

**\*\*\* NOVEMBER 30, 2004 \*\*\***

### **Lab 5 - Net Phone**

- The mirror server is now an extra credit
- To test it, we will use the 'netstat' command that displays the current TCP/UDP connections to make sure that during a call there are only connections from the phone to the mirror server.

-----

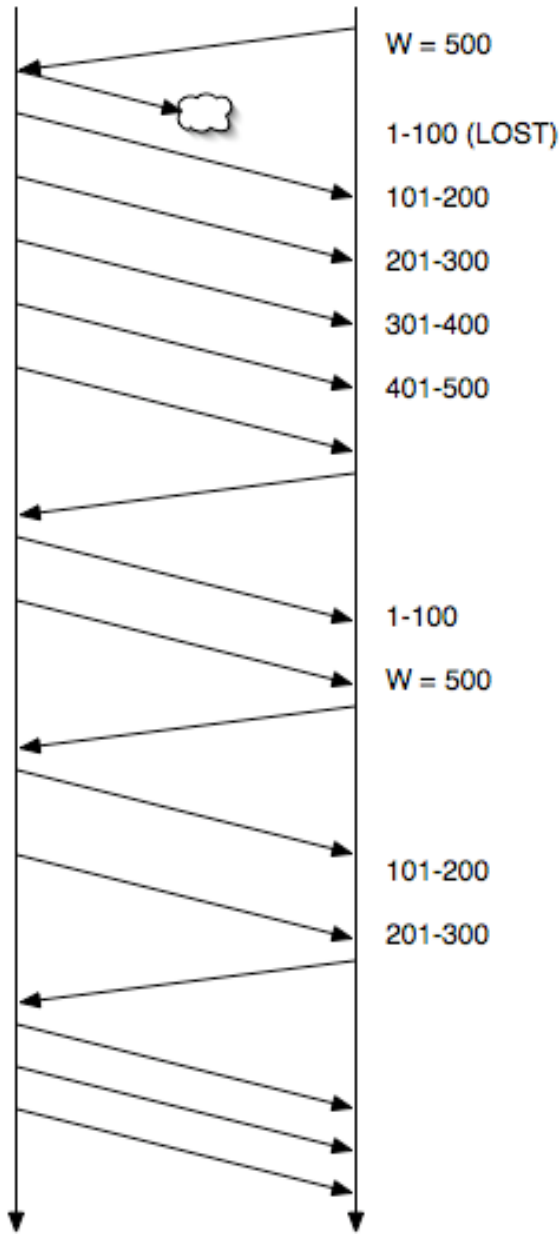
TCP (continued...)

- Repeat of Adaptive Retransmission and Flow Control from 11/23
  - See notes from then

**\*\*\* NEW STUFF \*\*\***

### **Congestion Control in TCP**

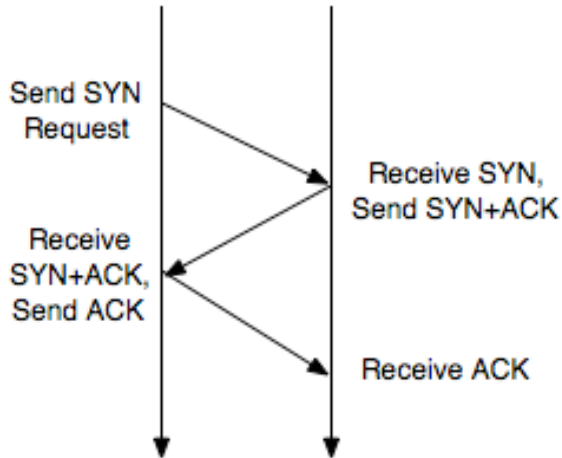
- When a network is congested, packets are dropped
  - 1000 Mbps -> Router -> 500 Mbps sent
  - > 500 Mbps dropped
- If packets rate dropped, TCP will start retransmitting packet making the congestion worse
- TCP alleviates this problem by slowing down retransmission in the presence of packets lost
- After a packet is lost, TCP will send only one packet of data instead of the full window
- If the packet is acknowledged on time, it sends two packets, then 4, and so on, until it complete 1/2 window. Then it increases the number of packets one by one (linearly) until it complete full window
  - (so exponential growth to 50% of window, then linear (+1) until full)
- This process is called "slow start", that is the reaction of TCP to packet lost
- For TCP, packet lost == congestion (even if it was caused by something else)
- "Slow start" in the presence of congestion works because all implementations of TCP cooperate and do the same. TCP is called a "nice" protocol because of this cooperation



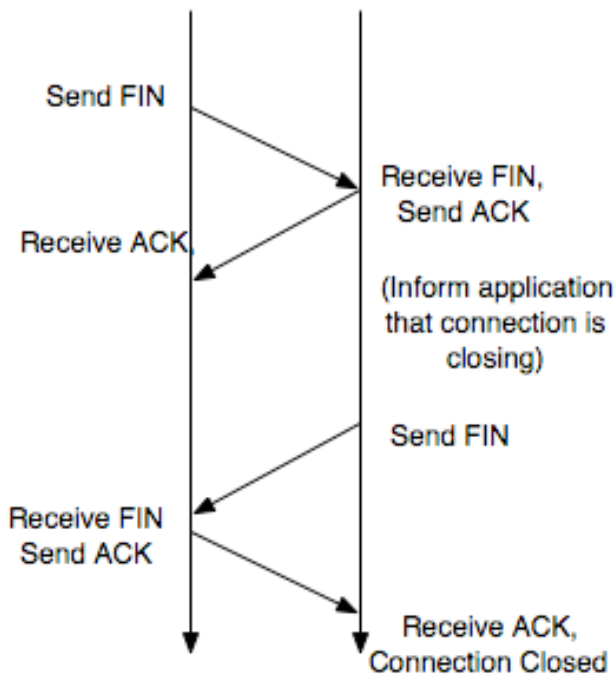
### Reliable Connection Startup And Shutdown

- To achieve reliability we need reliable connection startup / shutdown
- Why reliable connection startup / shutdown is difficult?
  - Packets can be lost, duplicated, out of order
  - Either side may crash and reboot
  - Duplicate shutdown messages may affect later connections
- TCP uses "three way handshake", which means both startup and shutdown need at least 3 messages

### Opening a Connection:



**Closing a Connection:**



- closing a connection is more complex, because it may take time for the application to acknowledge that its TCP connection is being closed.

\*\*\* DECEMBER 2, 2004 \*\*\*

**Lab 5 Notes**

---

- Grading will be done in CS 175 on Wednesday, December 8th
- I will post the available times on Monday on the CS 175 door

(1 slot per group)

- Make sure that your program works on the CS 175 machines
- Turn-in deadline is Tuesday Dec 7th, 11:59 pm
- Arrive 10 minutes before your time to setup the machine where you will present your program

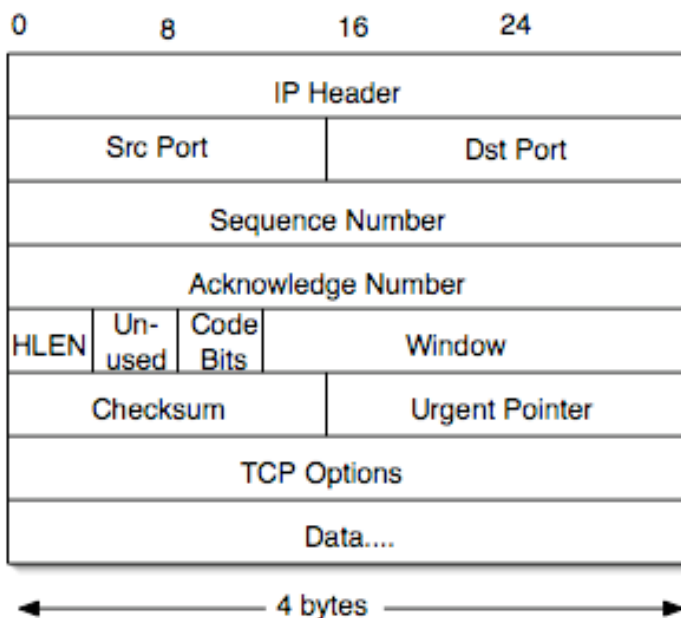
Optimizations for the Internet Phone:

- You could eliminate player thread and jitter buffer by copying directly to player secondary buffer directly
  - use the player secondary buffer AS the jitter buffer
  - when a packet arrives, you copy it directly to the player secondary buffer
- Do not start playing immediately
  - wait until the buffer is "full" or at least partially full
  - because packet 1 is there, packet 2 or 3, etc might be delayed (wait until maybe 0.5 seconds of sound is available?)
- If you decide to have a jitter buffer instead of using the secondary buffer, you still may remove the player thread by copying data from the jitter buffer to secondary buffer during capture events.
  - use the capture thread to:
    - copy data from the jitter buffer to the second buffer
    - copy captured data and form and send packet

----

### TCP Packet Format

- All TCP packets have a TCP header regardless of the data they contain
- Each connection is also a receiver and the receiver is a sender
- The mechanisms for retransmission, flow control, etc are available in each perr
- A program using TCP is identified by a port number



Source port / Destination ports

- They determine the source and destination applications
- Ports go 0-2<sup>16</sup>-1 (64k ports)
- Ports 0-1023 are reserved by the OS
  - In UNIX, only root can use them
- A TCP connection is defined uniquely in the whole internet by four values:
  - Src IP Addr, Src Port, Dest IP Addr, Dest Port

**- Sequence Number**

- It gives the offset or position of the data sent in the stream of data
- Data is accepted only if the sequence number -> seq num + len is inside the window
  - This is done so junk packets from previous connection with the same ports are not accepted in the current connection
  - TCP starts with a random sequence number to make the previous event more

unlikely

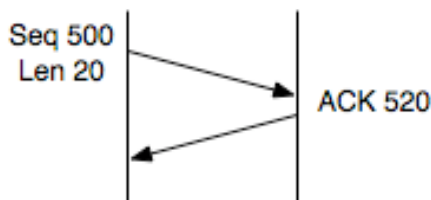
- Also by default TCP will not allow you to reuse a port for 5 minutes after it has been closed
  - to make sure that the packets that belong to the previous connection have been eliminated.

This makes the previous event more unlikely to happen

- Servers are allowed to use the SO\_REUSEADDR option to reuse a port immediately
  - Clients shouldn't use it and they don't need to
  - They don't need it since they can use a different port every time

**- Acknowledge Number**

- This is the sequence number of the last byte of data received correctly without any gaps



**- HLEN (4 bits)**

- length of header in 4 byte multiples. Most of the time the TCP header size is 20 bytes except when TCp options are included

**- Code Bits (6 bits) -**

- They include the types of message:
  - SYN bit, FIN bit, ACK bit

**- Window**

- It gives the available buffer space in the receiver

**- Checksum (2 bytes)**

- For error detection. It is computed by adding all the two-byte words in the data

**- Urgent Pointer**

- It is used for out-of-band data, that is information we want to be received ahead of any current data
  - (Example Control-C in Telnet session)

**- TCP Options**

**- TCP data**