The University Of Alabama in Huntsville Computer Science

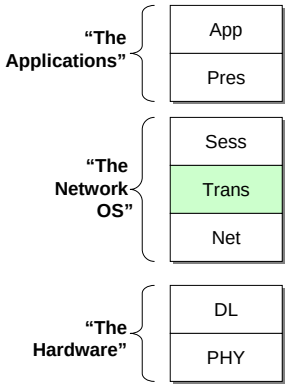
cs570

Transport Layer

The University Of Alabama in Huntsville Computer Science

Transport layer

cs570



Process-to-process communication

The unit of data at this level is usually Called a "segment" – sometimes a "Protocol Data Unit (PDU)

Transport-layer protocols in the TCP/IP stack

- We'll discuss:
  - User Datagram Protocol (UDP)
    - Packet-oriented, Best effort
  - Transmission Control Protocol (TCP)
    - Byte-stream oriented, Reliable
- Note that there are others (covered in CS670)
  - Remote Procedure Call (RPC)
    - Request/Reply paradigm
  - Real-Time Protocol (RTP)
    - For real-time (e.g., multimedia) apps
  - Others added through RFC ballotting

UDP

UDP

- "IP with ports"
- Best effort (not reliable)
- connectionless
- Why a Best-Effort protocol (instead of a reliable one)?
  - Speed
  - Simplicity

The University Of Alabama in Huntsville Computer Science

UDP

cs570

16 bits	16 bits
Source Port	Dest Port
Checksum	Length
Payload	

- Features that give simplicity, speed
  - Connectionless, so no pre-transmit setup
  - Small headers
  - Unregulated flow rate, no error re-xmits (but application can do that)
  - "stateless" (segments are unrelated to previous segments)

UDP applications

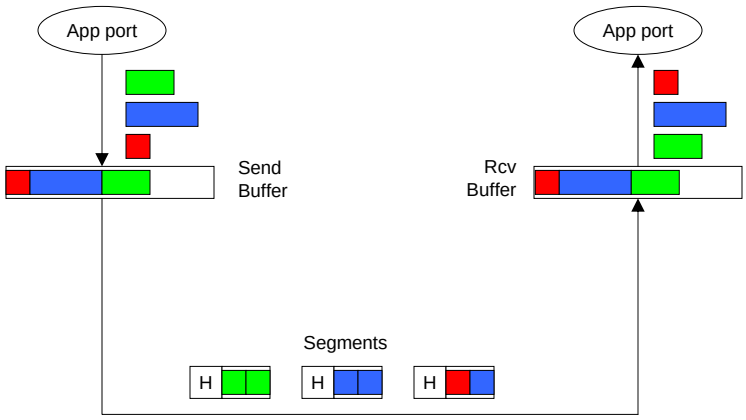
- Used for many apps where reliability is less critical than speed
  - Streaming multimedia
  - Internet telephony
  - DNS service

TCP

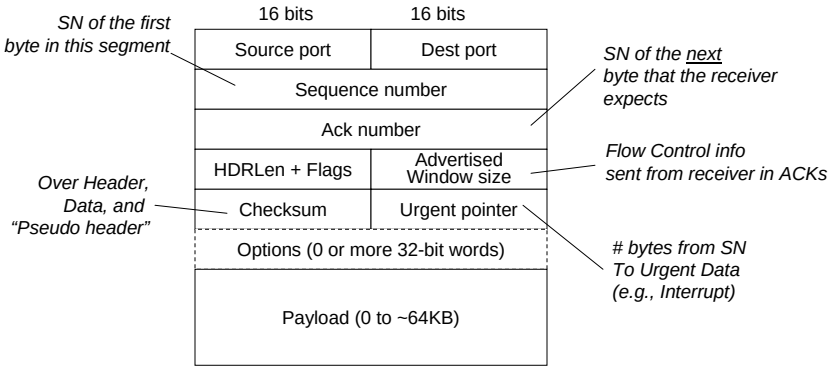
TCP

- TCP = Transmission Control Protocol
  - Reliable (Guarantees all bytes will be delivered, in-order, no errors)
  - Connection-Oriented
  - Designed for “byte-stream” data
- Includes:
  - Flow control (Sliding Window)
  - Congestion control (Discussed later)

Byte stream data



TCP header format



HDRLen + flags

4 bits	6 bits							
Header length	Unused	URG	ACK	PSH	RST	SYN	FIN

- Header length – # 32-bit words in header, including option fields, if any

- URG =1 if this segment contains Urgent Data (Urgent Pointer is valid)
- ACK =1 if this is an ACK segment (Acknowledgment Number is valid)
- PSH =1 if segment is to be delivered to receiving process immediately
- RST =1 error flag (invalid seg, connection refused,…)
- SYN Used when establishing a connection
- FIN Used to release a connection

"pseudo header"

Source IP address		
Dest IP address		
0000 0000	Protocol ID	TCP segment length

- Included in Checksum to help detect misrouted, mis-synched segments – if the address was changed en-route, the receiver would see a checksum error
- Also used in calculating Checksum for UDP
- Note that this violates the normal assumption of independence of protocol layers, but it's needed to catch mis-routed packets

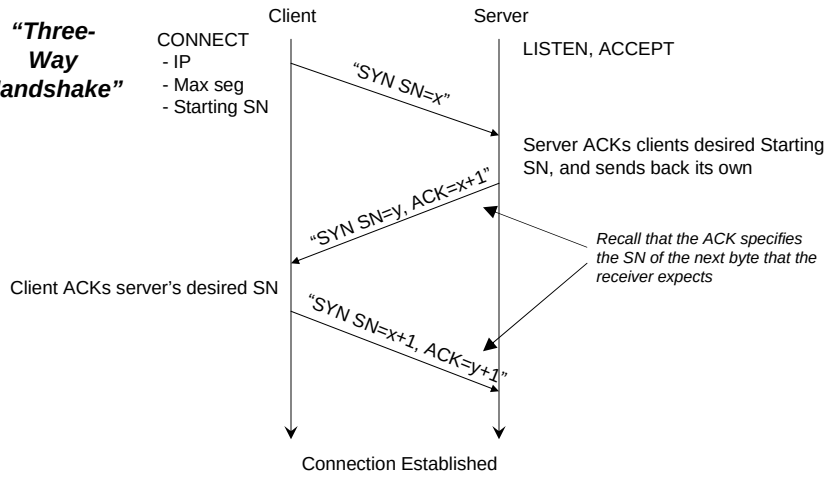
TCP options field

- Used to extend the protocol
- Some examples:
  - Specify max segment size you will accept (during setup)
  - Specify a scaling factor for the window size field

Establishing a TCP connection

The University Of Alabama in Huntsville Computer Science

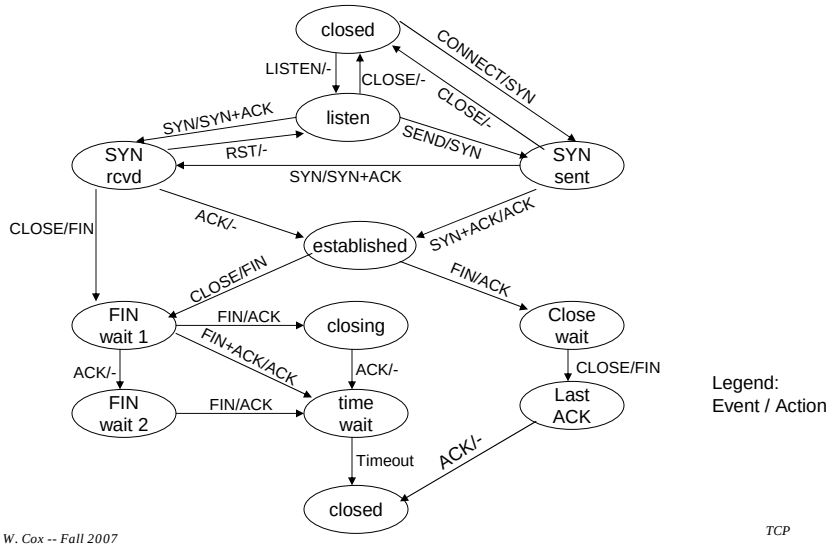
cs570



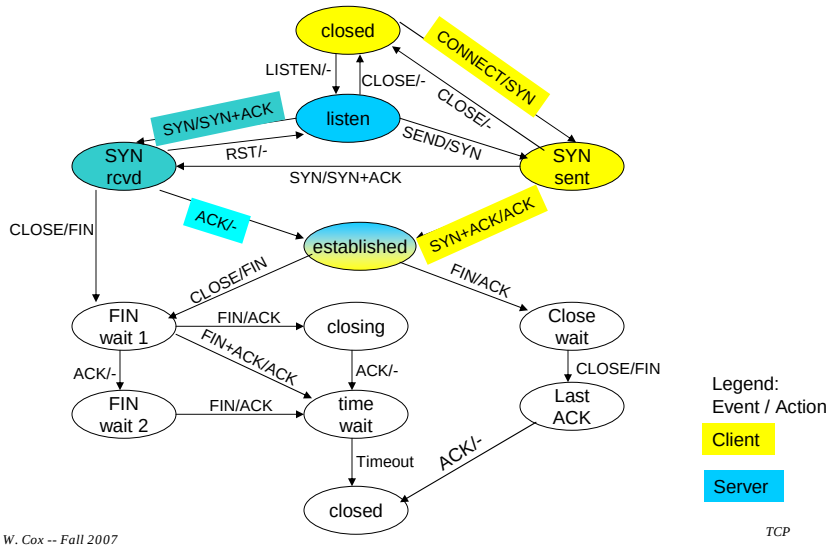
TCP connection management

The University Of Alabama in Huntsville Computer Science

cs570



Opening: 3-way handshake



Sliding Window Algorithm -- refresher

- Guaranteed delivery
  - All frames ACKed
  - Timeouts to detect lost frames, ACKs
  - No window movement until preceding block of frames delivered
- In order
  - All frames numbered
  - Receiver inserts frame in correct position, according to number
  - No ACK until frames in order
- Flow Control
  - Send rate regulated by send window size
  - No window movement until receiver ACKs

Extending SWA for TCP: Dynamic window size

- In each ACK, receiver specifies a max Send Window size "Advertised Window Size"
- This allows receiver to regulate the flow dynamically according to the traffic it is able to handle at any moment

Extending SWA for TCP: Larger sequence numbers

- In a network with high-speed links, a small SN field can roll over quickly
- That might be OK if all data was delivered quickly, but big networks can have big delays – segments can be as late as the IP TTL (120 sec, typ)
- Recall that if there are two items in flight with the same SN, the SWA algorithm can fail
- The fix: 32-bit sequence numbers

Extending SWA for TCP: Adaptive timeouts

- In a big network, response times can vary widely from moment to moment – this complicates the strategy of using RTT to set the timeout time
- The fix:
  - Sender keeps running average of time between SEND and ACK to each receiver
  - Timeout adjusted dynamically according to current average (for example, timeout = 2 x current_RTT)

How does TCP decide when to send a segment?

1. When a predetermined number of bytes have been accumulated in the send buffer (negotiated seg size)
2. On demand from higher level (e.g, to send urgent data)
3. On a time basis (so that segs are not delayed waiting on data from a slow-talking application)

A final note

- TCP has a surprising number of parameters and control options
- To learn more, read Peterson 5.2.
- To learn still more, take CS670