ACIT 2620

Principles of Enterprise Networking

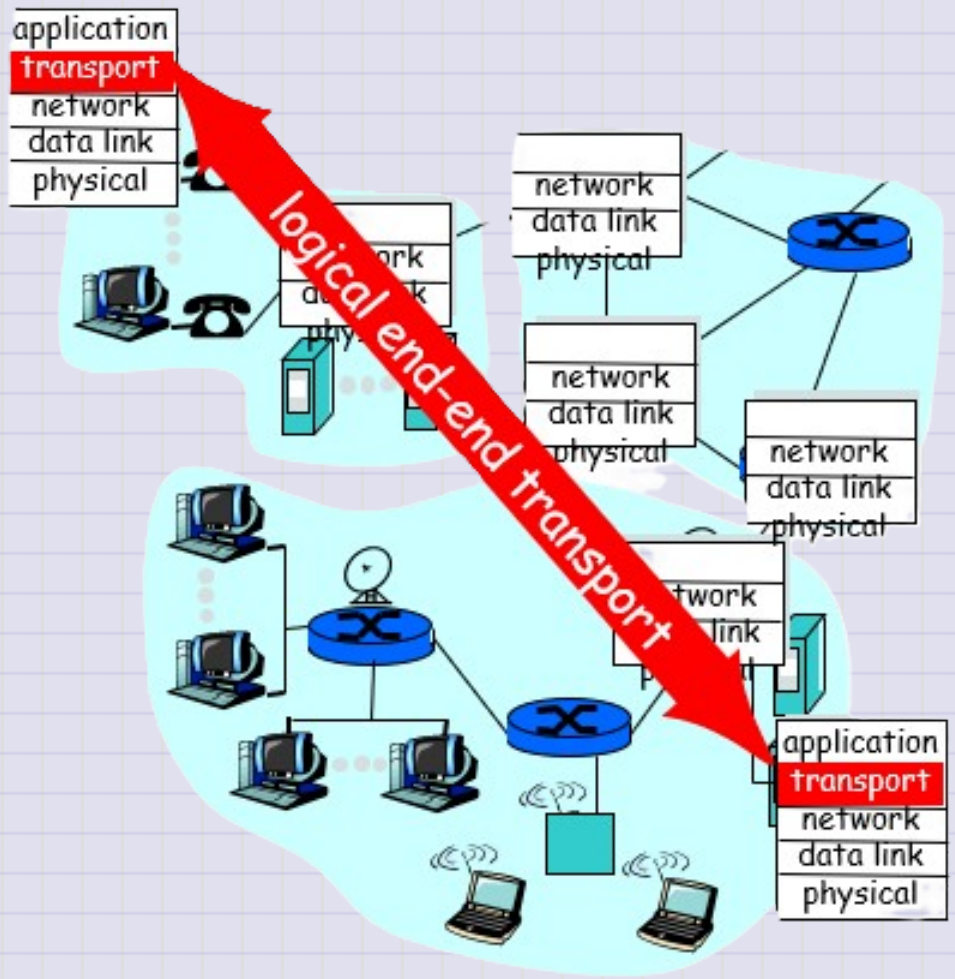
By: Yves Rene Shema

Transport Layer Protocols

TCP & UDP

Transport Services and Protocols

- provide logical communication between app' processes running on different hosts
- transport protocols run in end systems
- network layer: data transfer between end systems
- transport layer: data transfer between processes
    - relies on, enhances, network layer services



Transport-layer protocols

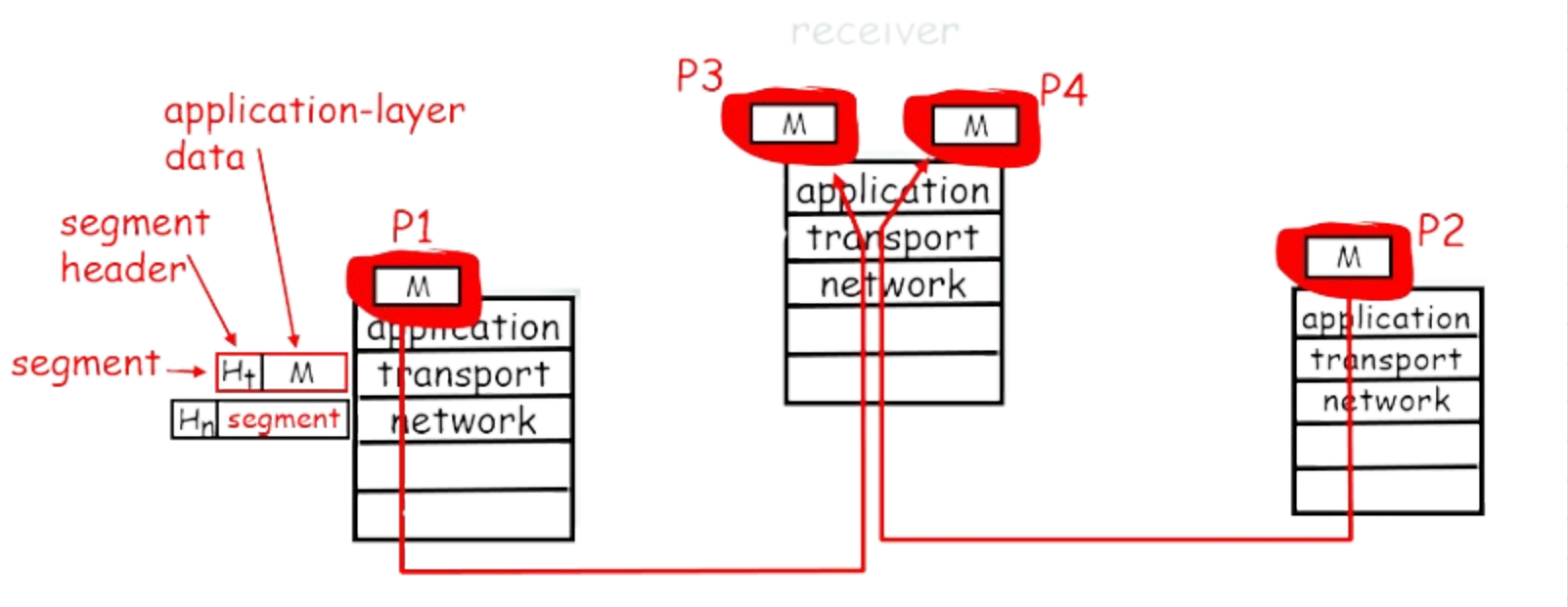
Internet transport services

- reliable, in-order unicast delivery (TCP)
  - congestion
  - flow control
  - connection setup
- unreliable (“best-effort”), unordered unicast or multicast delivery: UDP
- services not available:
  - real-time
  - bandwidth guarantees
  - reliable multicast

Multiplexing / Demultiplexing: What

segment - unit of data exchanged between transport layer entities

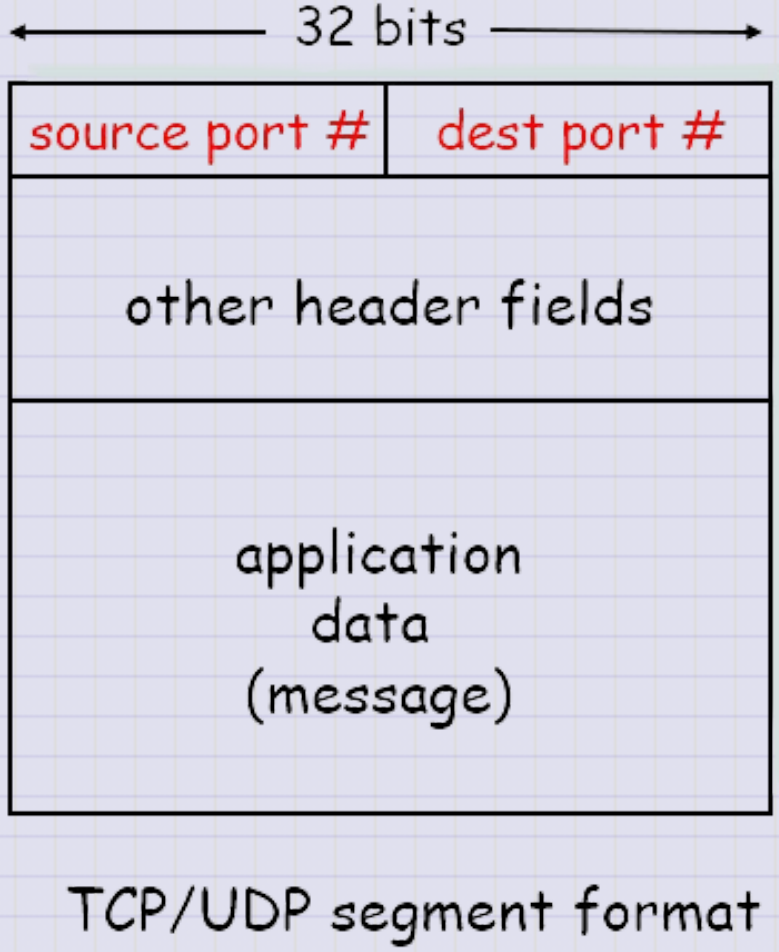
Demultiplexing - delivering received segments to correct app layer processes



Multiplexing / Demultiplexing: How

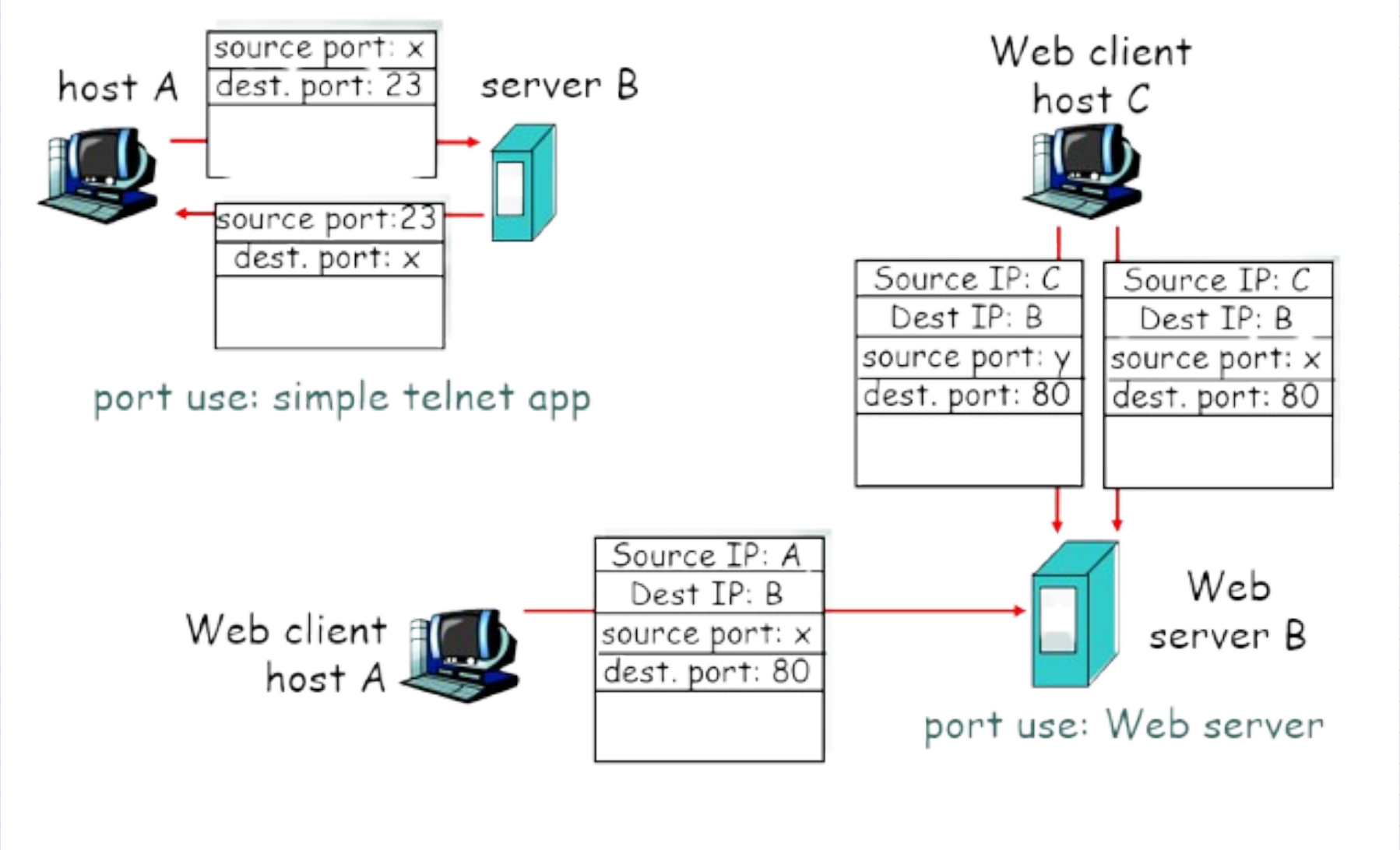
Multiplexing - gathering data from multiple app processes, enveloping data with header (later used for demultiplexing)

- based on sender, receiver port numbers, IP addresses
  - source, destination port #s in each segment
  - well-known port numbers for specific applications



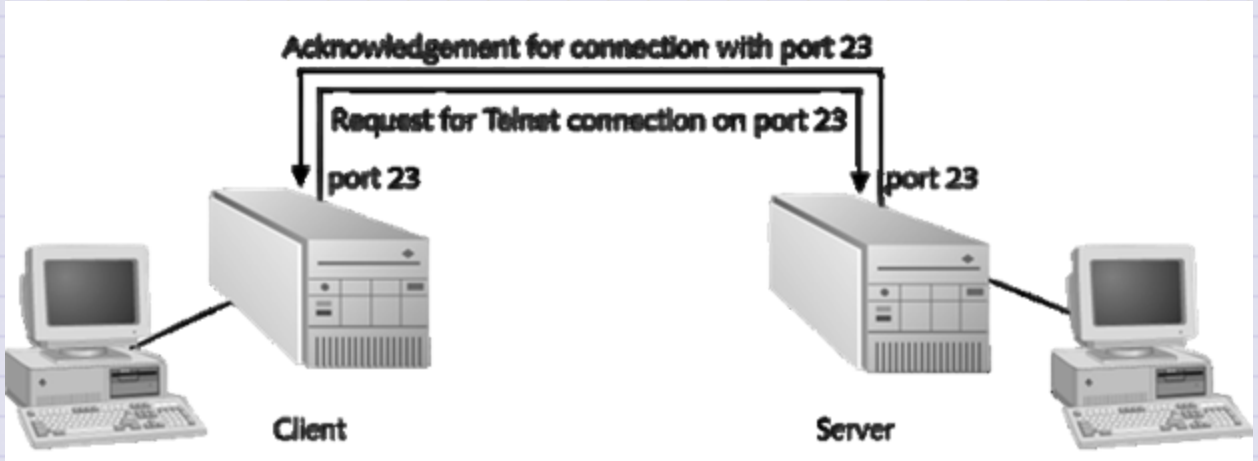
TCP/UDP segment format

Multiplexing / Demultiplexing: Examples



Socket

- Logical address assigned to a specific process running on a host computer
- The socket's address combines the host computer's IP address with the port number associated with a process



Port Numbers

- 0 to 1023 : “well-known” ports
  - systems restrict to use by privileged processes only.
  - IANA controls the mapping of service names to port numbers
- 1024 to 49151 “registered” ports
  - systems permit ordinary user processes to use.
  - IANA maintains a mapping of names to port numbers but does not exert control over them
- 49152 to 65535 : “dynamic” or “private” ports
  - are not subject to IANA registration.
- http://www.iana.org/assignments/port-numbers
- `/etc/services` – list of ports and services

UDP: User Datagram Protocol [RFC 768]

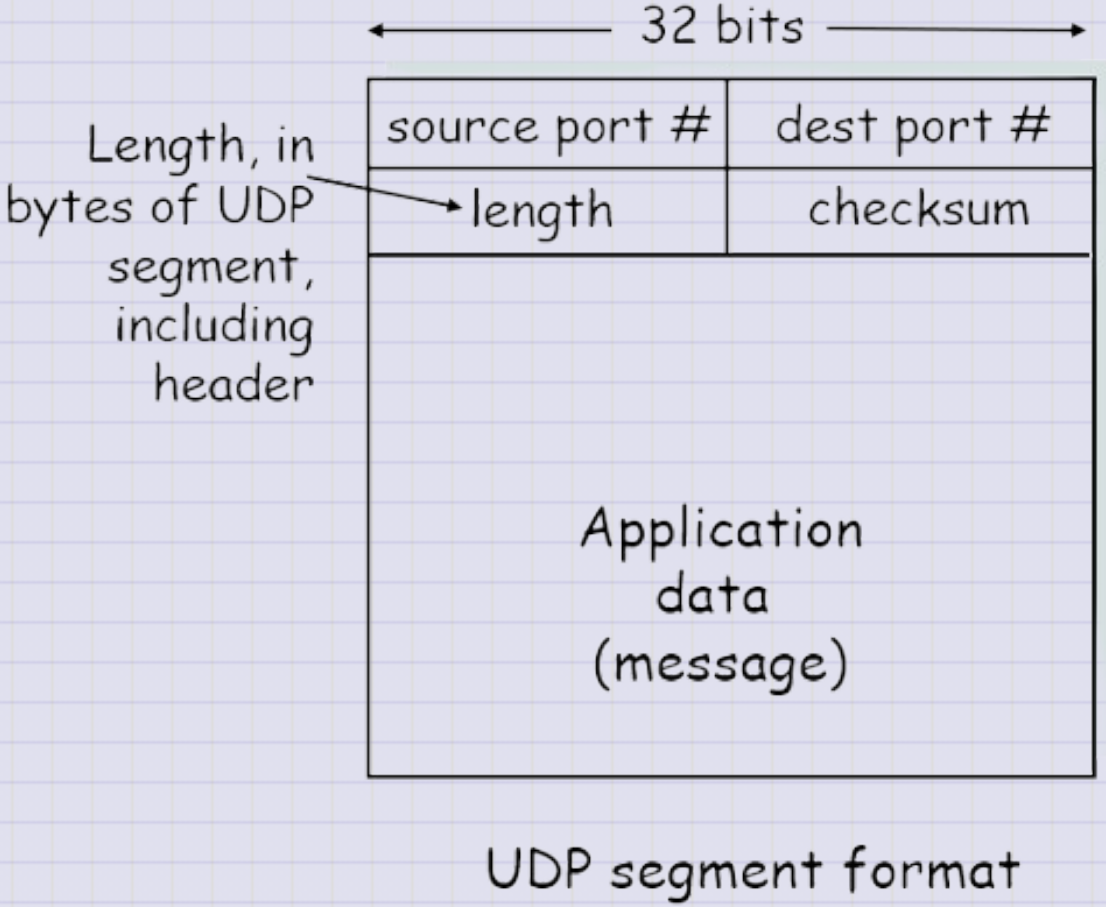
- “no frills,” “bare bones” Internet transport protocol
- “best effort” service, UDP segments may be:
- lost
- delivered out of order to app
- **connectionless**:
  - no handshaking between UDP sender, receiver
  - each UDP segment handled independently of others

Why is there a UDP?

- no connection establishment (which can add delay)
- simple: no connection state at sender, receiver
- small segment header
- no congestion control: UDP can blast away as fast as desired

UDP Uses and Format

- often used for streaming multimedia apps
  - loss tolerant
  - rate sensitive
- other UDP uses
  - DNS
  - SNMP
- reliable transfer over UDP: add reliability at application layer
  - application-specific error recovery



UDP segment format

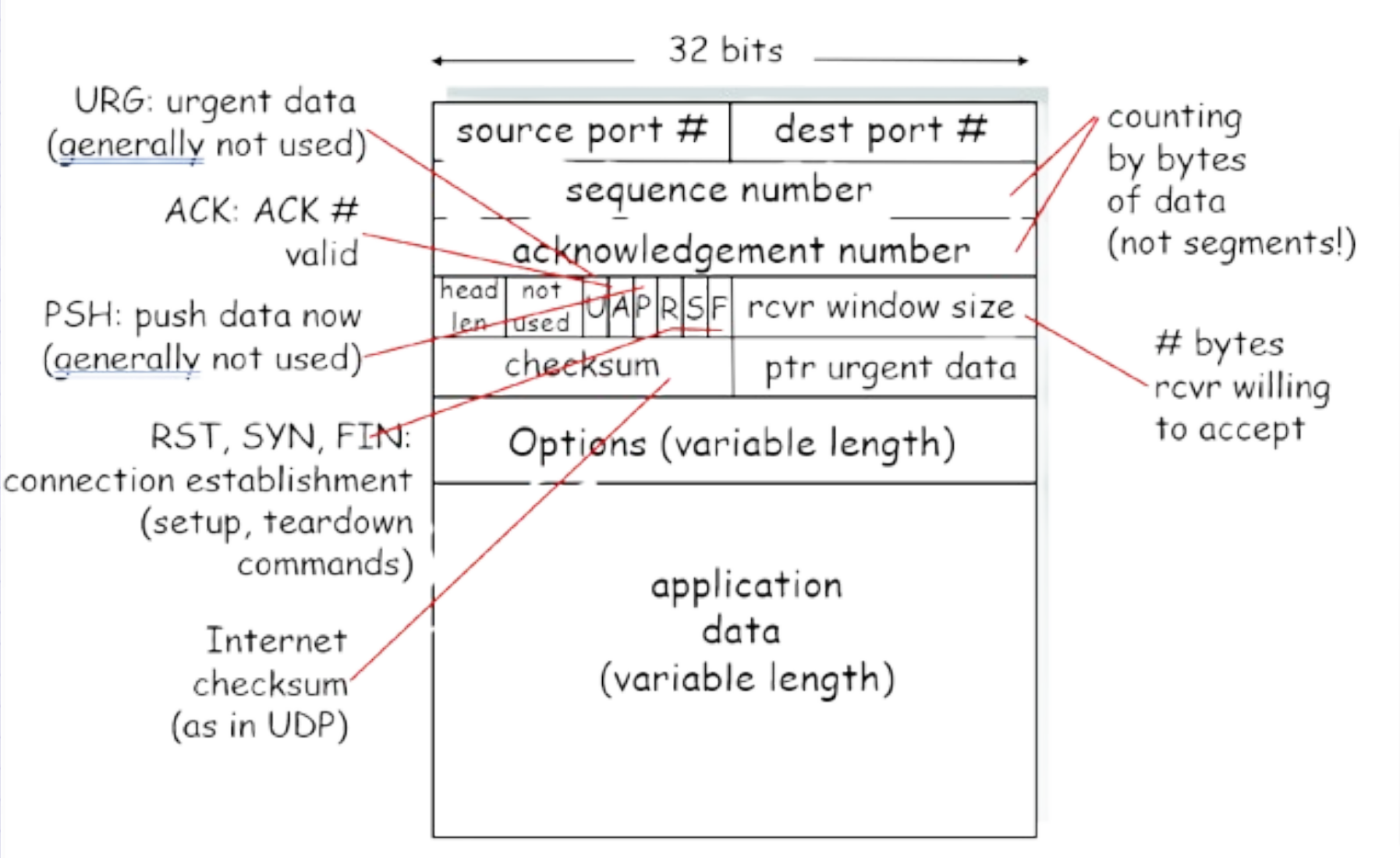
TCP: Overview RFCs: 793, 1122, 1323, 2018, 2581

- **point-to-point:**
  - one sender, one receiver
- **reliable, in-order byte stream:**
  - no “message boundaries”
- **pipelined:**
  - TCP congestion and flow control set window size
- **send & receive buffers**

TCP: Overview (cntd)

- **full duplex data:**
  - bi-directional data flow in same connection
  - MSS: maximum segment size
- **connection-oriented: **
  - handshaking (exchange of control msgs) init's sender, receiver state before data exchange
- **flow controlled:**
  - sender will not overwhelm receiver

TCP segment structure



TCP Sequence #’s and Acknowledgements

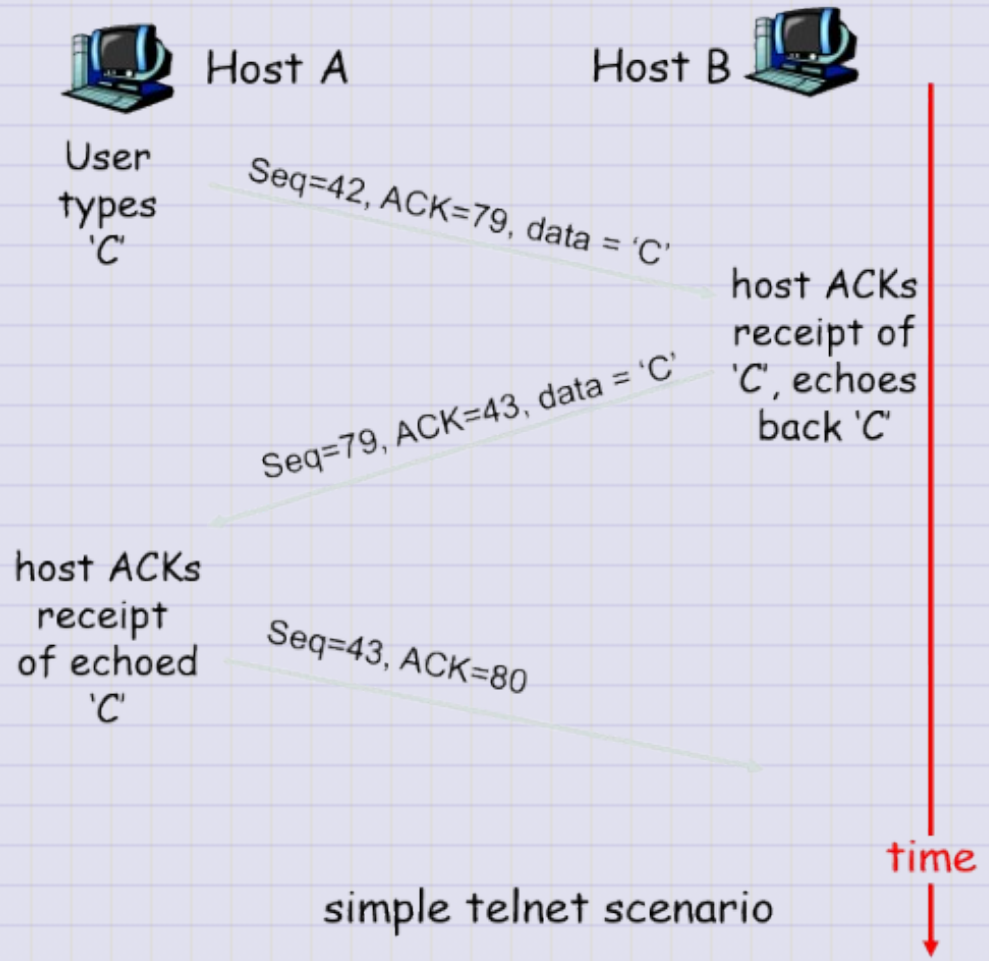
Seq. #’s:

- byte stream “number” of first byte in segment’s data

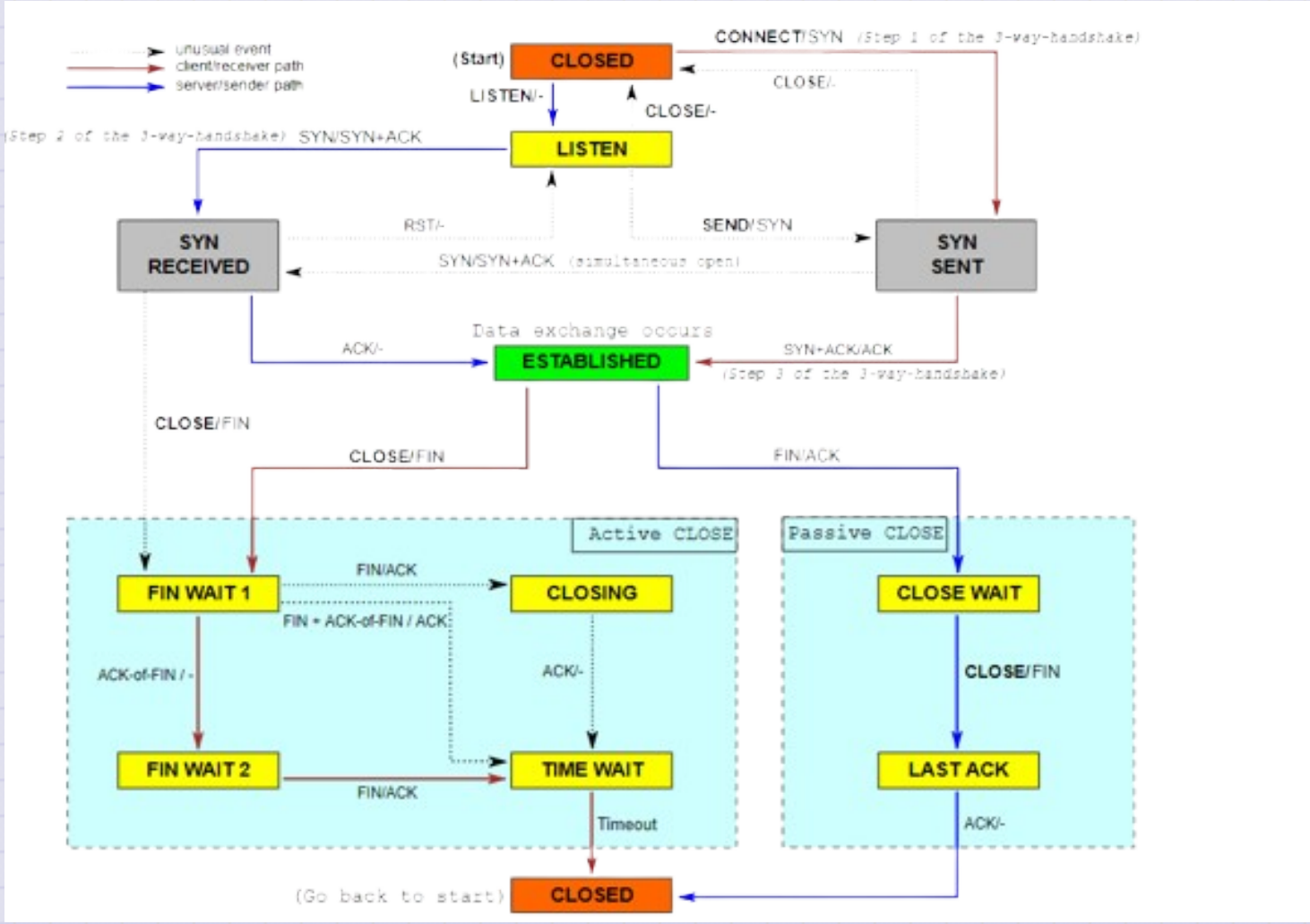
ACKs:

- seq # of next byte expected from other side
  cumulative ACK

TCP spec doesn’t say how receiver handles out-of-order segments



TCP in action: States



TCP States in Action

```
#ss -tn
State       Recv-Q Send-Q        Local Address:Port                 Peer Address:Port
ESTAB       0      0             142.232.107.85:43420             216.58.193.78:443
FIN-WAIT-1  0      1             142.232.107.85:40359             142.232.2.1:80
ESTAB       0      0             142.232.107.85:34221             142.232.204.52:445
ESTAB       0      0             142.232.107.85:48256             142.232.204.50:445
SYN-SENT    0      1             142.232.107.85:41974             142.232.2.1:10123
ESTAB       0      0             172.16.100.6:47180               10.0.255.26:22
FIN-WAIT-1  0      1             142.232.107.85:54207             23.58.117.226:80
ESTAB       0      0             142.232.107.85:36948             142.232.204.52:445
```

Reading List

- Managing firewall with nftables