67, Port 123

Q26: Determine the IP address of your local DNS sever (use ipcongif). Is it the same as destination IP address of the DNS query?

Destination address of the DNS query: 132.206.44.21. It is the same as the destination IP address of the local DNS server.

Q27: Examine the DNS response message. How many answers are provided in this message? What do each of these answers contain?

5 answers are provided in the message. Each of those answers contain Name, Type, Class, Time to live, Data Length, CNAME

Q28: By checking the trace, determine whether UDP is a reliable protocol or not. Explain your answer.

The trace demonstrates that UDP is unreliable because it does not guarantee packet delivery or handle retransmissions, as evidenced by the repeated DNS queries. It is the best trying mechanism. The DNS queries was sent repeatedly indicating the transmission failed. Hence not reliable.

Q29: Why does DNS use UDP services?

It uses UDP due to its speed, efficiency, and suitability. It is a connectionless protocol. No need to establish connection which reduces latency, allowing DNS to resolve queries more quickly. UDP has low overhead. It is ideal for DNS, where most queries and responses are small and don't need the reliability mechanism of TCP.

Part III: Hyper-Text Transfer Protocol (HTTP)

Q30. What HTTP request method is used to retrieve the HTML file?

The HTTP request method used to retrieve the HTML file is **GET.**

Q31. What is the URI of the requested file?
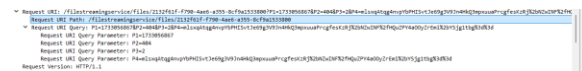
The URI is as shown in the screenshot

Q32. What HTTP version is your browser running? What are the other versions of HTTP?
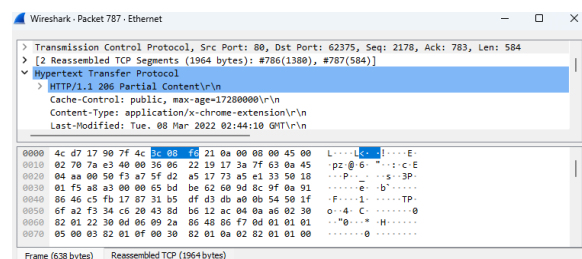
The HTTP version is HTTP/1.1

The other versions of HTTP include HTTP/0.9, HTTP/1.0, HTTP/1.1, HTTP/2, and HTTP/3.

Q33. What languages does your browser accept for response?
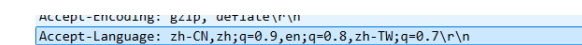
These are the accepted languages.

Q 34. What is the IP address of your computer?

The IP of my computer is 10.69.4.170

Q35. What is the server's IP address?

The IP address of the server is 34.223.124.45

36. What is the relationship between source and destination IP addresses of the first GET and the source and destination IP addresses of the first Response?

The first GET's source and destination IP addresses is the first response's destination and source IP addresses.

37. What is the status code of the first response message? What does this code indicate? What code is returned if the requested file cannot be found on the server?

The status code is 200. It indicates that the request is successful, and that the content has delivered. If the requested file cannot be found the code will be 404.

38. When was the last time that the received HTML file was modified at the server?
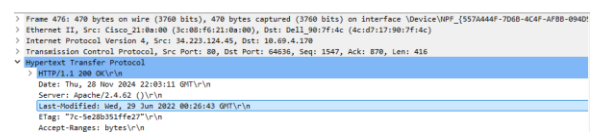
Last-Modified: Wed, 29 Jun 2022 00:26:43 GMT\r\n

39. What is the size of the content that is returned to your browser?
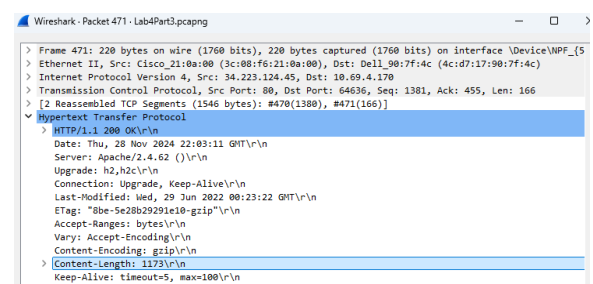
The size is 1173 Bytes.

Q40: How many HTTP GET request messages are sent by your web browser?

Two GET are sent by my browser.

41. By inspecting the entire trace, determine the number of packets that contain HTTP header. Explain your answer.

Four packets contain the HTTP header because it is being filtered out by http.

Q42. How many TCP segments are transmitted to your computer? Why multiple segments are required to retrieve this single HTML file?

9 TCP segments are transmitted to my computer.

The reason behind needing multiple segments is that the TCP handshake and connection management (e.g., SYN, ACK, FIN packets) and segmentation of HTTP headers and body based on the Maximum Segment Size (MSS).

43. Determine the length of these TCP segments. Do they have the same size? Explain your answer.

The lengths are as follows:

66 bytes for packets 433 and 456

60 bytes for packets 468 and 486

1434 bytes for packets 470

54 bytes for packets 472, 479 and 687

They do not have the same size. The size of the packets depends on the purpose of the packet. If they are control packets, they tend t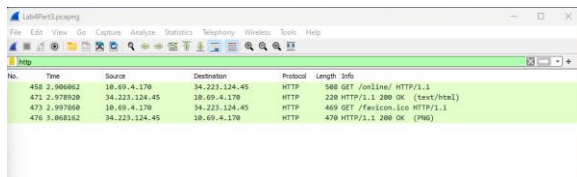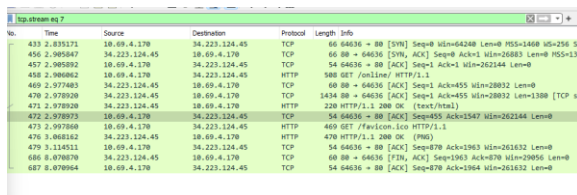o be small like those with a length of 54, 60, and 66. If they are data packets, they tend to be bigger like the one with the 1434 bytes.

Q44. Which message and what field in that message indicate that the server was able to process the request successfully?

The 200 ok in the status code field tells us that indicates the server was able to process the request successfully.

Q45. What is the status code of the first response message?

200

Q46. What is the value of the content size of the first response message?

Q47. What is the etag of the first response message?

Q48. What is the application of etag in conditional HTTP request? Which line in the second response contains the etag value of the first response?

ETags optimize HTTP interactions by enabling conditional requests that validate cached resources, prevent unnecessary updates, and avoid redundant data transfers. This helps improve web performance and ensures data consistency during interactions between the client and server. It is found under Hypertext transfer protocol section.

Q49. Which HTTP Get contains the IF-MODIFIED_SINCE line? What is the usage of this field?

```
625 7.959996    34.223.124.45    10.69.4.112    HTTP    220 HTTP/1.1 200 OK  (text/html)
```

```
Last-Modified: Wed, 29 Jun 2022 00:23:22 GMT\r\n
```

It is used in HTTP GET requests to optimize the retrieval of resources by checking if they have been modified since a specific date and time.

Q50. What is the status code of the second response message? What does this code mean?

200, it is a three digit number that is included in the server's response to a client's request. It indicates the result of the request and provides information about how client should handle the request. 200 represent that the request was successful, and the server is returning the requested resource.

Q51. What is the content length of the second response? Explain.

```
File Data: 124 bytes
```

This information does not contain any header. It's the smallest and content only.

Q52. How many HTTP GET Requests are sent by your web browser?

2 request

Q53. What is the content type of each response message?

Text/html, and PNG

Q54. Did your browser download the two images serially or in parallel? Explain. What are the pros and cons of each approach.

There is only 1 image in this web. The browser can download image both serially and in parallel.

Serial:

Pros: Lower resource usage, images downloaded in a predictable order, reduce the risk of overwhelming the network

Cons: Slower total completion time, inefficient use of resources

Parallel:

Pros: faster overall download time, better resource utilization, resilient to slow connections

Cons: higher resource usage, potential server overload, complex error handling, image may not download in the desired order.

Q55. Has the HTTP used persistent or non-persistent connection? Explain your answer.

HTTP use persistent connection. TCP connection reused for multiple requests. It is reflected through source and destination ports on Wireshark.

Q56. What is the requested URL in the frame #101? What HTTP field contains the username and password information? What are the submitted values for the username and password?
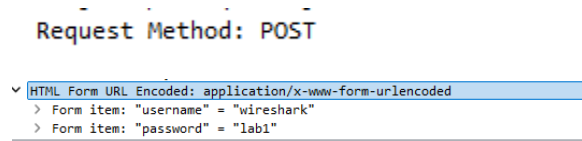
Request URI Query contains the information

```
Request Method: GET
```

```
[Full request URI: http://192.168.0.2:8000/lab1Ex5a.html?username=wireshark&password=lab1]

∨ Request URI Query: username=wireshark&password=lab1
    Request URI Query Parameter: username=wireshark
    Request URI Query Parameter: password=lab1
```
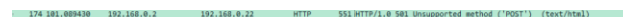
Q57. What HTTP request method is used in the frame #172? What HTTP field contains the username and password information? Explain the difference between this request method and the GET method.

```
Request Method: POST
```

```
∨ HTML Form URL Encoded: application/x-www-form-urlencoded
  > Form item: "username" = "wireshark"
  > Form item: "password" = "lab1"
```

GET fetch data from the server. Data located in URL query string. It is safe. Responses can be cached. Data is visible in URL. Data size is limited by URL length.

POST send data to the server. Data locates in HTTP request body. It is not safe. Responses typically not cached. Data is hidden in request body. There is no practical data size limit.

Q58. What is the status code of the frame #174? What is the description of this code?

```
174 181.089430  192.168.0.2    192.168.0.22    HTTP    551 HTTP/1.0 501 Unsupported method ('POST')  (text/html)
```

501 Not Implemented indicates that the server lacks the ability to perform the requested operation because the required functionality is not implemented.

Conclusion

This lab effectively demonstrated the functionality and interaction of DNS, UDP, and HTTP protocols within network communication. Through hands-on analysis, we observed how DNS resolves domain names to IP addresses, facilitated by UDP for its efficiency and speed. Wireshark packet traces highlighted the headers, fields, and payloads involved in these exchanges, reinforcing the concepts of protocol design and reliability. In the HTTP section, we explored GET and POST methods, status codes, and resource transfer efficiency. This lab underscored the significance of these protocols in enabling reliable and efficient internet communication while exposing their respective strengths, limitations, and trade-offs in real-world applications.