

ECSE 308:
Introduction to
Communication Systems and Networks

L5:

Part 1: TCP

Part 2: HTML

Part 1: TCP

- The TCP is a connection-oriented protocol meaning that it establishes an end-to-end connection before any data is sent.
 - Typically, TCP connections go through three phases: connection establishment, data transfer and termination. These phases in TCP segments are identified by flags.
 - TCP uses an end-to-end flow control protocol to avoid having the sender send data too fast for the TCP receiver to receive and process it reliably.
 - It applies a *sliding window flow control* protocol. In each TCP segment, the receiver specifies in the *receive window* field the amount of additionally received data (in bytes) that it is willing to buffer for the connection. The sending host can send only up to that amount of data before it must wait for an acknowledgment and window update from the receiving host.
 - TCP applies congestion control mechanisms which control the rate of data entering the network, keeping the data flow below a rate that would trigger collapse.
 - For each connection, TCP maintains a congestion window, limiting the total number of unacknowledged packets that may be in transit end-to-end.
 - This is somewhat analogous to TCP's sliding window used for flow control.
 - TCP uses a mechanism called slow start to increase the congestion window after a connection is initialized or after a timeout.
 - It starts with a window of two times the maximum segment size (MSS). Although the initial rate is low, the rate of increase is very rapid; for every packet acknowledged, the congestion window increases by 1 MSS so that the congestion window effectively doubles for every round-trip time (RTT).
-
- Use Start up your web browser and clear the browser's cache memory.
 - Open up the Wireshark and start packet capture.
 - Go to the <http://www.info308a.ece.mcgill.ca/Lab1Ex2.html>.

- Stop Wireshark packet capture.
- Find the IP address of the website i.e. : `***.***.***.***`
- In the filter field, type “`ip.addr == ***.***.***.*** && tcp`” to see only the TCP packets. Press Apply

1. How many TCP datagrams are exchanged between your computer and the server to establish the TCP connection? Why each of these segments is needed to setup the TCP connection?
2. Which end point started the TCP Connection-Establishment phase?
3. What flags are set in each of these TCP datagrams?
4. What is the initial value of the sequence number on the client's side?
5. What is the initial value of the sequence number on the server's side?
6. What is the value of the Acknowledgement field in the SYN ACK datagram? How did the server determine that value?
7. For the TCP SYN datagram, determine the following
 8. the source port number
 9. the destination port number
 10. the size of the window
 11. the header length
12. For the TCP SYN ACK datagram, determine the following
 13. the source port number
 14. the destination port number
 15. the size of the window
 16. the header length
17. What is the usage of the window field in the TCP segments?
18. Consider the TCP segment containing the HTTP GET as the first segment in the TCP connection. For the first three TCP segments, answer the following questions:
 19. When was each segment sent?
 20. At what time was the ACK for each segment received?

Part 2: HTTP

The 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.
 - 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.

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?

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

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?

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.

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.

- 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?