

ECSE 308:
Introduction to
Communication Systems and Networks

L5N2: DNS & HTTP

Part 1: Domain Name System (DNS)

Part 2: User Datagram Protocol (UDP)

Part 3: Hyper-Text Transfer Protocol (HTTP)

L5N2: DNS & HTTP

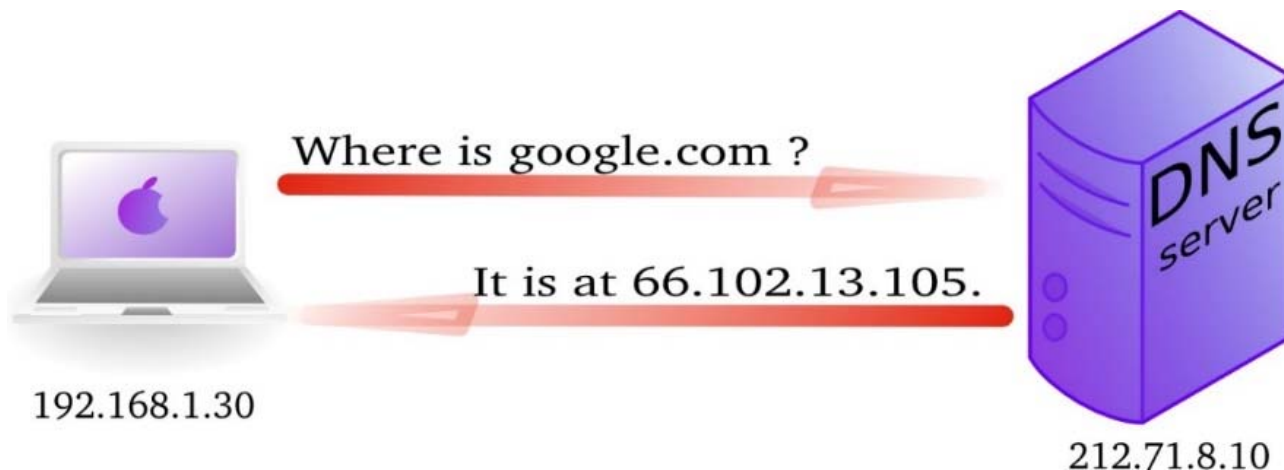
Part 1: Domain Name System (DNS)

Objectives: Use Wireshark to investigate the Domain Name System (DNS) protocol from the DNS client's standpoint

- DNS query/response structure
- Authoritative name servers
- DNS load balancing

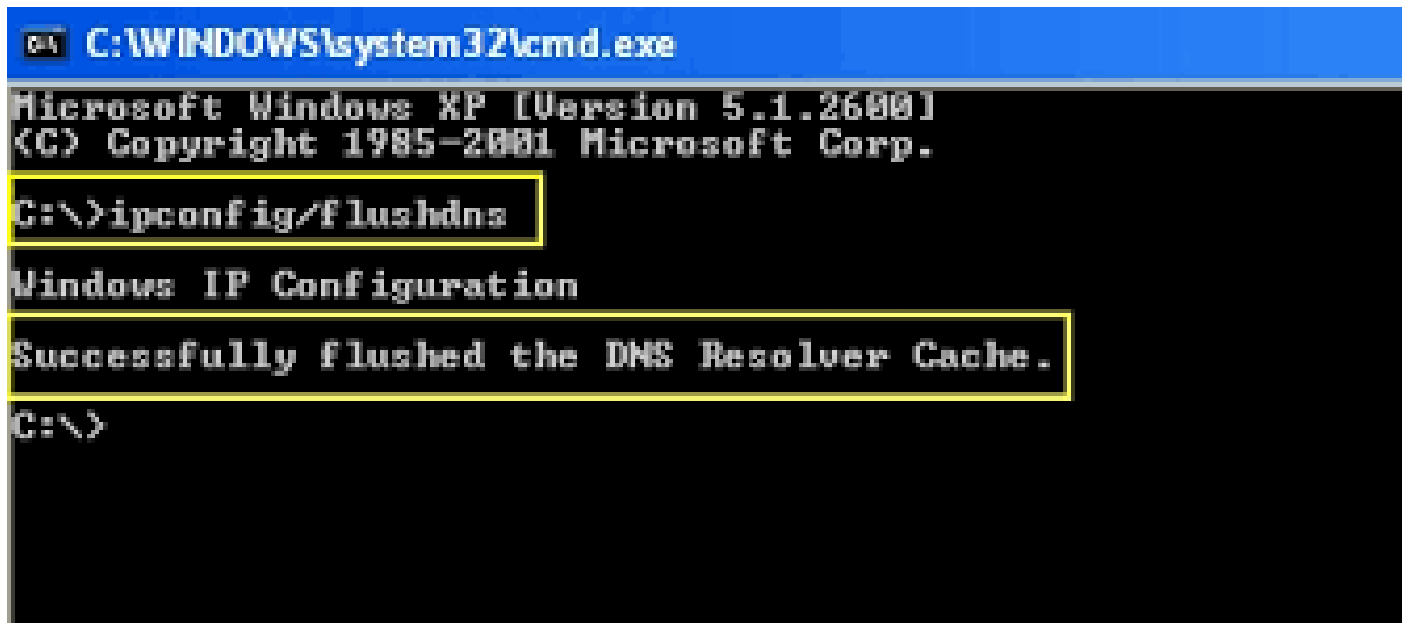
Domain Name System (DNS): Overview

- The DNS translates hostnames to IP addresses needed for locating and identifying computer services and devices with the underlying network protocols.
- The client's role in the DNS is relatively simple; a client sends a query to its local DNS server. If the local DNS server can resolve query by using locally cached information, the query is answered and the process is completed. Otherwise, the resolution process continues with the client querying a DNS server to resolve the name.



ipconfig

- **ipconfig:** *ipconfig* (for Windows) and *ifconfig* (for Linux/Unix) are among the most useful little utilities in your host, especially for debugging network issues. Here we will only describe *ipconfig*, although the Linux/Unix *ifconfig* is very similar. *ipconfig* can be used to show your current TCP/IP information, including your address, DNS server addresses, adapter type and so on.
- **Running ipconfig:**
 - Windows: open the Command Prompt and run *ipconfig* on the command line.
 - Linux/Unix: type the *ifconfig* command on the command line.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\>ipconfig/flushdns

Windows IP Configuration

Successfully flushed the DNS Resolver Cache.

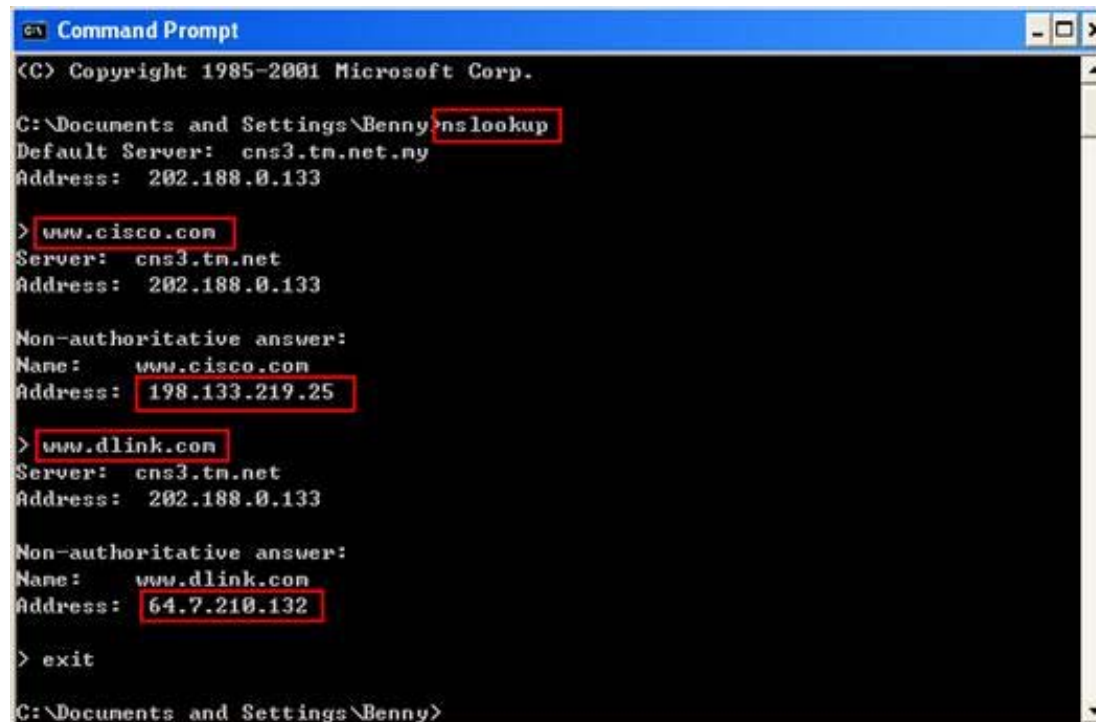
C:\>
```

ipconfig commands

- `ipconfig /all`
 - Displays all current TCP/IP network configuration values
- `ipconfig /displaydns`
 - A host can cache DNS records it recently obtained. To see these cached records, we can use this command. Each entry shows the remaining Time to Live (TTL) in seconds.
- `ipconfig /flushdns`
 - This command clears all entries of the DNS cache and reloads the entries from the hosts file.

nslookup

- This tool allows the user to query any specified DNS server for a DNS record.
- 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.
- Running *nslookup*:
 - Windows: open the Command Prompt and run *nslookup* on the command line.
 - Linux/Unix: just type the *nslookup* command on the command line.



```
Command Prompt
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Benny>nslookup
Default Server: cns3.tn.net.ny
Address: 202.188.0.133

> www.cisco.com
Server: cns3.tn.net
Address: 202.188.0.133

Non-authoritative answer:
Name: www.cisco.com
Address: 198.133.219.25

> www.dlink.com
Server: cns3.tn.net
Address: 202.188.0.133

Non-authoritative answer:
Name: www.dlink.com
Address: 64.7.210.132

> exit

C:\Documents and Settings\Benny>
```

nslookup commands

- *nslookup*
 - This command identifies which DNS server the computer is currently configured to use for its DNS lookups.
- *nslookup "hostname"*
 - 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 "hostname".
- *nslookup -type=NS "hostname"*
 - This command specifies a DNS name server for the given hostname.
- *nslookup "hostname" "dns server"*
 - This command indicates that the query should be sent to the given DNS server rather than to the default DNS server.
- *nslookup -option1 -option2 "hostname" "dns-server"*
 - This is the general syntax of *nslookup* commands. As we have seen in the above, *nslookup* can be run with zero, one, two or more options. Furthermore, the dns-server is optional as well; if it is not given, the query is sent to the default local DNS server.

Questions

1. Use *nslookup* to determine the IP address of www.cbc.ca. What is the IP address of this web server?
2. Use *nslookup* to determine the authoritative DNS servers for McGill University.
3. 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.

DNS queries and responses - Instructions

- Start packet capture.
- Run Command Prompt and enter the command:
`nslookup www.ietf.org`
- Stop packet capture.

DNS queries & responses: example **screenshot**

The screenshot displays a Wireshark capture of DNS traffic. The packet list pane shows a series of DNS queries and responses. The packet details pane shows the structure of a DNS query packet. The packet bytes pane shows the raw hex and ASCII data.

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-----------|---------------|---------------|----------|--------|--|
| 39 | 14.695652 | 192.168.0.105 | 192.168.0.1 | DNS | 89 | Standard query 0x0007 AAAA www.ietf.org. campus.mcgill.ca |
| 40 | 14.697431 | 192.168.0.1 | 192.168.0.105 | DNS | 89 | Standard query response 0x0007 No such name AAAA www.ietf.org. campus.mcgill.ca |
| 41 | 14.697862 | 192.168.0.105 | 192.168.0.1 | DNS | 78 | Standard query 0x0008 A www.ietf.org. local |
| 42 | 14.699193 | 192.168.0.1 | 192.168.0.105 | DNS | 78 | Standard query response 0x0008 No such name A www.ietf.org. local |
| 43 | 14.699591 | 192.168.0.105 | 192.168.0.1 | DNS | 78 | Standard query 0x0009 AAAA www.ietf.org. local |
| 44 | 14.701115 | 192.168.0.1 | 192.168.0.105 | DNS | 78 | Standard query response 0x0009 No such name AAAA www.ietf.org. local |
| 45 | 14.701536 | 192.168.0.105 | 192.168.0.1 | DNS | 72 | Standard query 0x000a A www.ietf.org |
| 46 | 14.713664 | 192.168.0.1 | 192.168.0.105 | DNS | 159 | Standard query response 0x000a A www.ietf.org CNAME www.ietf.org.edgekey.net CNAME e1630.c.akamaiedge.net A... |
| 47 | 14.716702 | 192.168.0.105 | 192.168.0.1 | DNS | 72 | Standard query 0x000b AAAA www.ietf.org |
| 48 | 14.719345 | 192.168.0.1 | 192.168.0.105 | DNS | 146 | Standard query response 0x000b AAAA www.ietf.org CNAME www.ietf.org.edgekey.net CNAME e1630.c.akamaiedge.net |

Note: In this screenshot, we can see that *nslookup* actually sent several DNS queries and received their DNS responses. However, in answering the questions in the following page, only focus on the last query and ignore the rest sets of queries/responses, as they are specific to *nslookup* and are not normally generated by standard Internet applications.

Questions

4. What are the destination port number for the DNS query message and the source port number of the DNS response message?
5. What is the destination IP address of the DNS query? Is this the IP address of your default local DNS server?
6. 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?
7. Which bit in the “Flags” field indicates that the message is a query or a response?
8. Which field of the response message contains the IP address of www.ieee.org?
9. Provide a screenshot.

III. Authoritative name servers - Instructions

- Again repeat the previous instructions, but instead use the command:

`nslookup -type=NS www.wireshark.org`

III. Questions

10. What is the destination IP address of the DNS query? What does this address correspond to?
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?
12. Provide a screenshot.

DNS Load Balancing - Instruction

- Start Wireshark packet capture.
- Open Command Prompt and use an appropriate *ipconfig* to clear your DNS cache.
- In the Command Prompt, enter the command:
`nslookup www.google.com`
- After getting the results for the previous command, run another command
“nslookup www.google.com 8.8.8.8”.
- Stop Wireshark packet capture.

Questions

13. What are the destination IP addresses for the two DNS queries? What do these IP addresses correspond to?
14. What IP addresses are returned by these two queries? Do they return the same IP addresses for www.google.com ? Explain your answer.
15. Provide a screen shot.

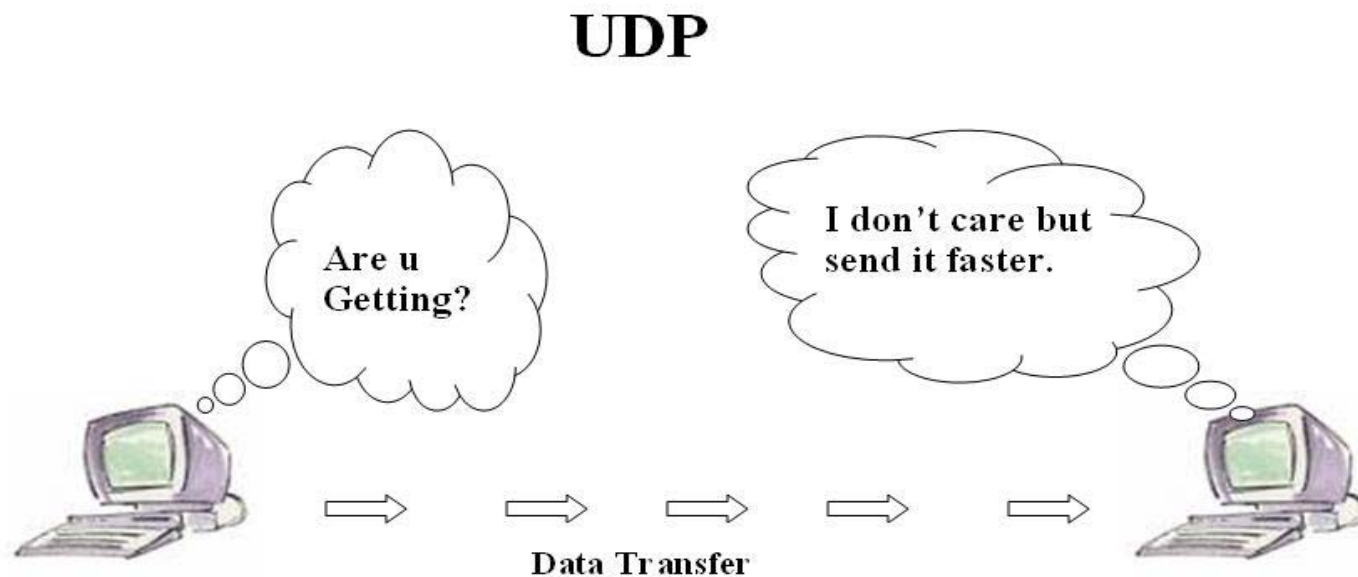
L5N2: DNS & HTTP

Part 2: User Datagram Protocol (UDP)

Objectives: Use Wireshark to investigate the UDP structure and protocol properties.

User Datagram Protocol (UDP): Objective

- In this experiment, we will focus on the UDP, which is one of the transport level protocol of the TCP/IP model. UDP is a connection-less protocol meaning that no connection-establishment and connection-terminations are used in this protocol. To analyze UDP headers, we need to run an application that uses the UDP service. A good candidate is DNS.



UDP/DNS: Instructions

- Use *ipconfig* to clear the DNS cache in your host.
- Open your browser and clear your browser cache. (With Internet Explorer, go to Tools menu and select Internet Options; then in the General tab select Delete Files.)
- Use *ipconfig* to obtain your IP address.
- Open up the Wireshark and enter “ip.addr == your_IP_address” into the filter. This filter removes all packets that neither originate nor are destined to your host.
- Open up the Wireshark and start packet capture.
- With your browser, visit the Web page: <http://www.ietf.org>
- Stop packet capture.

UDP/DNS: Questions (1/2)

16. What transport layer protocol is used to transfer the DNS query and the response message?
17. To setup the connection, how many UDP datagrams are exchanged between your computer and the server? Explain your answer.
18. Select the first DNS packet in your trace. From this packet, determine the header fields of UDP.
19. 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.
20. The value in the Length field indicates the length of what? Verify your claim with your captured UDP packet.
21. 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)
22. What is the largest possible source port number?

UDP/DNS: Questions (2/2)

23. Determine whether a checksum is provided for the first DNS message or not. What is the usage of this field?
24. 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?
25. List two other well-known port numbers used by UDP.
26. Determine the IP address of your local DNS server (use ipconfig). Is it the same as destination IP address of the DNS query?
27. Examine the DNS response message. How many “answers” are provided in this message? What do each of these answers contain?
28. By checking the trace, determine whether UDP is a reliable protocol or not. Explain your answer.
29. Why does DNS use UDP services?

L5N2: DNS & HTTP

Part 3: Hyper-Text Transfer Protocol (HTTP)

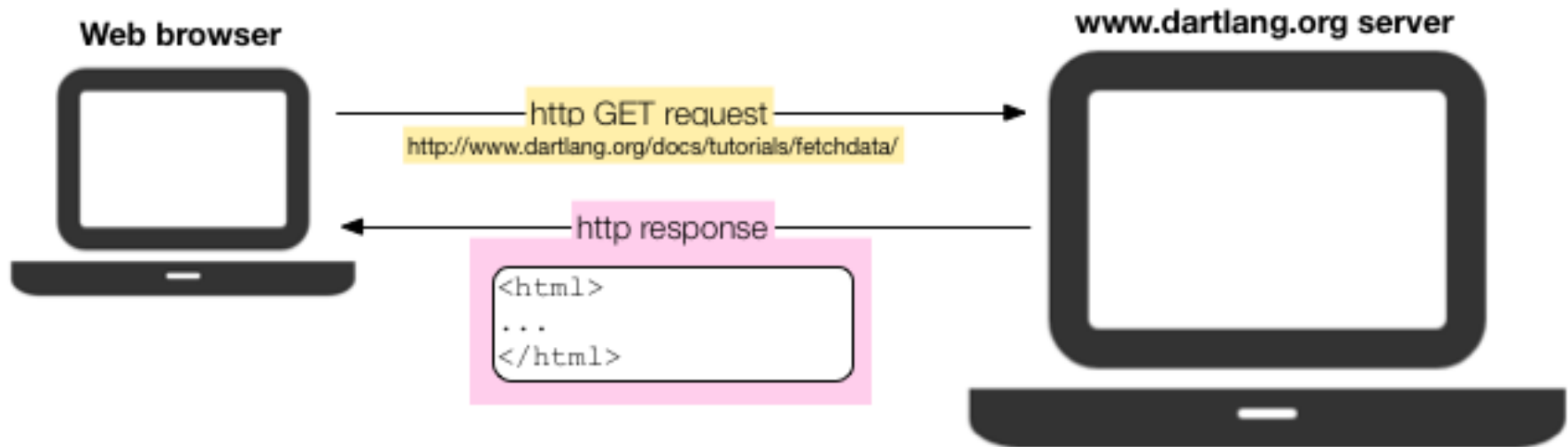
Objectives: Use Wireshark to collect traces and investigate the different aspects of the HTTP protocol operation:

- Simple HTTP GET request/response
- Long HTTP response
- HTTP caching mechanism
- HTML pages with embedded objects
- HTTP request methods

Simple HTTP GET request/response

- Overview:

We will investigate the basic HTTP GET request/response interaction by retrieving a simple HTML file which is very short, and has no embedded objects.



HTTP GET request/response: Instructions

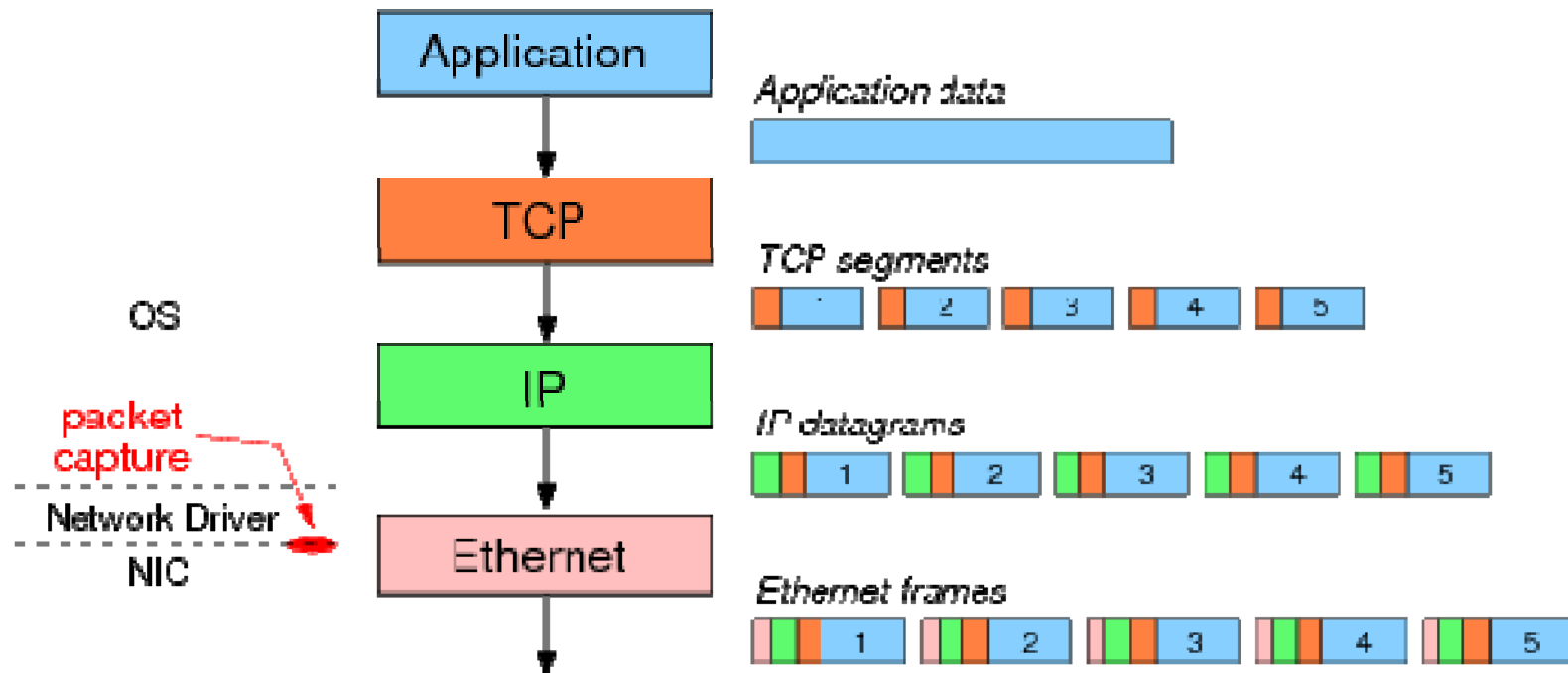
- Start up your web browser.
- Open up the Wireshark. In the filter field, type “http” (without the quotation marks), so that only captured HTTP messages will be displayed later in the packet-listing window.
Note: After you have changed the expression in the filter input box, do not forget to press the Apply button (or the Enter/Return key twice), to apply this filter string to the displayed trace file.
- Start Wireshark packet capture.
- Go back to your web browser and enter:
<http://www.info308a.ece.mcgill.ca/Lab1Ex1.html>
- Stop Wireshark packet capture.

HTTP GET request/response: Questions

30. What HTTP request method is used to retrieve the HTML file?
31. What is the URI of the requested file?
32. What HTTP version is your browser running? What are the other versions of HTTP?
33. What languages does your browser accept for response?
34. What is the IP address of your computer?
35. What is the server's IP address?
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?
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?
38. When was the last time that the received HTML file was modified at the server?
39. What is the size of the content that is returned to your browser?

Long HTTP response: Overview

- In the previous exercise, we retrieved a short HTML file. Here, we will see what happens when we download a long HTML file.



Long HTTP response: Instruction

- Start up your web browser.
- Open up the Wireshark.
- Start Wireshark packet capture.
- Enter the following URL into your browser
<http://www.info308a.ece.mcgill.ca/Lab1Ex2.html>
- Stop Wireshark packet capture, and type “http” in the filter field.

Long HTTP response: Questions

- 40. How many HTTP GET request messages are sent by your web browser?
- 41. By inspecting the entire trace, determine the number of packets that contain HTTP header. Explain your answer.
- 42. How many TCP segments are transmitted to your computer? Why multiple segments are required to retrieve this single HTML file?
- 43. Determine the length of these TCP segments. Do they have the same size? Explain your answer.
- 44. Which message and what field in that message indicate that the server was able to process the request successfully?

HTTP caching mechanism

- Overview:

In this exercise, we will focus on the caching mechanism of the HTTP protocol. Most web browsers keep the recently retrieved HTTP objects in their cache memory. When they receive a request to retrieve an HTTP object, they first check whether the object is cached or not. If the object exists in the cache memory, a conditional GET request is sent to the server. The server sends the object if it is modified, otherwise it sends a “Not Modified” response.

- Note:

Before performing the instructions, make sure your browser's cache is empty.

- Firefox: select Tools->Clear Recent History and check the Cache box
- Internet Explorer: select Tools->Internet Options->Delete File; these actions will remove cached files from your browser's cache.

HTTP caching: Instruction

- Start up your web browser, and make sure your browser's cache is cleared.
- Open up the Wireshark.
- Start Wireshark packet capture.
- Enter the following URL into your browser
<http://www.info308a.ece.mcgill.ca/Lab1Ex3.html>
- Quickly enter the same URL into your browser again (or use the refresh button on your browser)
- Stop Wireshark packet capture, and type “http” in the filter field.

HTTP caching: Questions

45. What is the status code of the first response message?
46. What is the value of the content size of the first response message?
47. What is the etag (identity tag) of the first response message?
48. What is the application of etag in conditional HTTP request? Which line in the second response contains the etag value of the first response?
49. Which HTTP GET contains the “IF-MODIFIED-SINCE” line? What is the usage of this field?
50. What is the status code of the second response message? What does this code mean?
51. What is the content length of the second response? Explain.

Retrieving a web page with embedded objects

■ Overview:

In this experiment, we will retrieve an HTML file with embedded objects, i.e., a file that includes other objects (in the example below, image files) that are stored on another server(s).

```
<object width="320" height="320"><param  
value="http://popplet.com/app/Popplet_Alpha.swf?page_id=580600&e  
m=1" name="movie"></param><param value="true"  
name="allowFullScreen"></param><param value="always"  
name="allowscriptaccess"></param><embed  
src="http://popplet.com/app/Popplet_Alpha.swf?page_id=580600&em=  
1" height="320" width="320" allowfullscreen="false"  
allowscriptaccess="always" type="application/x-shockwave-  
flash"></embed></object>
```

Retrieving a web page: Instruction

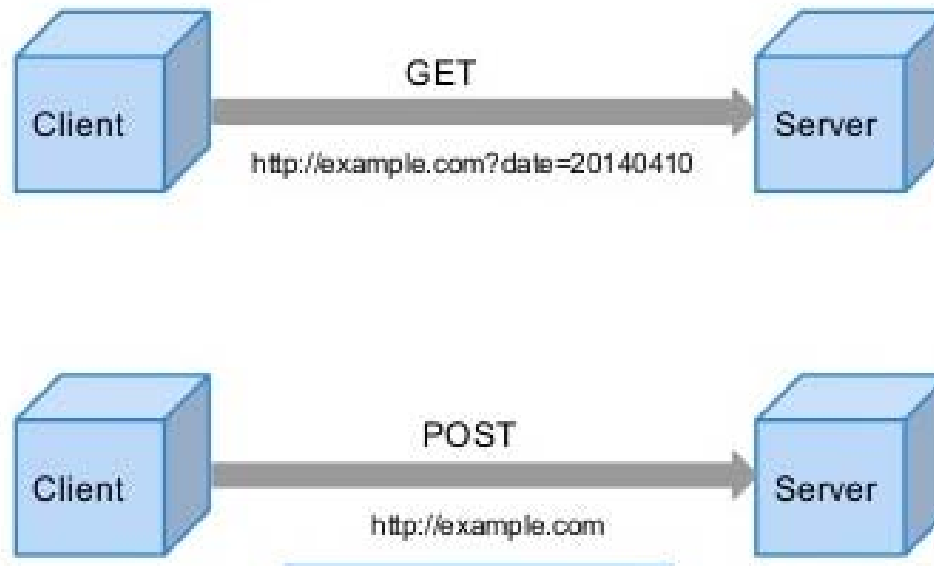
- Start up your web browser, and make sure your browser's cache is cleared.
- Open up the Wireshark.
- Start Wireshark packet capture.
- Go back to your web browser and enter:
<http://www.info308a.ece.mcgill.ca/Lab1Ex4.html>
- Stop Wireshark packet capture, and type “http” in the filter field.

Retrieving a web page: Questions

- 52. How many HTTP GET requests are sent by your web browser?
- 53. What is the content type of each response message?
- 54. Did your browser download the two images serially or in parallel? Explain. What are the pros and cons of each approach?
- 55. Has the HTTP used persistent or non-persistent connection? Explain your answer.

HTTP request methods: Overview

- In this last exercise, we will examine a trace file in which the user tried to connect to a password-protected website. We will see what HTTP messages are exchanged in this scenario.



HTTP request methods: Instructions

- Enter the following URL into your browser:
- <http://www.info308a.ece.mcgill.ca/HTTP-Authentication.pcapng>, save the file on your computer.
- Open up the Wireshark and open the trace file that you just download.
- Type “http” in the filter field so that only captured HTTP messages will be displayed later in the packet-listing window.

HTTP request methods: Questions

- 56. 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 the password?
- 57. 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.
- 58. What is the status code of the frame#174? What is the description of this code?