

Computer Networks Lecture Note–March 24, 2020

qcm8ab

March 26, 2020

1 Exam 2 Information and Logistics

- Exam will be online on Collab under quizzes. It will be open for 48 hours, starting around 12AM on Thursday. It is designed to take an hour and 20 minutes.
- The exam is **open book, open note, open slides, etc.** Python documentation is permitted, but otherwise the internet is not allowed. Talking to other students or the TAs is not allowed. Professor Graham will be on Zoom during normal class time to answer any questions that come up.
- **The exam will only cover the Transport Layer and the HTTP Protocol.** Understand the following topics:
 - Socket programming
 - HTTP 1.1
 - State machines
 - Reliable transport
 - Calculating utilization and the effect of windowing
 - Pipelining
 - On-Off protocol
 - Estimating round trip time (RTT)
 - Exponential weighting strategy for RTT samples
- The exam will be **10 questions.**
 - 4 or 5 questions will involve analyzing a Wireshark trace.
 - 1 coding question.
 - The remainder are short answer and multiple choice.

2 Review Topics

2.1 Difference between UDP and TCP

- Overall, we have an unreliable network. Packets get dropped due to queueing, corruption, and collisions. We want to build reliable transport on top of the unreliable network.
- **UDP:** Don't care if the packets get lost. It has no overhead, and can be used for loss-tolerant transmission (audio and video) to improve performance.
- **TCP:** Guarantees reliable in-order delivery of packets. It requires the use of ACK numbers and sequence numbers, and processes such as Go Back N and Selective Repeat to keep track of which packets have been received.
- TCP requires flow control

Problem: if the receiving application layer doesn't process information quickly enough, the receiving side buffer might be overwhelmed.

Solution: On-Off messages can be sent from the receiver to the sender to tell the sender when the receiver's buffer is going to be full. This can be calculated using propagation time and the difference in the rate between the sender and the receiver. *Alternative Solution:* Limit the amount of information sent using a window.

2.2 HTTP Protocol

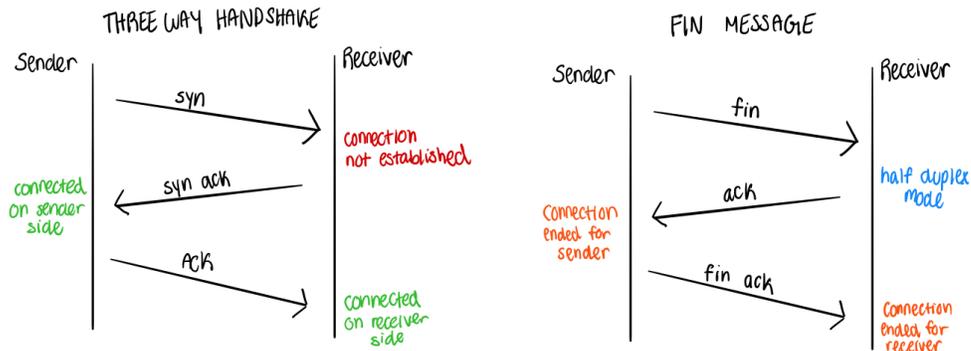
HTTP Protocol Steps

1. Have a server running TCP on socket 20
2. Have a browser, which asks the DNS server to retrieve an IP address for a website
3. A TCP connection is opened to this IP address on port 8.
4. Three-way handshake occurs.

Data is sent in the form of request-response pairs. For example, the browser could send a GET request, and the server would respond with information about the server and data (usually HTML stuff), which is then processed, parsed, and displayed by the browser.

2.3 Three-Way Handshake and Fin Messages

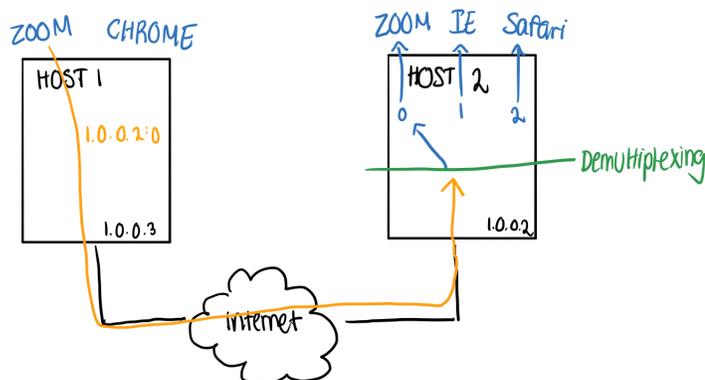
The three-way handshake is a method for establishing a communication link before transmitting any data. Both parties must take two actions in order for the connection to be fully established. A communication link is disconnect through a similar method using Fin messages.



2.4 Multiplexing and De-multiplexing

Multiplexing: A host may be running more than one process at a time.

De-multiplexing: Need to determine what process a packet is destined for on the receiving side. The operating system stores a mapping of process IDs and port numbers.



3 Socket Programming Example

The following program sets up a server-side socket connect and waits for greetings to be sent to it. When it receives a message, it response with "Hello Back."

```
from socket import *
serverPort = 8080
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind(("", serverPort)) #use ifconfig to find IP address
serverSocket.listen(55) #55 is number of connections allowed
print("Setup socket and listening")
while True:
    connectionSocket, addr = serverSocket.accept()
    print("We're connected! " + str(addr))
    message = connectionSocket.recv(1024)
    print(str(addr) + " says " + str(message))
    responseMessage = "Hello Back"
    connectionSocket.send(responseMessage.encode())
    connectionSocket.close()
```