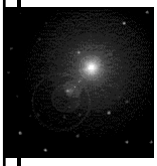
CS 4/54201 Computer

Communication Network

Kent State University

Dept. of Computer Science

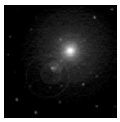
www.mcs.kent.edu/~javed/class-NET06F/



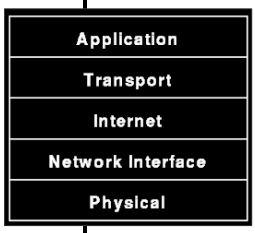
A Course on Networking and

Computer Communication

- IP- Internet Protocol
  - Addressing Scheme
  - Address Resolution
  - Datagram Forwarding
  - Encapsulation, Fragmentation & Reassembly
- TCP- Transmission Control Protocol
  - Connection startup & shutdown
  - Reliability: ordering, missing data handling
  - Flow control

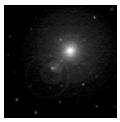


COMPUTER COMMUNICATION NETWORK



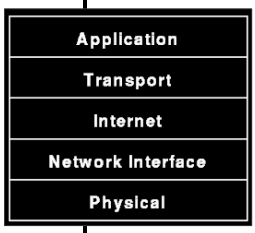
TCP

Why TCP?

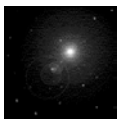


COMPUTER COMMUNICATION NETWORK

- Packet service provided by IP is unreliable.
  - IP takes no responsibility if packets are lost,
  - if there are duplicate packets,
  - if a part of the router fails.
- But, an application program wants to assume that when a byte is send it will be delivered correctly at the other end.
- TCP layer of TCP/IP protocol suits bridges this gap.



TCP Services



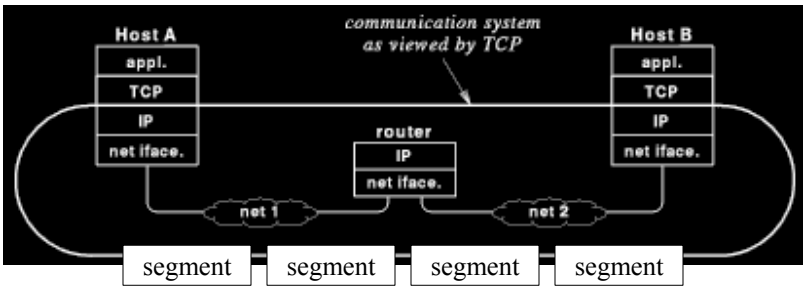
COMPUTER COMMUNICATION NETWORK

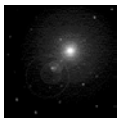
- Connection Orientation
  - application sets up virtual connections.
- Point-to-point Communication
  - each connection has exactly two endpoints.
- Complete Reliability
  - TCP guarantees all bytes will be delivered.
- Full Duplex Communication
  - Data can flow in both direction over connections.
- Stream Interface
  - Byte order is maintained but no records.
- Reliable Connection Startup
  - No interference from earlier connections.
- Graceful Connection Shutdown
  - All sent data will be delivered before shutdown.

TCP provides a completely reliable (no data duplication or loss), connection oriented, full duplex stream transport to applications.

Virtual Connection

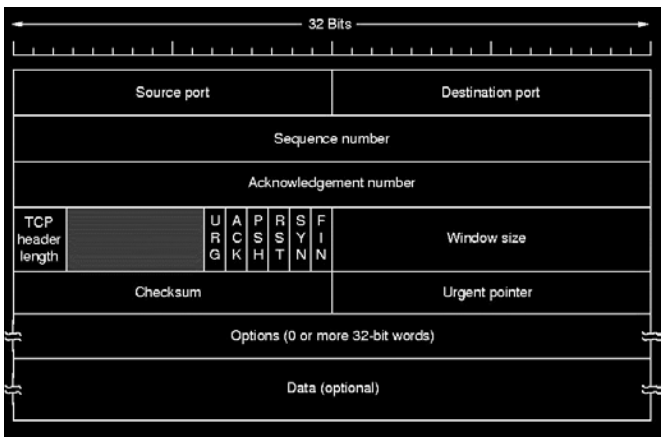
- TCP provides a feeling to the applications that a completely reliable connection exists between two applications.
- TCP use IP to carry packets.
- However, the IP or underlying hardware do not know about the connection.

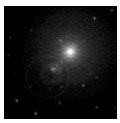




COMPUTER COMMUNICATION NETWORK

TCP Segment Format





COMPUTER COMMUNICATION NETWORK

FIN=1 is used for closing a connection.

SYN=1, is used to request a connection

URG=1 if urgent point is in use Urgent pointer is used to request prompt delivery of data at the receiving end, even if out of sequence.

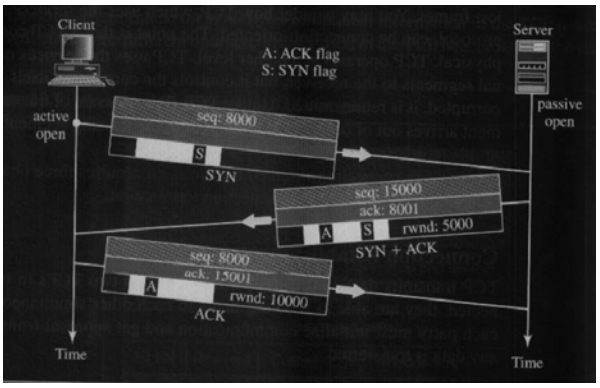
ACK=1 if ack field is valid

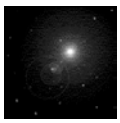
PSH=1 to indicate immediate send.

Connection Management

Connection Open: Three Way Handshake

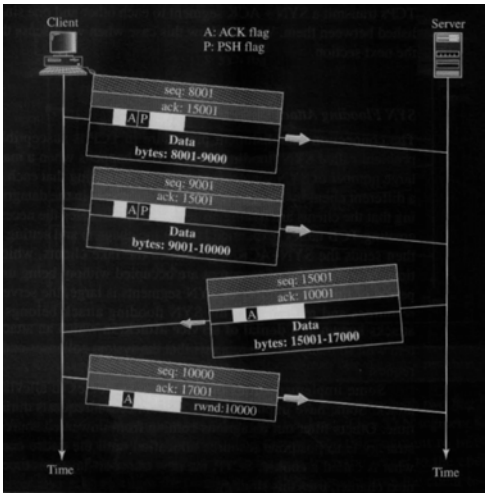
- TCP uses a three way acknowledgement scheme for making sure that connection can be created and gracefully terminated even when there is computer reboots.

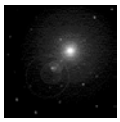




Data Transfer

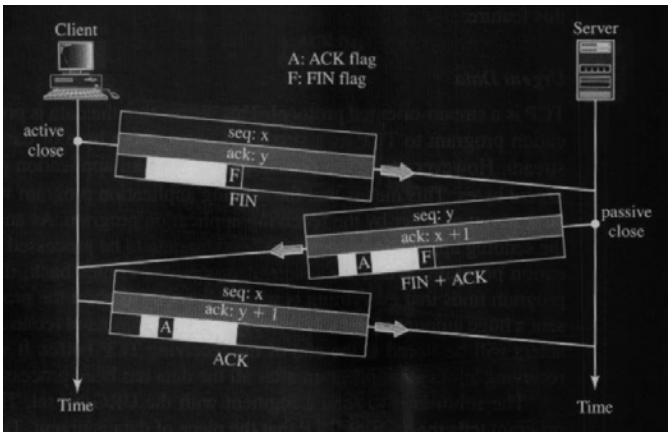
- Once connection is OPEN, it sets up a bidirectional connection and data can be sent both ways.

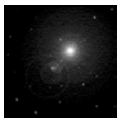




Connection Termination: Three way Handshake

- Both FIN segment and replied FIN+ACK segment consumes one sequence number each if there is no data to send.

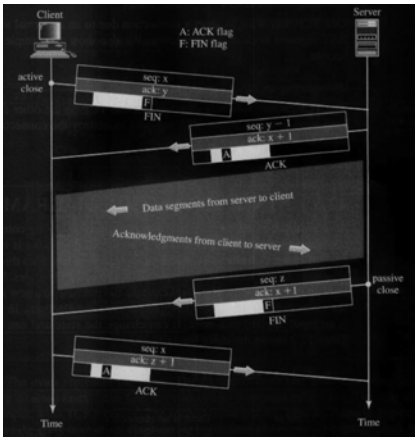


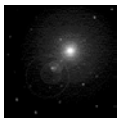


Quiz: What if one side has data to send and other side wants to close?

Connection Termination: Half-Close

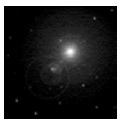
- Each side can decide to close the connection separately. In this case the server side wants to send more data. So it only sends “ACK”, and a sequence number y-1. Thus would not sent data in this segment. So the client side will keep the receiver open.



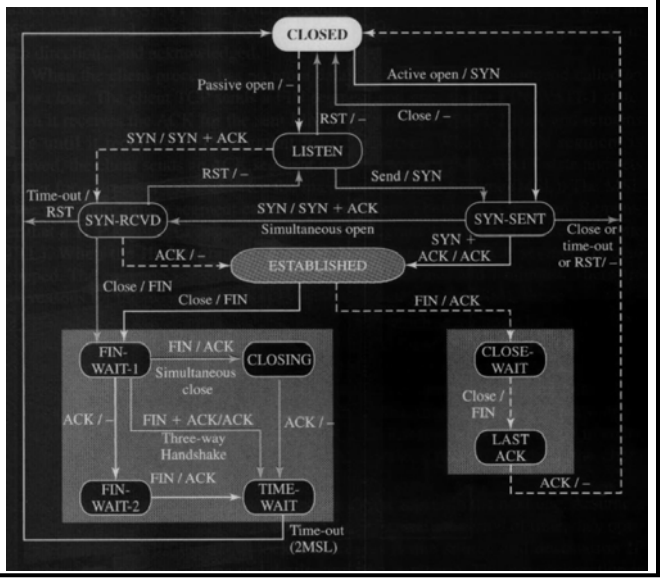


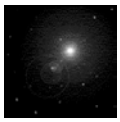
TCP States

CLOSED	There is no connection
LISTEN	Passive open received; waiting for SYN
SYN-SENT	SYN sent; waiting for ACK
SYN-RCVD	SYN+ACK sent, waiting for ACK
ESTABLISHED	Connection established, data transfer in progress
FIN-WAIT-1	First FIN sent; waiting for ACK
FIN-WAIT-2	ACK to first FIN received; waiting for second FIN
CLOSE-WAIT	First FIN received, ACK sent; waiting for application to close
TIME-WAIT	Second FIN received, ACK sent, waiting for 2MSL time-out
LAST-ACK	Second FIN sent, waiting for ACK
CLOSING	Both sides have decided to close simulteneously.



TCP State Diagram

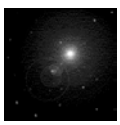




COMPUTER COMMUNICATION NETWORK

Recent Advances in Networked & Applications Workshop Series:

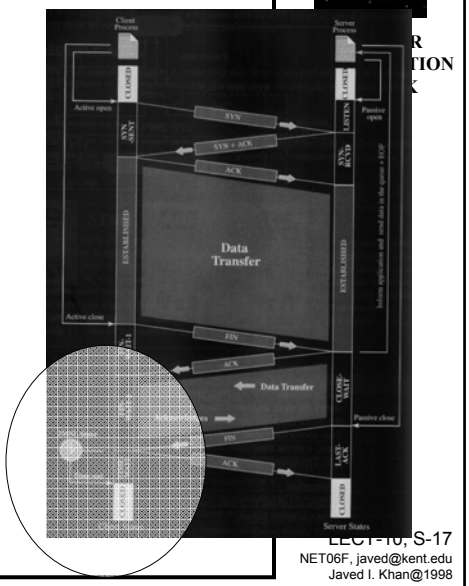
- Today at 3:30-5:00 pm
- 1. November 30, 2005: Advanced Internet Systems: Internet Caching & Content Distribution Networking
- 2. December 07, 2005: Internet Traffic Management: Inter Domain Routing with Autonomous Systems using BGP.
-
- 3. December 14, 2005: The Emerging Era-of cooperative Computing: Recent Advances in Peer-to-Peer, Grid and Community Computing.



COMPUTER COMMUNICATION NETWORK

Client States

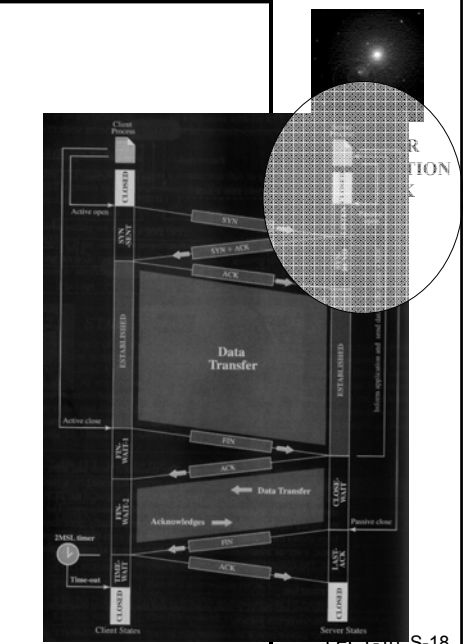
- Client requests Active Open for a connection and moves through SYN-SENT to ESTABLISHED state.
- When done, it issues a Active Close and moves to FIN-WAIT-1 state.
- If server is not done with data transfer, it moves to FIN-WAIT-2.
- When receives FIN from server it sends ACK but does not close the connection immediately. It starts a timer 2MSL. Why?
  - In case ACK is lost, the server have to keep connection open for ever. To avoid that if server does not see this last ACK it resends FIN. Upon receiving this retransmitted FIN the client must resend the ACK. Client must wait till server's retransmitted FIN can arrive.
  - Segments from one socket must not reincarnate in the second socket with same id. Thus, the first socket must wait in zombie state (2MSL) time before it can be reused.



NET06F, javed@kent.edu
Javed I. Khan@1998

Server States

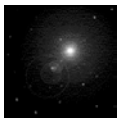
- Server issues a passive open command and moves into LISTEN state.
- Once it receives SYN from client it moves to SYN-RCVD and then than to ESTABLISHED states.
- If it receives a FIN request from client, but it has data to send, it moves to CLOSE-WAIT state and the sends its remaining data.
- The it sends its FIN request to the other side and starts a regular times and moves to LAST-ACK state.
- It retransmits the FIN if there is a time out until it receives the last ACK.



NET06F, javed@kent.edu
Javed I. Khan@1998

Quiz

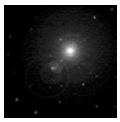
- Quiz 601: What may cause Silly Window Syndrome?
- Quiz 602: Why TCP sending end waits some extra time to tear down the socket even after sending the very last FIN+ACK?
- Quiz 603: Jacobson added a new consideration to get a more effective estimation of TCP timeout. What is that?
- Quiz 604: What ambiguity Kern removed?



Reliability

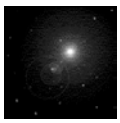
Reliability Problems

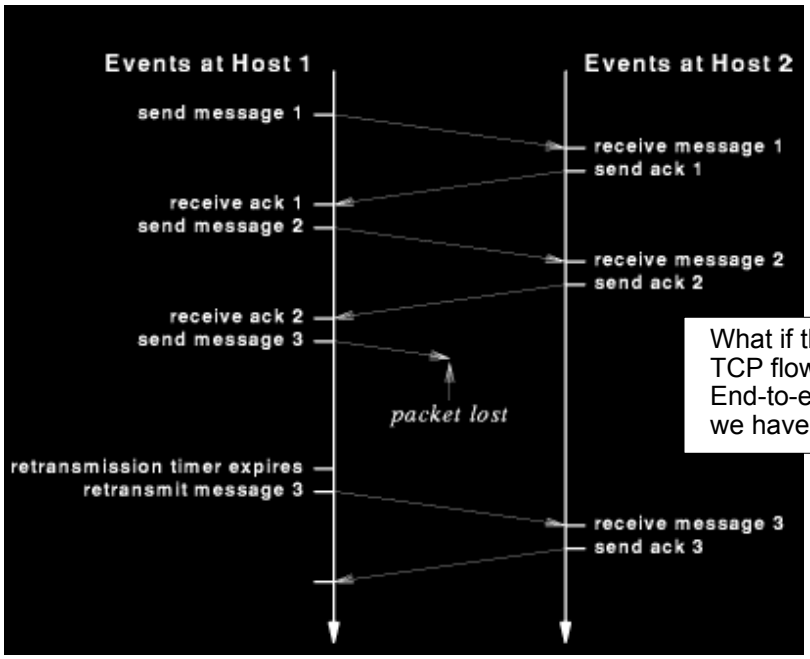
- Packet can be lost
- Packet can be duplicated
- Packets can get reordered.
- Computers can reboot
- A packets from earlier connection can disrupt current connection.

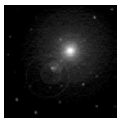


Packet loss and Retransmission

- TCP uses retransmission assisted by a timer.
- When a packet is send a timer is started.
- Each packet must be acknowledged by the receiver.
- If acknowledgement does not arrive within the specified time, retransmission occurs.







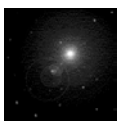
COMPUTER COMMUNICATION NETWORK

What if the difference of TCP flow control from the End-to-end flow control we have studied earlier?

Heterogeneous Environment

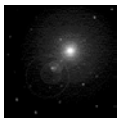
- What should be a good time out time?
  - Local computers can respond within ms.
  - A satellite connected computer needs about .25min.
  - Use long delay for satellite connections and short for local connections ?

What if the time over the same connection varies?



COMPUTER COMMUNICATION NETWORK

Adaptive Retransmission

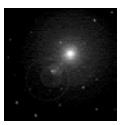


COMPUTER COMMUNICATION NETWORK

Unlike the Data-Link Layer TC protocols:

- TCP uses an adaptive scheme.
- TCP monitors the round-trip-time for each communication.
- Measures average delay and variance.
- Sets the times as a weighted sum of average and variance.

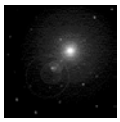
Original Algorithm



COMPUTER COMMUNICATION NETWORK

- Measure Sample RTT M for each segment/ACK pair
- Compute smooth average of RTT
  - RTT = α. RTT+(1- α). M
  - α between 0.8 and 0.9
- Set timeout based on RTT
  - Time Out = 2 x RTT

Jacobson Algorithm (Jacobson' 1988)

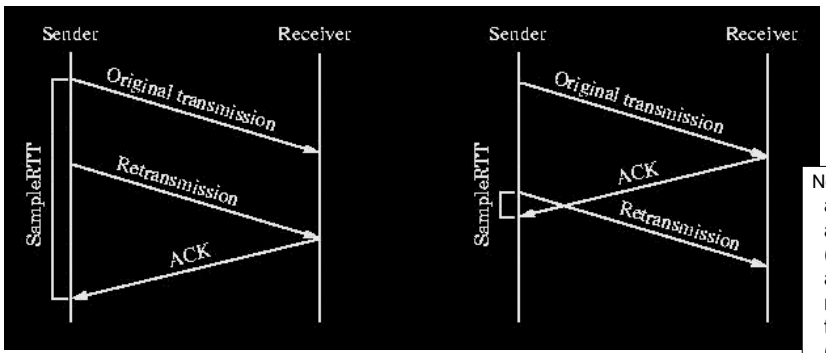


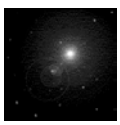
COMPUTER COMMUNICATION NETWORK

- Keep Track of the Variance also
  - Current Variance = |RTT - M|
  - Keep a smooth estimate of D = α D + (1- α) |RTT-M|
  - Timeout= RTT + 4*D

If there is a retransmission, and one ACK comes back which RTT should be used?

Karn/Partridge Algorithm





COMPUTER COMMUNICATION NETWORK

Notes
algorithm only as good as granularity of clock (500ms on Unix)
accurate timeout mechanism important to congestion control (later)

- If there is a retransmission which RTT are we measuring?
- KP Solution:
  - Do not sample RTT when retransmitting
  - Double timeout after each retransmission (exponential back-off)

Flow Control