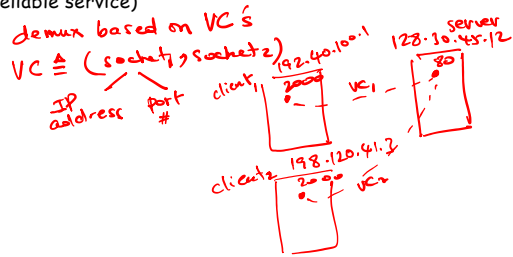


TCP

- Connection-oriented (reliable service)



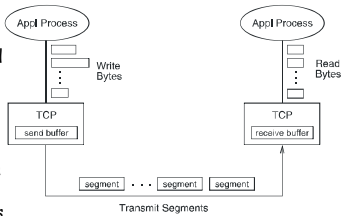
Matta @ BUCS - Transport 1-7

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

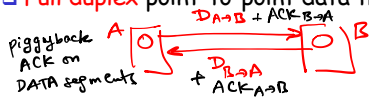


Matta @ BUCS - Transport 1-8

8

TCP

- Full duplex point-to-point data transfer



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

Matta @ BUCS - Transport 1-9

9

Basics of Reliable Transmission

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

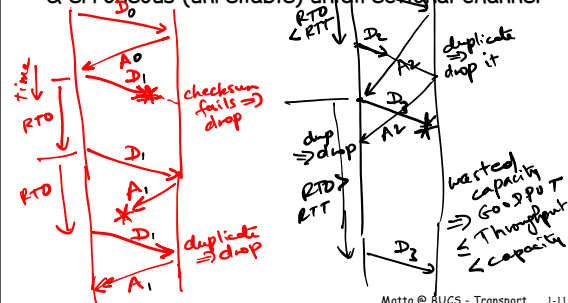
- error control
- 1) No loss @ receiver \Rightarrow flow control \Rightarrow one packet at a time & wait for ACK (stop-and-wait)
 - 2) detect corrupted packets \Rightarrow checksum if checksum fails \Rightarrow drop
 - 3) detect lost packets \Rightarrow sender timer $RTO \approx RTT$ expires \Rightarrow loss
 - 4) recover corrupted/lost packets \Rightarrow retransmit data packet ("idle" ARQ)
 - 5) detect duplicates \Rightarrow seq no.
 - 6) in-order delivery comes for free (S&W)

Matta @ BUCS - Transport 1-10

10

Basics of Reliable Transmission

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

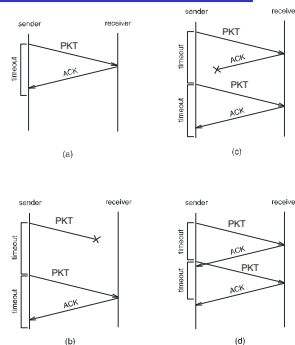


Matta @ BUCS - Transport 1-11

11

Basics of Reliable Transmission

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



Matta @ BUCS - Transport 1-12

12

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

Matta @ BUCS - Transport 1-13

13

ARQ

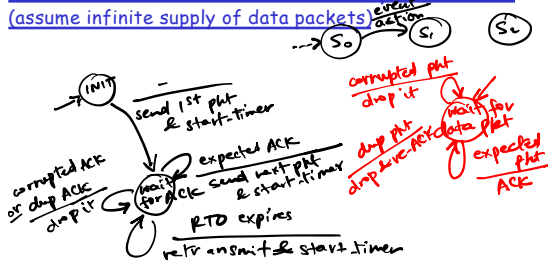
- The packet's header contains the control information needed to implement ARQ (type, sequence number, checksum, ...)
- At the receiver, only the data portion of an error-free data packet is delivered to the higher layer
- Sequence numbers are needed for detecting duplicates (and more ... later!)
- A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

Matta @ BUCS - Transport 1-14

14

Finite State Machines for Stop-and-Wait

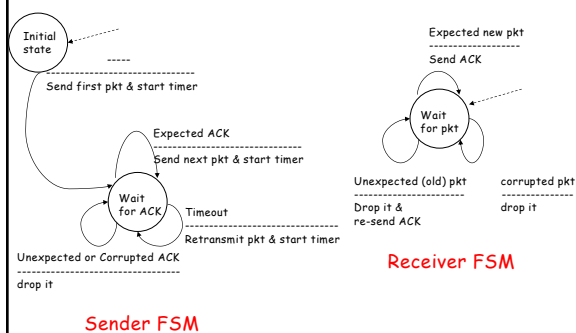
(assume infinite supply of data packets)



Matta @ BUCS - Transport 1-15

15

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



Matta @ BUCS - Transport 1-16

16

Basics of Reliable Transmission

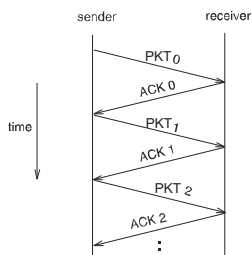
- Building it from the ground up over a FIFO lossy & erroneous (unreliable) unidirectional channel
- Add flow control (stop-and-wait)
- Dealing with corruption - add checksum
- Dealing with loss - add timer
- Add recovery by retransmission (ARQ)
- Dealing with duplicates - add sequence numbers
- Formally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

Matta @ BUCS - Transport 1-17

17

Stop-and-Wait (Idle RQ)

- Sender sends only one packet at a time

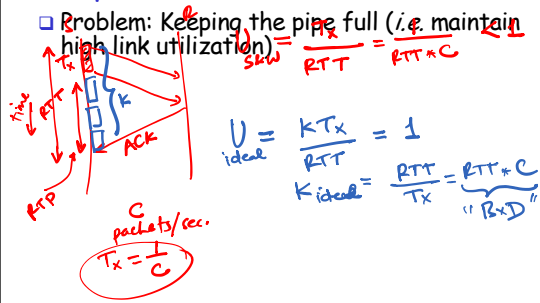


Matta @ BUCS - Transport 1-18

18

Stop-and-Wait (cont'd)

- Problem: Keeping the pipe full (i.e. maintain high link utilization)



Matta @ BUCS - Transport 1-19