



CS118 Discussion Week 3

Taqi

Outline

- Lecture Review: Transport Layer

Transport Layer V.S. Network Layer

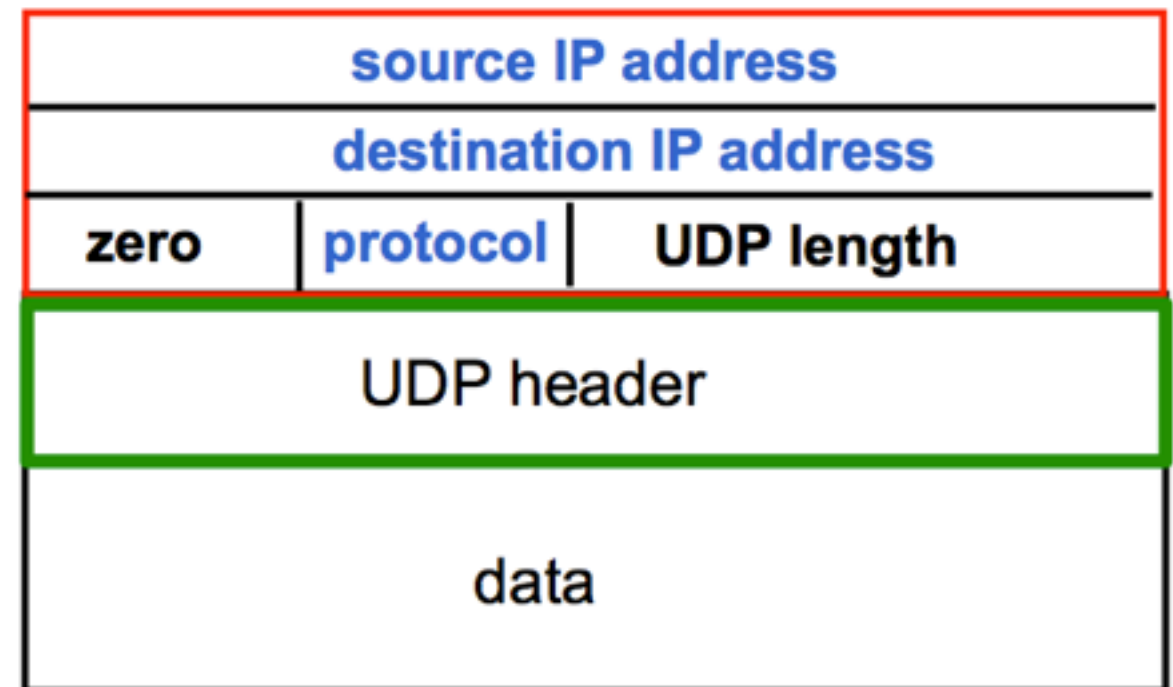
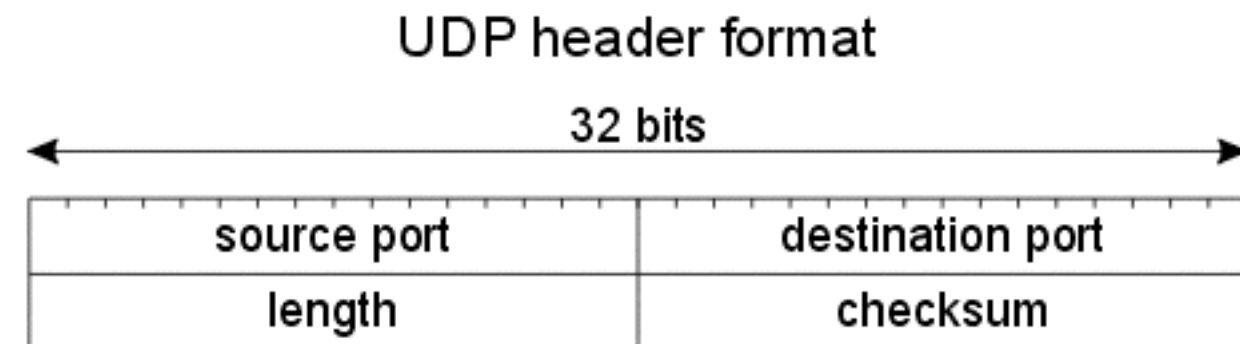
- Network layer: logical communication between **hosts**
 - **IP address** is used for identifying a host
- Transport layer: logical communication between **processes**
 - **IP address and port number** are used for identifying a process

Multiplexing and De-multiplexing

- Multiplexing at send host: gather data from multiple sockets
- De-multiplexing at receiving host: deliver received segments to the right socket
- **Five tuples** (src_ip, src_port, dst_ip, dst_port, protocol) are used for multiplexing/demultiplexing
 - How to identify a TCP/UDP socket? **Isot -i**
 - Can TCP and UDP share the same port numbers? **Yes! e.g. DNS**

UDP

- No connection establishment
- No connection state
- Small packet overhead (8 byte)
- How to calculate checksum?
 - **Pseudo header** + **UDP header** + data
 - Also applicable to TCP
 - Why pseudo header?



Principles of Reliable Data Transfer

- How to deal with bit errors?
 - Error detection (e.g. checksum)
 - Receiver feedback
 - Retransmission
 - Why not error correction?
- How to deal with duplicate packets due to retransmission? **Sequence number**
- How can the sender detect that ACK or data is lost? **Timer**

Stop and Wait Protocol

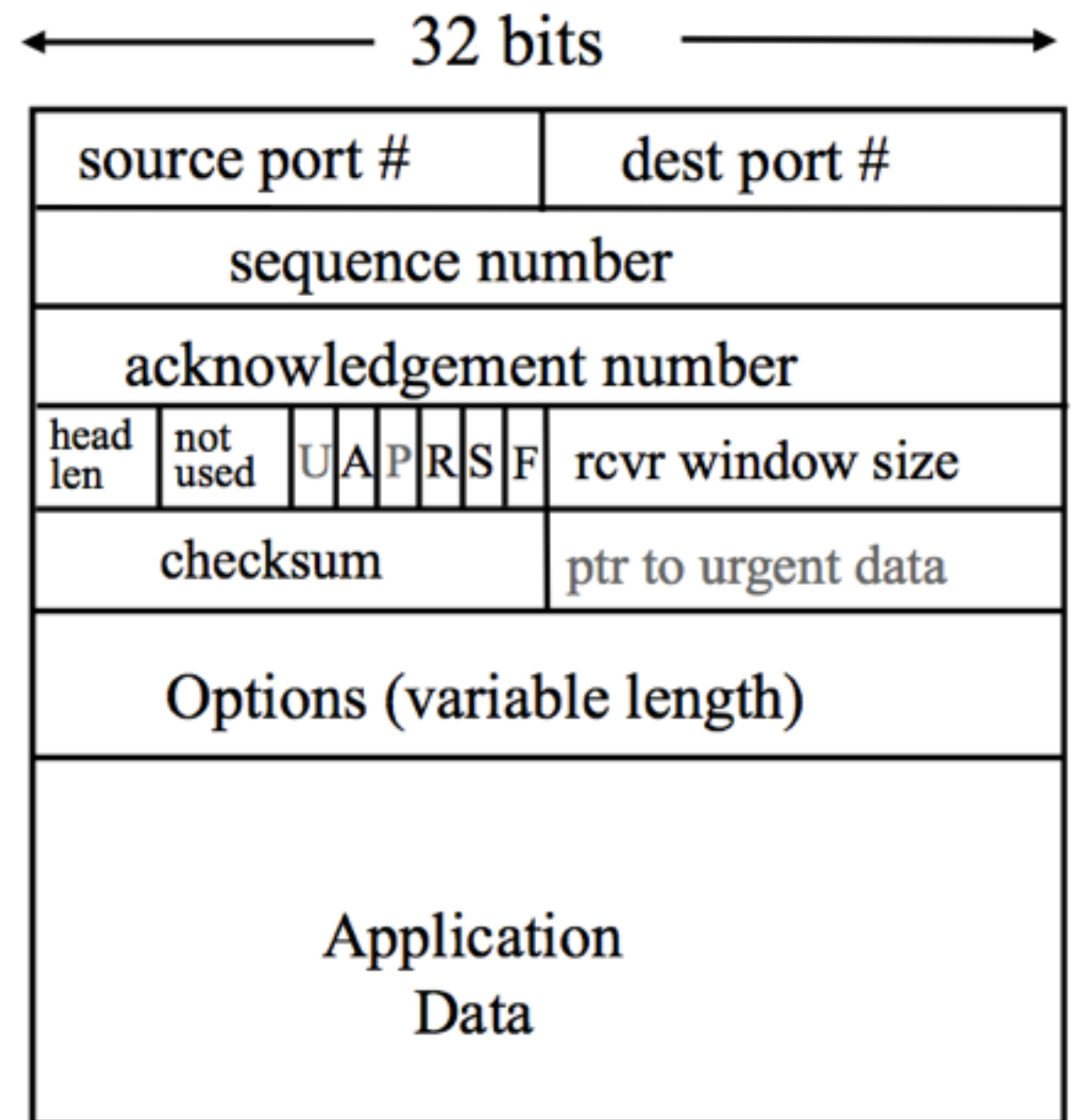
- Main Issue: **limited performance**
- Consider two hosts that are directly connected by a 50 Kbit/sec satellite link that has a 250 milliseconds propagation delay. If these hosts send 1000 bits segments, what is the maximum throughput in stop-and-wait protocol if we ignore the transmission time of ACK?
 - Ans: $1000 / (1000 / 50 + 250 + 250) = \mathbf{2 \text{ Kbit/sec!}}$

Pipelined Protocols

- Go-back-N: receiver only sends cumulative ACKs
 - Drop out-of-order segments
 - reACK packet with highest in-order sequence number
 - Timer for oldest unACKed packet only, retransmit all unACKed packets
- Selective repeat: receiver ACKs individual packets
 - Buffer out of order segments
 - Timer for each individual unACKed packet, retransmit any unACKed packet

TCP

- Point-to-point, byte-stream reliable transport protocol
 - **Multiplexing/de-multiplexing:** Source/Dest port
 - **Reliable data transfer:** sequence number, ack, checksum, RTT estimation
 - **Connection setup:** sequence number, SYN, receive window
 - **Connection teardown:** sequence number, FIN



TCP Connection Management

- Connection setup: three-way handshaking
 - 1st round: SYN+initial sequence number
 - 2nd round: SYN+SYN-ACK+server's initial sequence number
 - 3rd round: SYN-ACK+ACK+(optional)data
- Connection Teardown
 - “Half-closed” connection
 - Why timed wait?

Comparison of Reliable Transport Protocol

Protocol	Buffer at sender	Buffer at receiver	ACK	Timeout/Retransmission
Stop & Wait	No	No	No out-of-order	Retransmit timeout packet
Go-Back-N	Yes	No	Accumulative Seq#	Retransmit all packets in window
Selective Repeat	Yes	Yes	Received Seq#	Retransmit timeout packet
TCP	Yes	Yes	Next expected Seq#	Retransmit timeout packet

Resources

- https://courses.engr.illinois.edu/cs438/sp2010/slides/lec04_reliable.pdf
- Demo on Go-Back-N and Selective Repeat protocols.
http://www.ccs-labs.org/teaching/rn/animations/gbn_sr/