

Introduction to Networks and the Internet

CMPE 80N

Winter 2004

Lecture 21



Announcements

- Library presentation on Wed. March 3rd.
- “Internet behind the Web” video on March 10th.
- 5th. Quiz Friday, March 12th.
- Final exam, Thu March 18th.
- Stay tuned for:
 - Discussion sessions for 5th. quiz.
 - Discussion sessions for final.

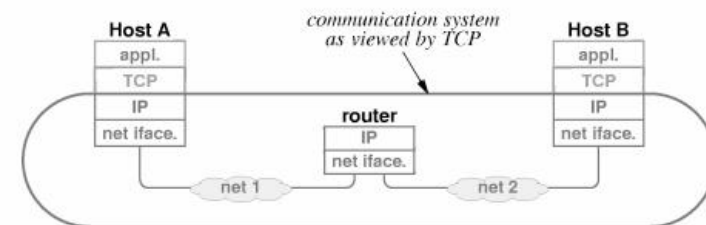


TCP (cont'd)



Delivering TCP

- TCP at destination interprets TCP messages



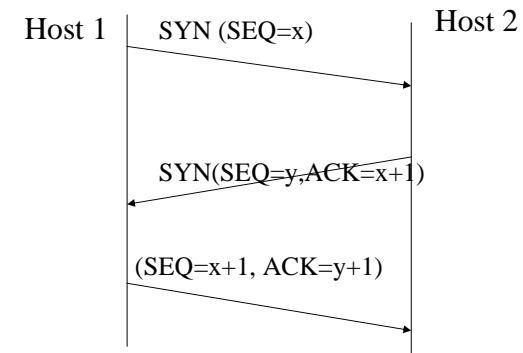
Features of TCP

- *Connection-oriented.*
- *Reliable delivery.*
- *Ordered delivery.*
- *Flow and congestion control.*



TCP Connection Setup

- *3-way handshake.*



TCP and Reliable Delivery

- *TCP provides reliable delivery, recovering from:*
 - *Lost packets*
 - *Duplicate packets*
 - *Delayed packets*
 - *Corrupted data*
 - *Source/destination mismatches*
 - *Congestion*



TCP Reliability

- *Reliable delivery.*
 - *Acknowledgements..*
 - *Timeouts and retransmissions.*
- *Ordered delivery.*
 - *Sequence numbers.*



Lost Packets

- Recipient sends **acknowledgment** control message (**ACK**) to sender to verify successful receipt of data
 - ACKs usually are carried onboard other TCP packets.
 - However, even if an application has nothing to transmit, it must transmit acknowledgment packets for each packet it receives.
- Thus, for each packet sent, a host expects to receive an acknowledgment, which ensures that the packet did not get lost.
 - What if the packet or the acknowledgment get lost?



Lost Packets (cont'd)

- **Retransmission timer**
 - When a data segment is sent, a timer is started
 - If the segment is acknowledged before the timer expires, the timer is stopped and reset
 - Otherwise, the segment is retransmitted (and the timer is reset and started again)
- **The choice of the timeout is critical!**
 - If timeout is too long: overall throughput may be reduced (always waiting for acknowledgments)
 - If timeout is too short: too many packets get retransmitted (may increase network congestion)

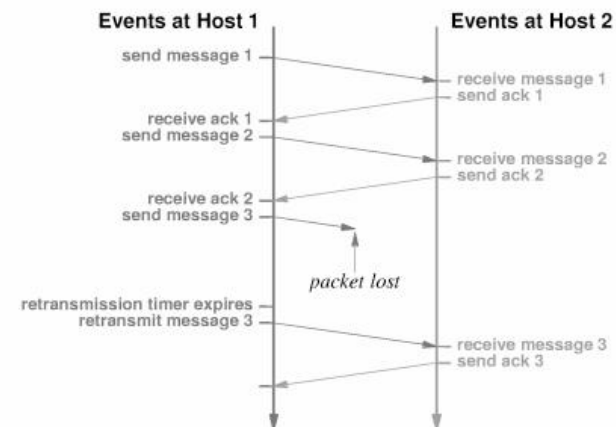


Lost Packets (cont'd)

- **IMPORTANT:** packet retransmission (especially if it has to be carried out on an end-to-end basis) **significantly increases latency (delay)**
 - For real-time video or audio transmission, **delay** is a more important performance issue than **error rate**
 - Thus, in many cases it is preferable to forget the error and simply work with the received data stream



Lost Packets - Example



TCP Transmission

- *Sender process initiates connection.*
- *Once connection established, TCP can start sending data.*
- *Sender writes bytes to TCP stream.*
- *TCP sender breaks byte stream into segments.*
 - *Each byte assigned sequence number.*
 - *Segment sent and timer started.*



TCP Transmission (cont'd)

- *If timer expires, retransmit segment.*
 - *After retransmitting segment for maximum number of times, assumes connection is dead and closes it.*



Flow Control

- **Flow control** is necessary so that source doesn't transmit too fast for given receiver.
 - *E.g., a fast server trying to send 1Gb/s data to a small PC.*
 - *Without some form of control, some data will get lost.*
- **Requires feedback from receiver.**
 - *So sender realizes it is sending too fast.*
 - *Depends on receiver's storage (buffer) capacity.*

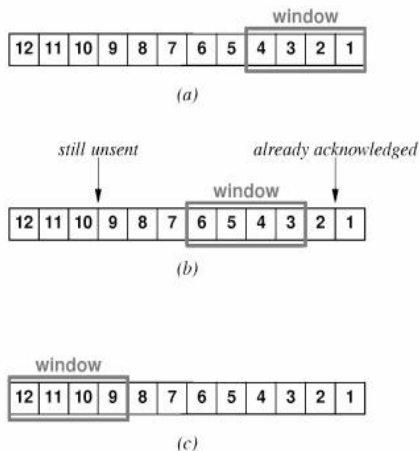


Sliding Window

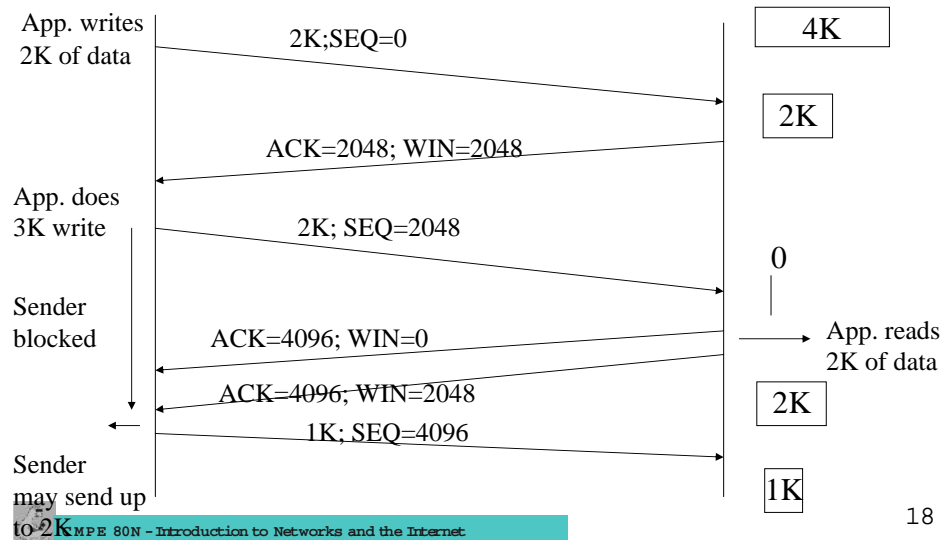
- *TCP uses ACKs and sliding window mechanism for flow control.*



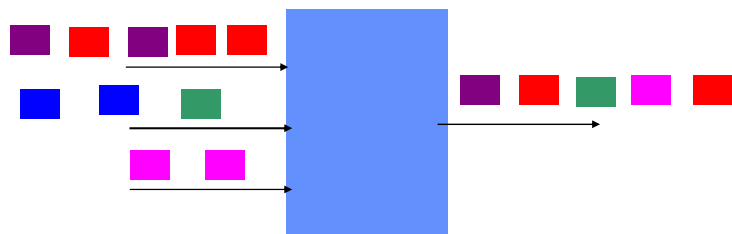
TCP Sliding Window



TCP Flow Control: Example



Queuing



Packets are queued inside router, waiting to be transmitted on outgoing link.

Queuing (cont'd)

- What happens if packets are coming in faster than router is able to forward them?
 - Needs to “queue” the packets as they are waiting to be transmitted.
- What happens when a queue becomes too long?
 - Packets are dropped!
- How can the receiver detect that a packet was dropped?
 - Look at the packet sequence number!

Congestion

- Network with 1 Mb/s lines and 1000 computers, half of which are trying to transfer files at 100 Kb/s to the other half.
 - The total offered traffic exceeds what the network can handle (**congestion**).
- **Congestion collapse:**
 - When congestion occurs, packets get dropped.
 - Due to packet loss, packets get retransmitted.
 - Congestions gets worse and worse!



Congestion Control

- Why do it at the transport layer?
 - Real fix to congestion is to slow down sender.
- Use law of “conservation of packets”.
 - Keep number of packets in the network constant.
 - Don't inject new packet until old one leaves.
- Congestion indicator: packet loss.



TCP and Congestion Control

- Interprets packet loss as an indicator of congestion
 - When it senses packet loss, it slows down the rate of packet transmission
 - When packets are received correctly, sends packets faster
 - Still within the limits of the sliding window



TCP Congestion Control

- Like, flow control, also window based.
 - Sender keeps congestion window (cwin).
 - Each sender keeps 2 windows: receiver's advertised window and congestion window.
 - Number of bytes that may be sent is $\min(\text{advertised window}, \text{cwin})$.



TCP Segment Header

Source port		Destination port		
Sequence number				
Acknowledgment number				
Header length	U	A	P R S F	Window size
Checksum		Urgent pointer		
Options (0 or more 32-bit words)				
Data				



UDP

- Provides connection-less, unreliable service.
 - No delivery guarantees.
 - No ordering guarantees.
 - No duplicate detection.
- Low overhead.
 - No connection establishment/teardown.
- Suitable for short-lived connections.
 - Example: client-server applications.



UDP Segment Format

0	15	31
Source port		Destination port
Length		Checksum
Data		

Source and destination ports: identify the end points.

Length: 8-byte header+ data.

Checksum: optional; if not used, set to zero.



TCP and UDP

- **TCP** provides end-to-end communication. It takes care of **reliable, error-free transfer** of data, and **in-sequence delivery**
- **UDP** has **less overhead** compared to TCP, but **does not guarantee transfers**
 - TCP is preferred to transfer files
 - UDP is preferred to transfer audio/video streams
 - In real-time streaming, we cannot afford the delay consequent to packet retransmission
- Both protocols support **multiplexing**, i.e. they allow several distinct streams of data between two hosts



The Domain Name System (DNS)



DNS

- IP addresses are not easy to remember.
- The **Domain Name System (DNS)** maps IP addresses to host names.
- Host name is formed by machine name followed by **domain name**.
 - *Host_name.domain_name*



DNS (cont'd)

- The *domain_name* is formed by the **institutional site name** and the **Top-Level Domain name (TLD)**.
 - So the host name is of the form: *machine_name.Ist_site_name.TLD_name*
- **Examples:**
 - *sundance.ucsc.edu*
 - *soe.ucsc.edu* (alias for *sundance.ucsc.edu*)
 - *italia.cse.ucsc.edu*
 - *helios.jpl.nasa.gov*
 - *www.cnn.com*



TLD

- TLD names identify **organization types** or **country codes**.
- **Examples:**

<i>.com</i> Commercial org.	<i>.au</i> Australia
<i>.edu</i> Educational site in US	<i>.ca</i> Canada
<i>.gov</i> Government site in US	<i>.fr</i> France
<i>.mil</i> Military organization in US	<i>.de</i> Germany
<i>.net</i> Network site	<i>.uk</i> Great Britain
<i>.org</i> Nonprofit organization	<i>.it</i> Italy
	<i>.es</i> Spain

Countries define their own internal hierarchy (e.g., *.ac.uk*, *.edu.au*)

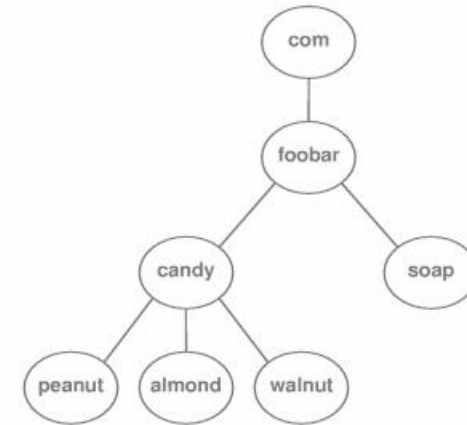


DNS (cont'd)

- Organizations can create any internal DNS hierarchy.
- Authority for creating new **subdomains** within a domain name is delegated to each domain.
 - Administration of *ucsc.edu* has authority to create *cse.ucsc.edu* and need not contact any central naming authority.



Example of DNS Hierarchy

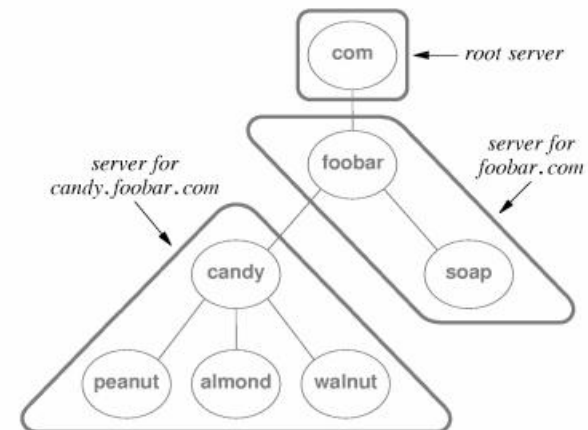


DNS Name Space

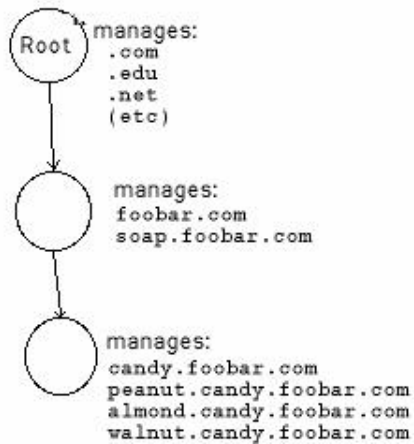
- DNS names are managed by a hierarchy of **DNS servers**.
 - Hierarchy is related to DNS domain hierarchy
- Root server at top of tree knows about next level servers.
- Next level servers, in turn, know about lower level servers.



Example of DNS Hierarchy



Example of DSN Hierarchy (cont'd)



Choosing DNS Server Architecture

- *Small organizations can use a single server.*
 - *Easy to administer.*
 - *Inexpensive.*
- *Large organizations often use multiple servers.*
 - *Reliability through redundancy.*
 - *Improved response time through load sharing.*



Name Resolution

- *“Resolving a name” means mapping the host name to the IP address.*
 - *Reverse mapping is also possible.*
- *A **client** computer calls a **DNS server** for name resolution*
 - *DNS request contains name to be resolved.*
 - *DNS reply contains IP address for name in request.*



Using DNS Servers

- *Each DNS server is the **authoritative server** for the names it manages.*
 - *If request contains name managed by receiving server, that server replies directly.*
 - *Otherwise, request is forwarded to the appropriate authoritative server.*
- *DNS request is originally sent to **root server**, which points at next server to use*
 - *Eventually, the authoritative server for the DNS name in the request is located and IP address is returned.*

