Electronics and Computer Science
University of Southampton

CM214-COMP2008
Data Communications and Networks

Protocols Over IP

Karl R. Wilcox
krw@ecs.soton.ac.uk

Objectives

- To look at higher level protocols that make use of the “lowest common denominator” packet delivery of IP
  - UDP
  - TCP

- (Peterson & Davie, Section 5.1, 5.2)

Review

- In the last lecture we looked at IP
- "Best Effort" packet delivery
  - No guarantee of delivery, uniqueness or ordering
- Datagram (connectionless) protocol
  - All packets are independent

User Datagram Protocol

- Very simple demultiplexer
  - Allows more than one process on each host to use network connections
- Introduces the concept of “ports”
  - Up to 64K different ports per host
- Adds an additional (UDP) header containing:
  - Source port number
  - Destination port number
  - Length of data and checksum

UDP Characteristics

- Adds only demultiplexing
- Packet delivery is still:
  - Best efforts
  - No guaranteed delivery, order or uniqueness
- The Application must sort out any problems
  - E.g. Early FTP (File Transfer Protocol) programs use UDP for data transfer
  - Retransmission etc. was handled by the application program

Transmission Control Protocol

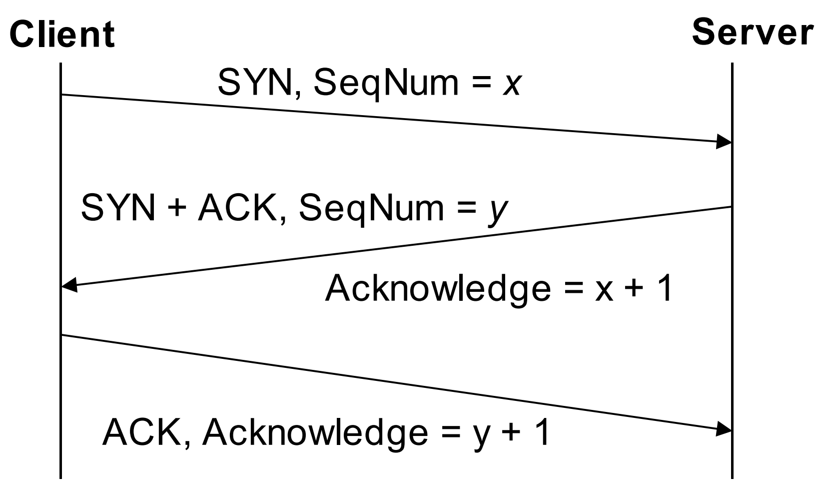
- A Transport Protocol offering:
  - A connection oriented service
  - A byte-stream interface
  - Reliable connections (guaranteed delivery)
  - Flow control
- When run over IP, known as TCP/IP

TCP Header

0	4	10	16	31
Source Port			**Destination Port**	
Sequence Number				
Acknowledgement Number				
Hdr Len	**0**	**Flags**	**Advertised Window**	
Checksum			**Urgent Pointer**	
Options (variable length)				
Data … (variable length)				

TCP Connections

- TCP is connection oriented
  - There must be an explicit set up phase to create an end-to-end connection
  - Uses a three way handshake



Full Duplex Connection

- TCP provides <u>Full Duplex</u> connections
- Both ends of the connection can simultaneously send & receive
  – Hence separate sequence numbers for each direction
- Acknowledgements are “piggy-backed” on sent packets
  – Or placed on an empty “dummy” packet

TCP Setup / Closedown

- In reality the handshaking is more complex, due to:
  - Possible loss of packets in the network
    - Different response to loss of each handshake
  - There is also an explicit close down phase
    - Either side can initiate close down
    - Different sequence of events depending on which side closed first and whether the other side continues to send packets

Byte Stream Interface

- Applications do not have to be concerned with packet sizes
  - They can write as few or as many bytes, as often as required
  - They can read as few or as many bytes are available, as often as required
- The Transport layer must provide input and output buffering

When to Send Packets

- A packet will be sent when
  - The minimum segment size (MSS) amount of data is available
    - Usually set to the maximum packet size of the local network minus the header
  - A timeout occurs
  - The application calls the “push” operation
    - Also forces acknowledgement from receiver
    - Therefore can be used to mark boundaries

Reliable Connections

- TCP provides reliable delivery
  - Check-summed
  - in order sent
  - No missing data
- Uses a “sliding window” algorithm
  - 32 bit sequence number of bytes
  - Receiver acknowledges every packet
    - Using ACK flag in header

Flow Control

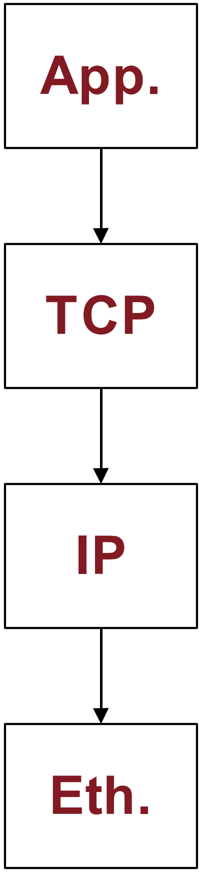
- Sliding window sized is NOT fixed
- Return packets contain “advertised window” field (in bytes)
  - Sender should not have more unacknowledged bytes than advertised window size
  - Advertised window may be zero
    - Sender responsible for retrying
  - Implements end-to-end flow control

Retransmission

- TCP sets a time out period before assuming a packet is lost (i.e. not acknowledged)
  - Timeout is adaptive, dependent on estimated round trip time (RTT)
- After timeout, packet is retransmitted
- Several RTT estimation algorithms
  - To take account of congestion, lost packets

TCP Applications

- Telnet
  - Uses “push” function to flush packets quickly
- HTTP
- SMTP
- POP3
- Routing protocols
- Client / Server programs in general



Summary

- TBD