

hey!

LPE ERaff

.

Definitions

Communication Protocols

Routing

→ Group Communication (multicast)

Mathematical Models

Automata

Hybrid Automata

Computation
Logic

Programming languages

LPEs

Birman "Building Reliable and Secure Network Applications"
Manning 1996

Next week Chapter 13.10 - 13.12

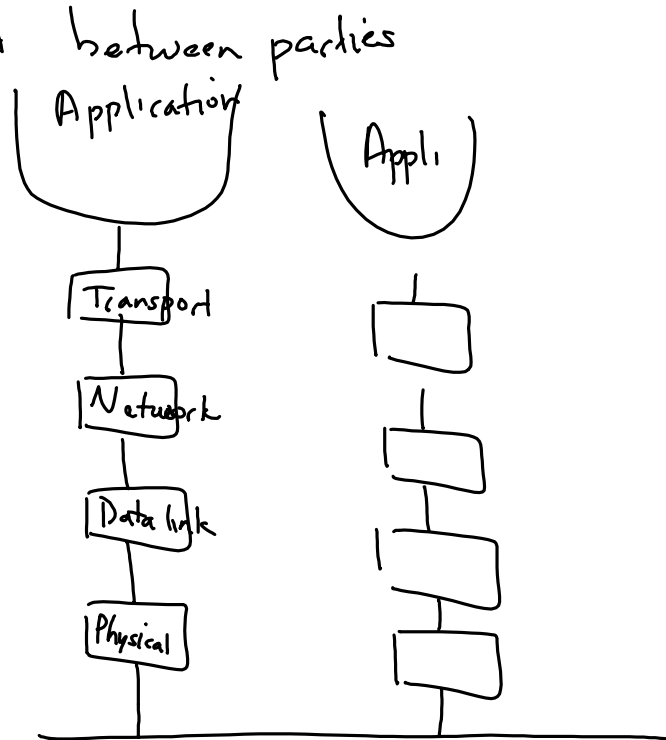
Tanenbaum "Computer Networks"
Chapter 2.5

Tanenbaum "Distributed Operating Systems"
Prentice Hall 1995 Chapter 2

Physical layer: how to send
0s and 1s - electrical volages
Connectors, ---

Data link: framing (packets)
error detection

Network: routing



Network: routing

Transport: Reliability

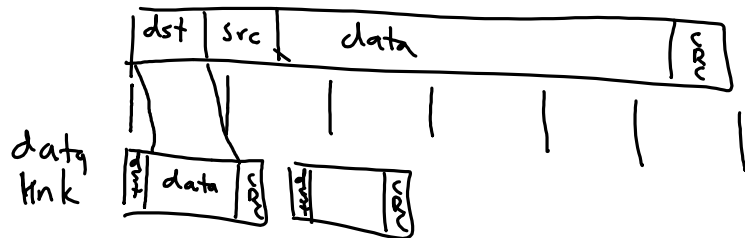
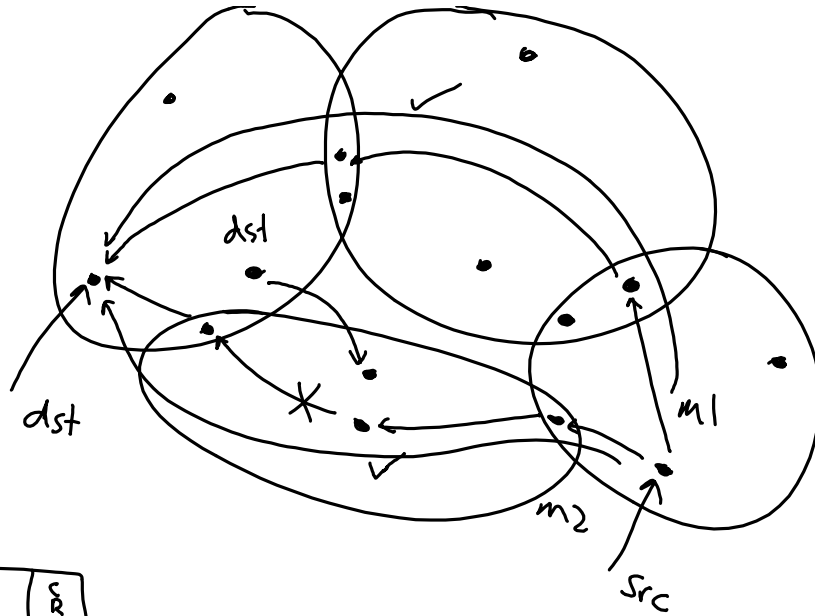
Fragmentation/reassembly

Fragmentation:

break message into pieces

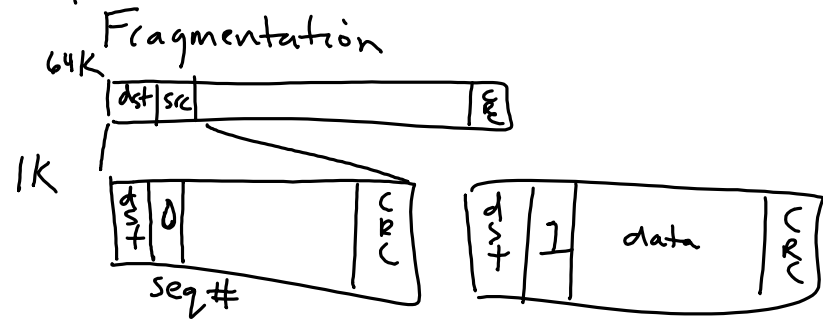
* Packets may be dropped

* Packets may be reordered



CRC - cyclic redundancy check
Checksum -
Transport -

TCP is transport (arbitrary sized)
IP network (1K byte)



TCP - connection-oriented protocol
 IP - connectionless

Sender send fragments with seq #
 recv recv and reassemble

Sender: timeout
 receiver sends acknowledgments
 ACK (seq # i)

1. send seq i

2a. recv recvs seq i; sends ACK(i)

Sender: Sender recvs ACK(i)
 if no ACK for a long time
 start over

2b. if ACK is lost
 recv ^{may} will get duplicate
 go ahead and send ACK(i)

Proofs: Synchronous vs. asynchronous
 bounded latency unbounded

Asynch: \diamond sender recvs ACK(i)
 eventually

Liveness: assumption
 if a message is sent ∞ times
 it is ~~eventually~~ received ∞ times

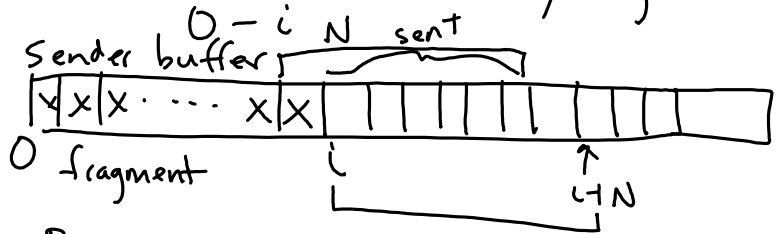
Synch: bounds on transmission time
 FL.

Liveness: ~~if~~ a message can be dropped
 with $P(\text{drop})$
 or at most N times

Sender: sliding window
 window size N

1. sender sends up to N fragments
 messages
2. receiver acknowledges periodically
 if N_2 messages received
 with seq # in order
 $\text{ACK}(10)$

If sender receives $\text{ACK}(i)$
 it can move ahead, forget messages



Receiver
 receives msg i
 msg $i+2$ ← oops

$\text{NAK}(i)$ negative acknowledgment
 Sender get $\text{NAK}(i)$, it resends msg i

$A \wedge B$	Point-to-point
$A \vee B$	Point-to-multipoint
$A \Rightarrow B$	Group comm / multicast protocols
$\forall A. B$	
$\exists A. B$	