CS 655
Computer Networks

Abraham Matta
Computer Science
Boston University

Chapter 3
Transport Services

Computer Networking: A Top-Down Approach, 7th or 8th editions. Jim Kurose, Keith Ross. Pearson.

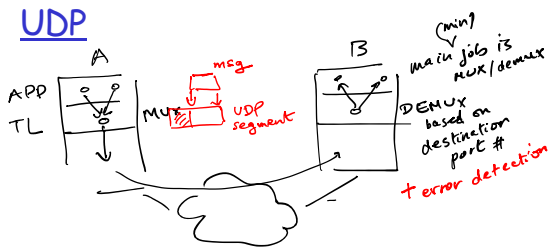
1

Transport Protocols in TCP/IP

- UDP (User Datagram Protocol)
  - provides applications with ***unreliable connectionless*** service
- TCP (Transmission Control Protocol)
  - provides applications with ***reliable connection-oriented*** service

2

UDP



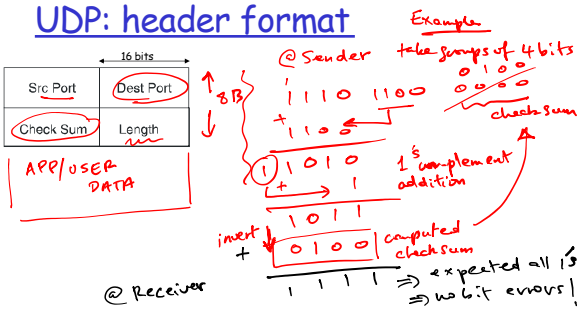
3

UDP

- Unreliable and unordered datagram service
- Adds multiplexing and checksum (covers user data)
- Endpoints identified by ports
  - servers have ***well-known*** ports
  - see /etc/services on Unix/Linux

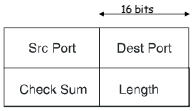
4

UDP: header format



5

UDP: header format

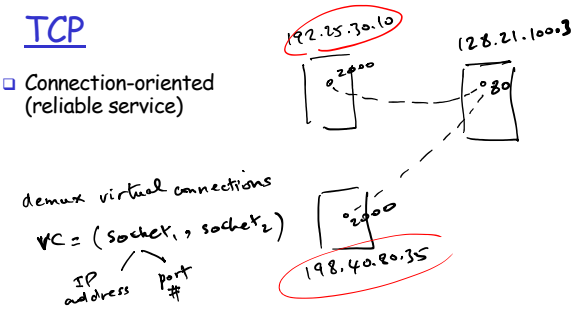


- UDP demultiplexes based on **destination port** in the UDP header
- Checksum idea: view message as a sequence of 16-bit integers. Add these integers together using 16-bit ones complement arithmetic, and then invert the result. That 16-bit number is the checksum
  - Receiver adds in ones complement and should get all 1's if no bit errors

6

TCP

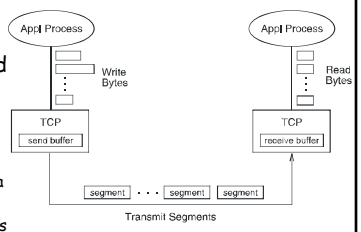
- Connection-oriented (reliable service)



7

TCP

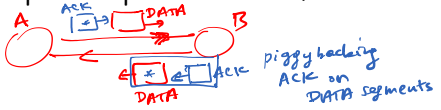
- Connection-oriented (reliable service)
- Byte-stream buffered transfer
  - sending process writes some number of bytes
  - TCP breaks into ***segments*** and sends via IP
  - receiving process reads some number of bytes



8

TCP

- Full duplex point-to-point data transfer



- Flow control: keep sender from overrunning receiver
- Congestion control: keep sender from overrunning network

9

Basics of Reliable Transmission

- Building it from the ground up over a FIFO lossy & erroneous (unreliable) unidirectional channel

10

Basics of Reliable Transmission

Assume FIFO channel but erroneous ⇒ corrupted or lost packets

1) avoid losses @ receiver ⇒ flow control: one packet at a time (stop-and-wait) ⇒ "ACK's

error detection

2) detect corrupted packets ⇒ "checksum" fails ⇒ drop

3) detect lost packets ⇒ timer ≈ RTT "RTO"

error recovery

4) recovery ⇒ retransmit once RTO expires (ARQ) "idle RQ"

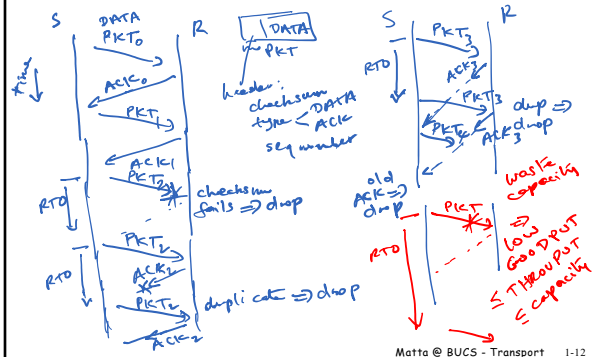
"error control"

5) detect duplicates ⇒ sequence numbers

No out-of-order delivery here because FIFO channel

11

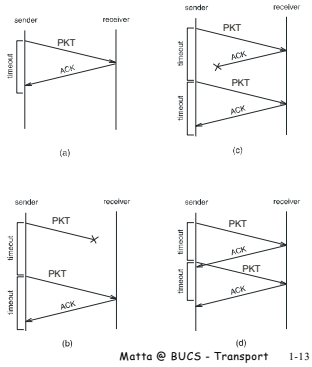
Basics of Reliable Transmission



12

Basics of Reliable Transmission

- Recover from corrupted/lost packets using Acknowledgements and Timeouts; also called Automatic Repeat reQuest (ARQ)



13

Basics of Reliable Transmission

- Building it from the ground up over a FIFO lossy & erroneous (unreliable) unidirectional channel
- Add flow control (stop-and-wait)
- Dealing with corruption – add checksum
- Dealing with loss – add timer
- Add recovery by retransmission (ARQ)
- Dealing with duplicates – add sequence numbers

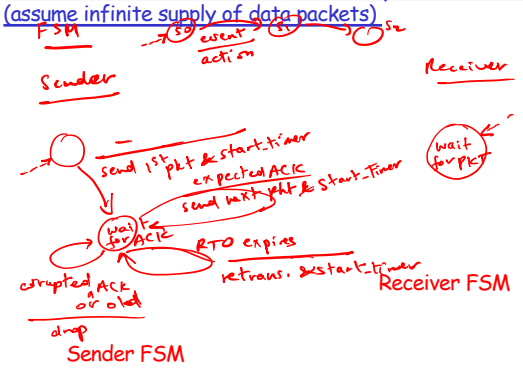
14

ARQ

- The packet's header contains the control information needed to implement ARQ (type, sequence number, checksum, ...)
- At the receiver, only the data portion of an error-free data packet is delivered to the higher layer
- Sequence numbers are needed for detecting duplicates (and more ... later!)
- A good timeout estimate is essential to avoid premature retransmissions *and* maintain high goodput (rate of receiving new/useful data)

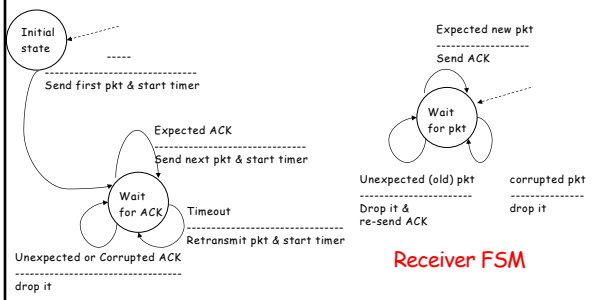
15

Finite State Machines for Stop-and-Wait
(assume infinite supply of data packets)



16

Finite State Machines for Stop-and-Wait
(assume infinite supply of data packets)



17