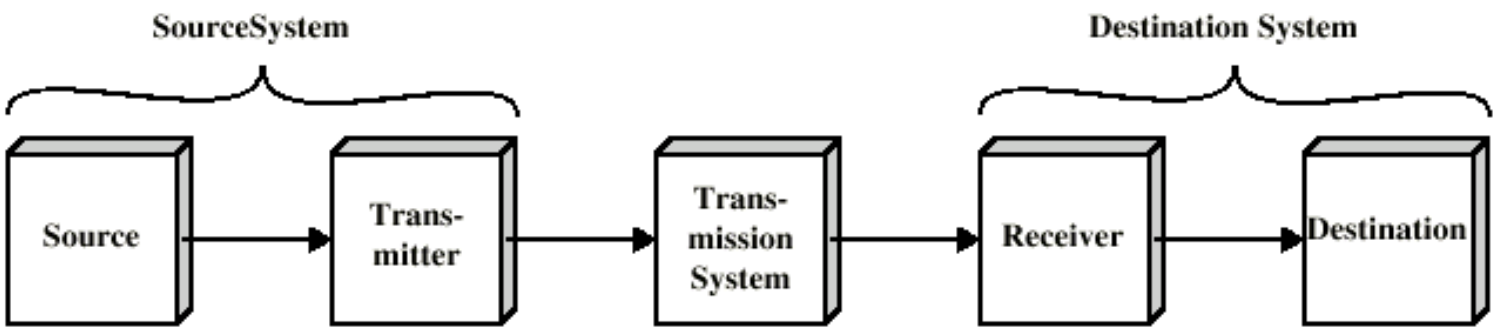
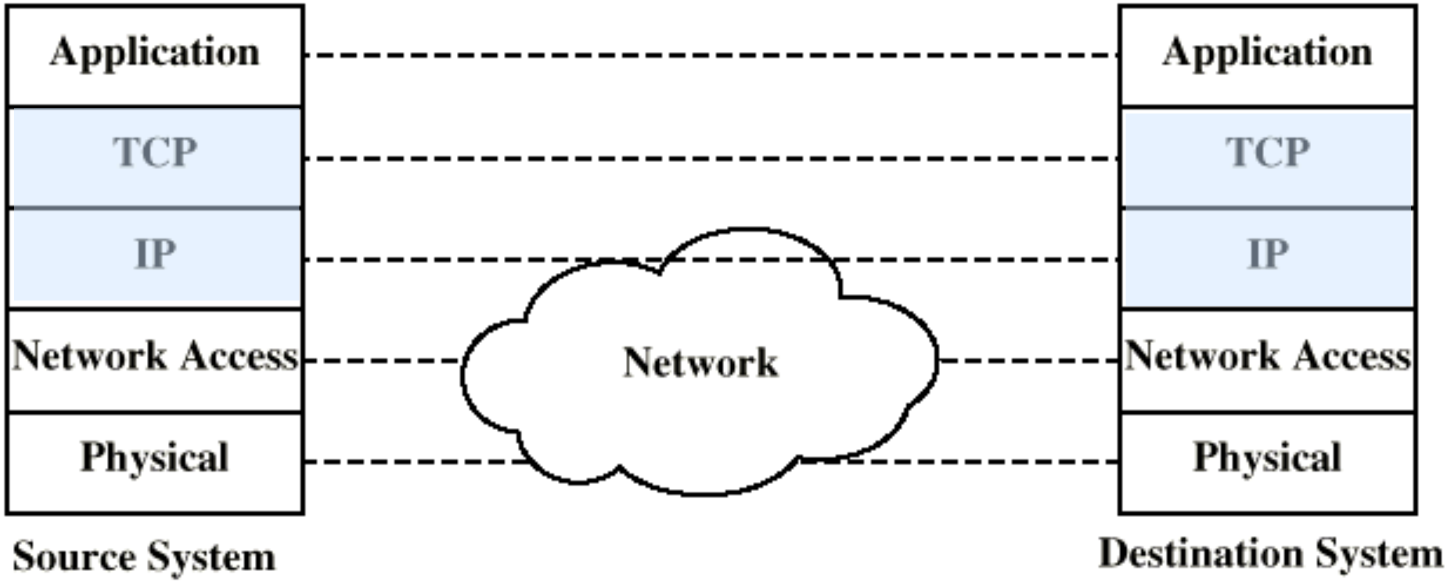
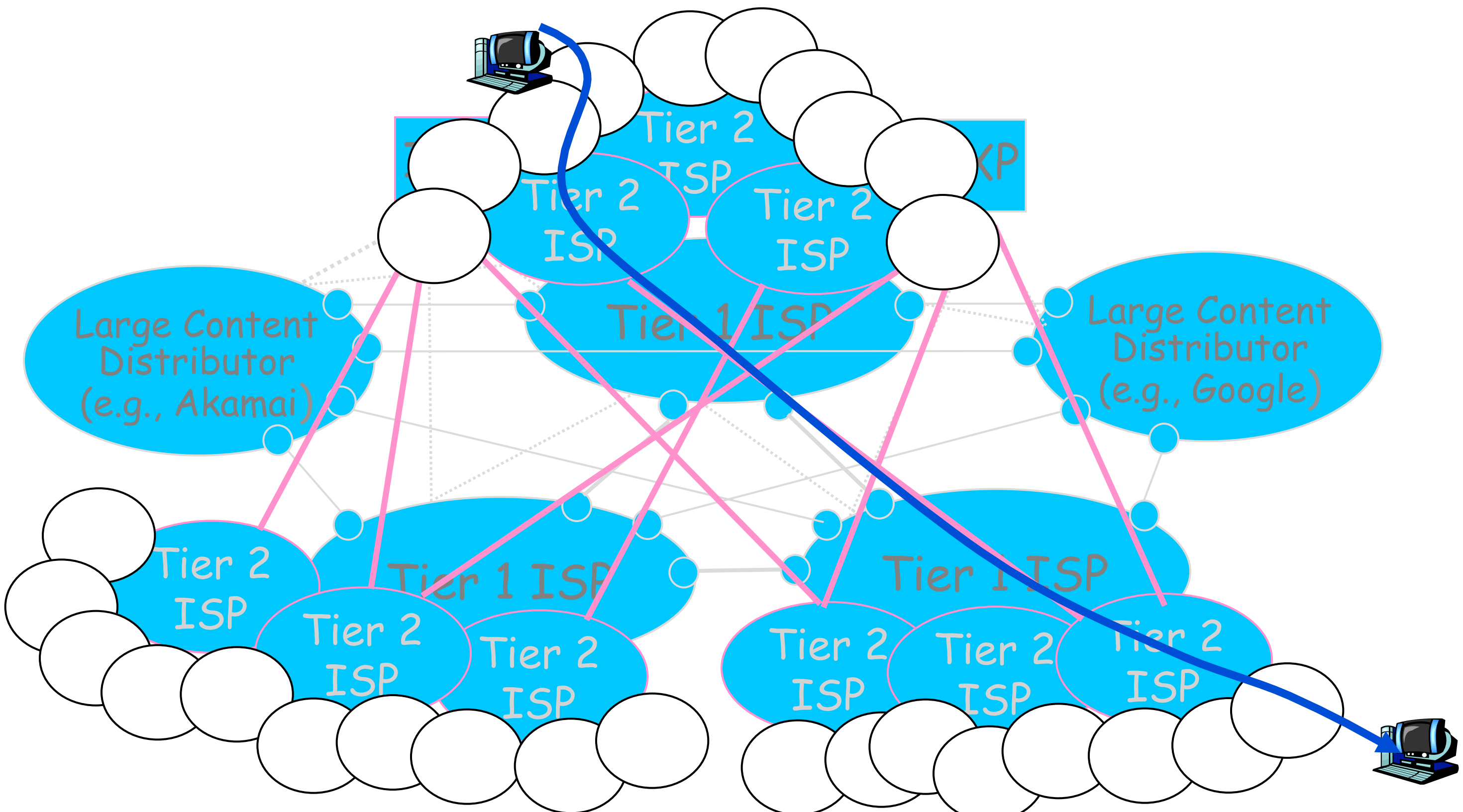
TCP/IP Protocol Architecture Model





Internet Structure: Network of Networks

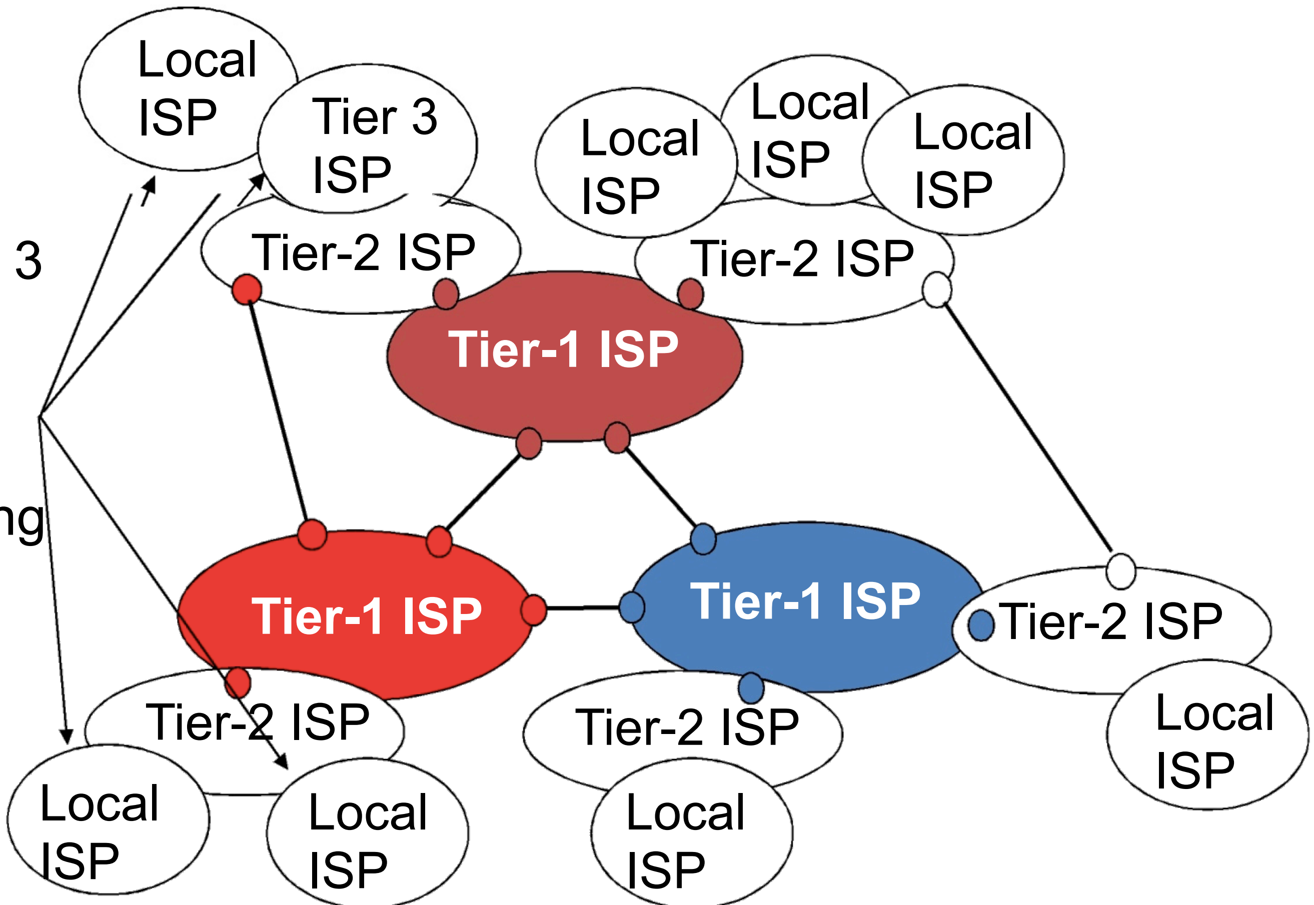
- A packet passes through *many* networks from source host to destination host



Review: Internet Structure

- A Network of Networks

Local and tier- 3 ISPs are *customers* of higher tier ISPs connecting them to rest of Internet



Chapter 1: Roadmap

1.1 What *is* the Internet?

1.2 Network edge

- end systems, access networks, links

1.3 Network core

- circuit switching, packet switching, network structure

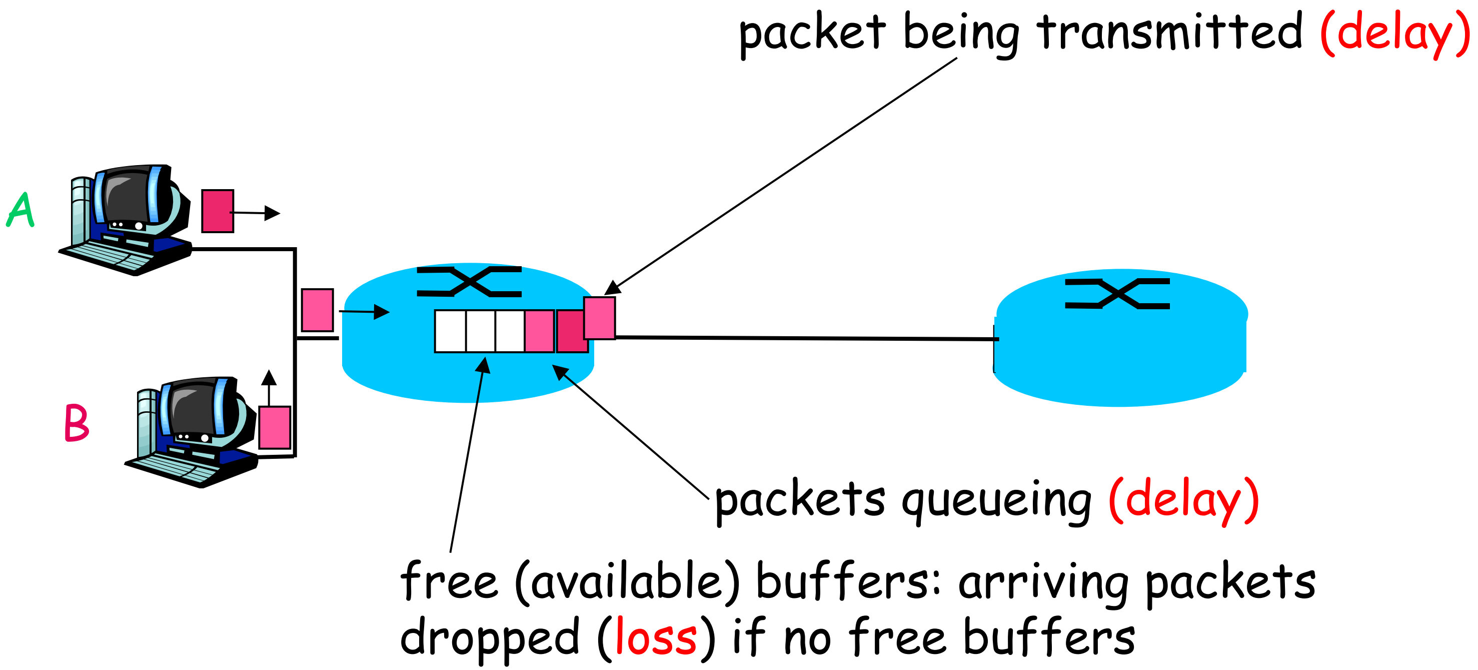
1.4 Delay, loss and throughput in packet-switched networks

1.5 Protocol layers, service models

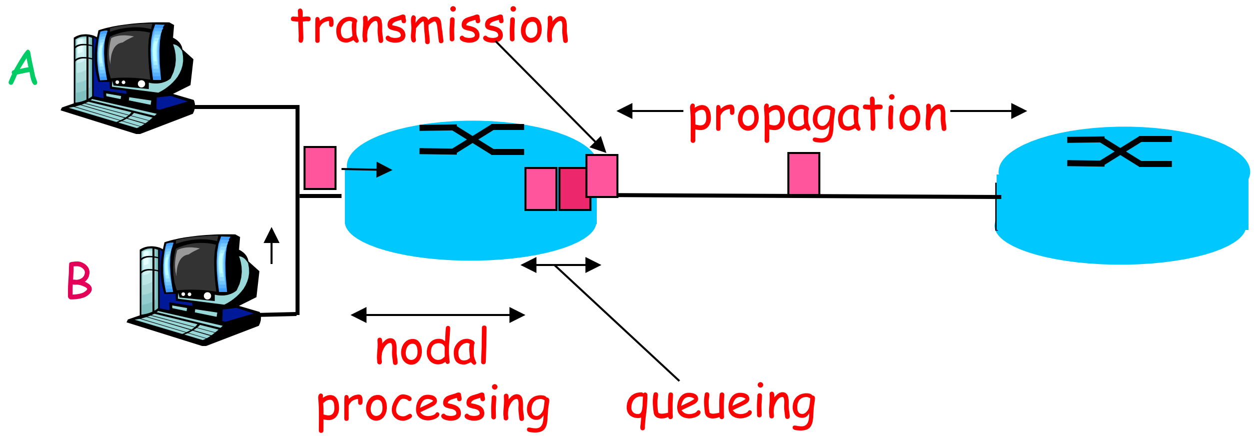
1.6 Networks under attack: security

How do Loss and Delay Occur?

- Packets *queue* in router buffers
  - packet arrival rate to link exceeds output link capacity
  - packets queue, wait for turn



Four Sources of Packet Delay



$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

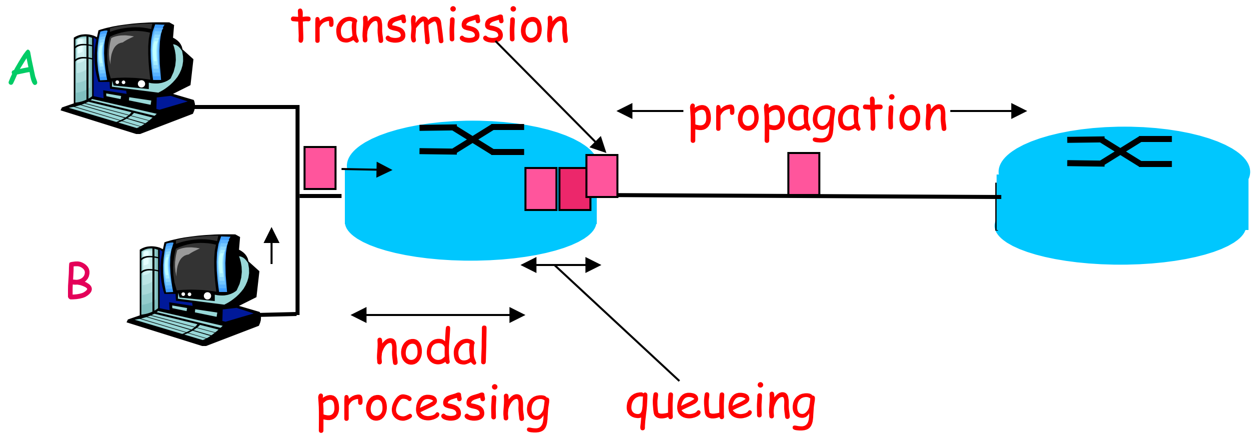
d_{proc}: nodal processing

- check bit errors
- determine output link
- typically < msec

d_{queue}: queueing delay

- time waiting at output link for transmission
- depends on congestion level of router

Four Sources of Packet Delay



$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

d_{trans}: transmission delay:

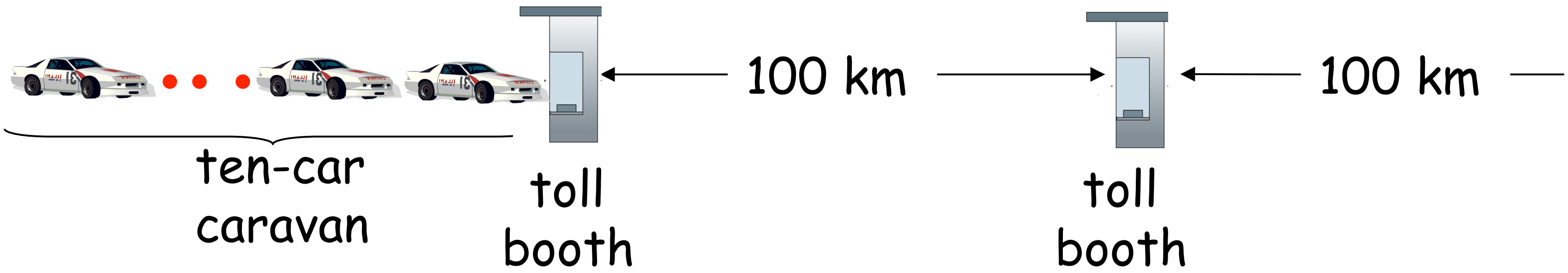
- L: packet length (bits)
- R: link bandwidth (bps)
- $d_{trans} = L/R$

d_{prop}: propagation delay:

- d: length of physical link
- s: propagation speed in medium (~2×10^8 m/sec)
- $d_{prop} = d/s$

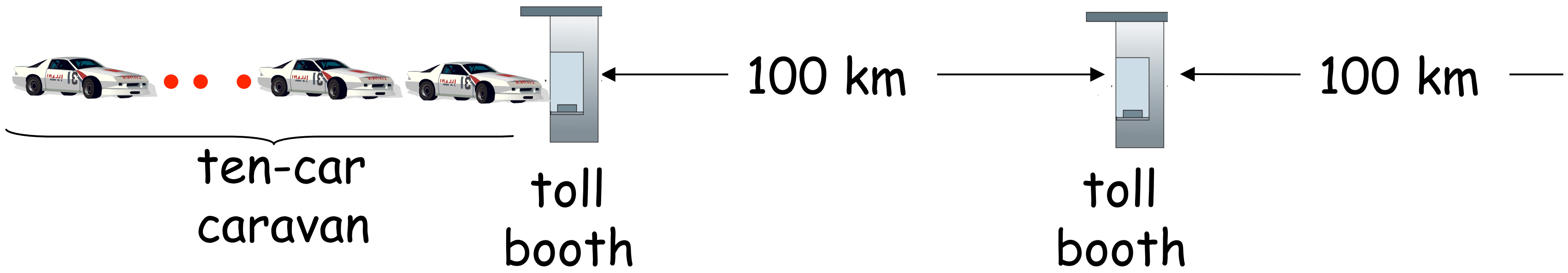
d_{trans} and d_{prop} *very* different

Caravan Analogy



- Cars "propagate" at 100 km/hr
- Toll booth takes 12 sec to service car (transmission time)
- Car~bit; caravan ~ packet
- **Question**: How long until caravan is lined up before 2nd toll booth?

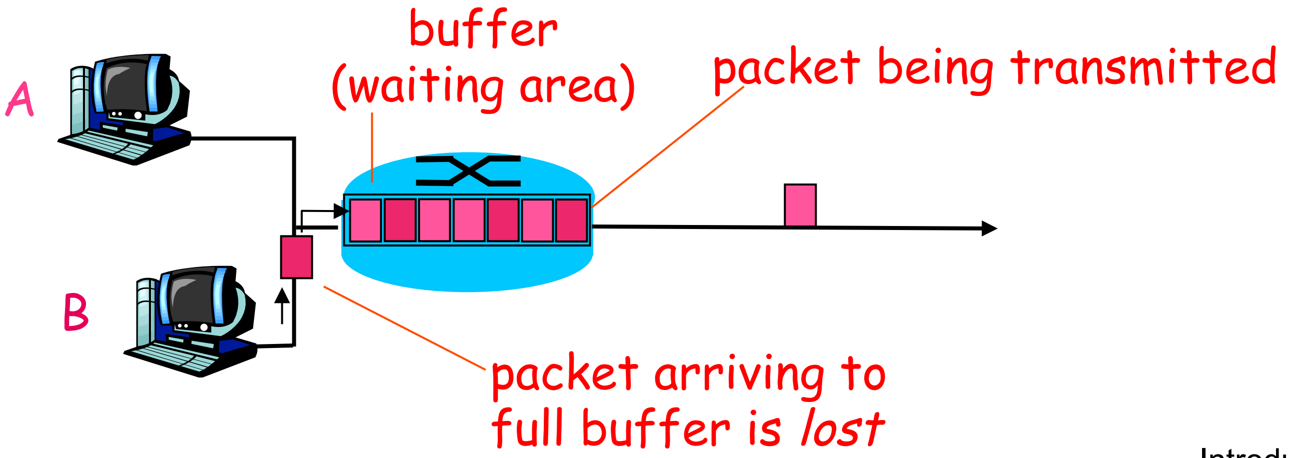
Caravan Analogy (More)



- Cars now "propagate" at 1000 km/hr
- Toll booth now takes 1 min to service a car
- ***Question**:* Will cars arrive to 2nd booth before all cars serviced at 1st booth?

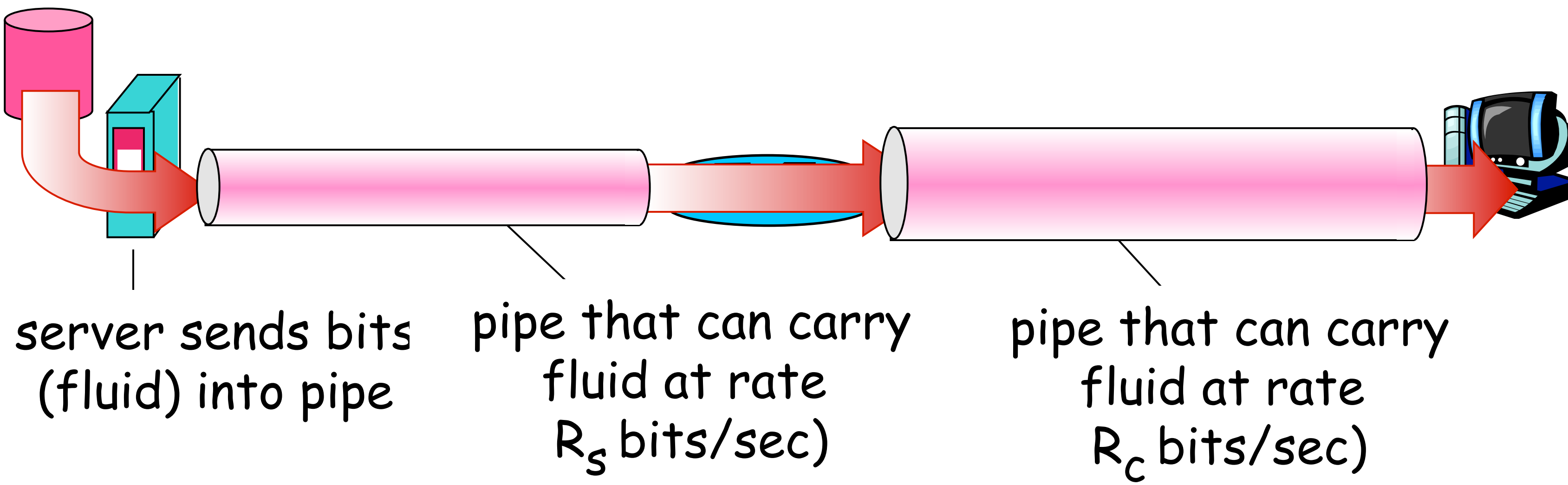
Packet Loss

- Queue (aka buffer) preceding link in buffer has finite capacity
- Packet arriving to full queue dropped (aka lost)
- Lost packet may be retransmitted by previous node, by source end system, or not at all



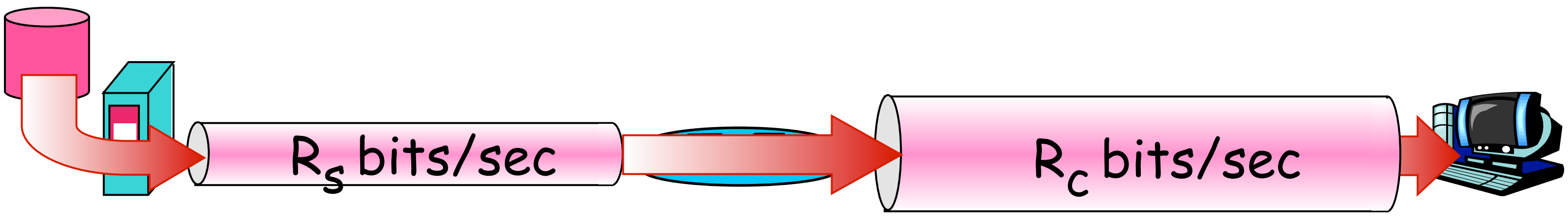
Throughput

- *Throughput:* rate (bits/time unit) at which bits transferred between sender/receiver
  - *instantaneous:* rate at given point in time
  - *average:* rate over longer period of time

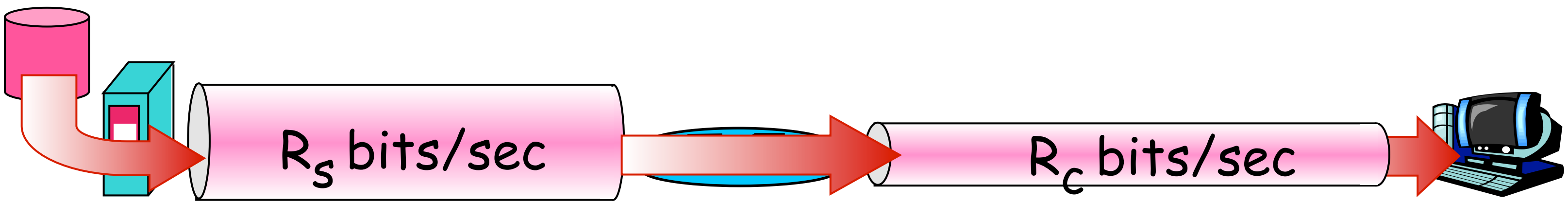


Throughput (more)

- $R_s < R_c$ What is average end-end throughput?



- $R_s > R_c$ What is average end-end throughput?



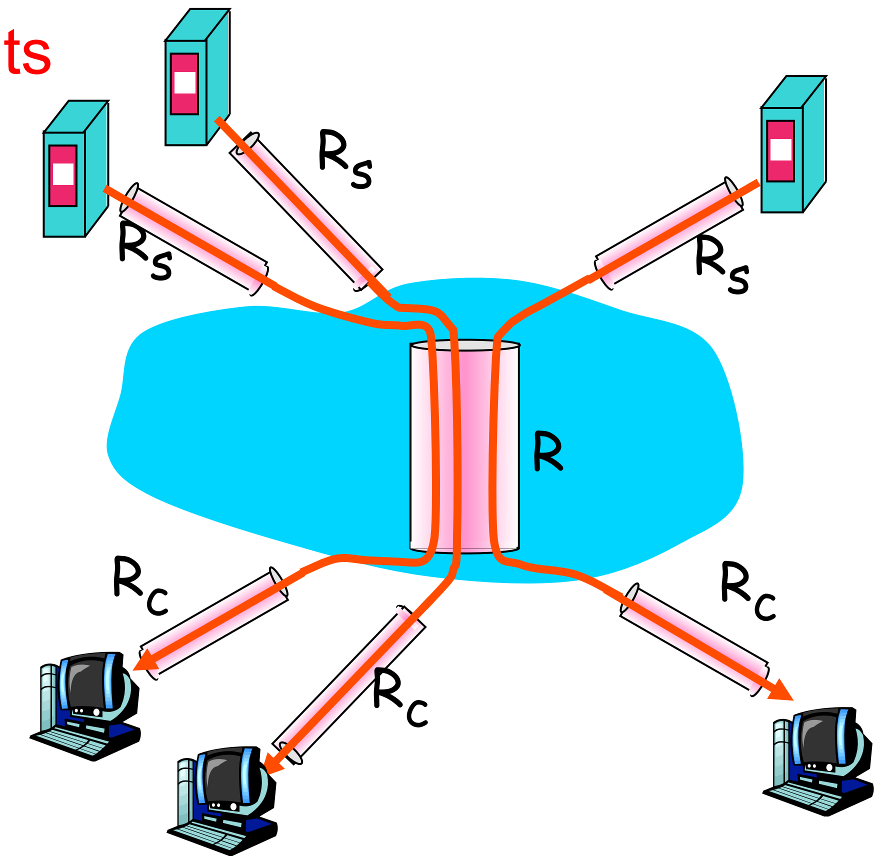
bottleneck link

Link on end-end path that constrains end-end throughput

Throughput: Internet Scenario

- Throughput: rate at which bits transferred between sender/ receiver

- **Question**: 10 connections (fairly) share backbone bottleneck ("goulot d'étranglement") link R bits/ sec.

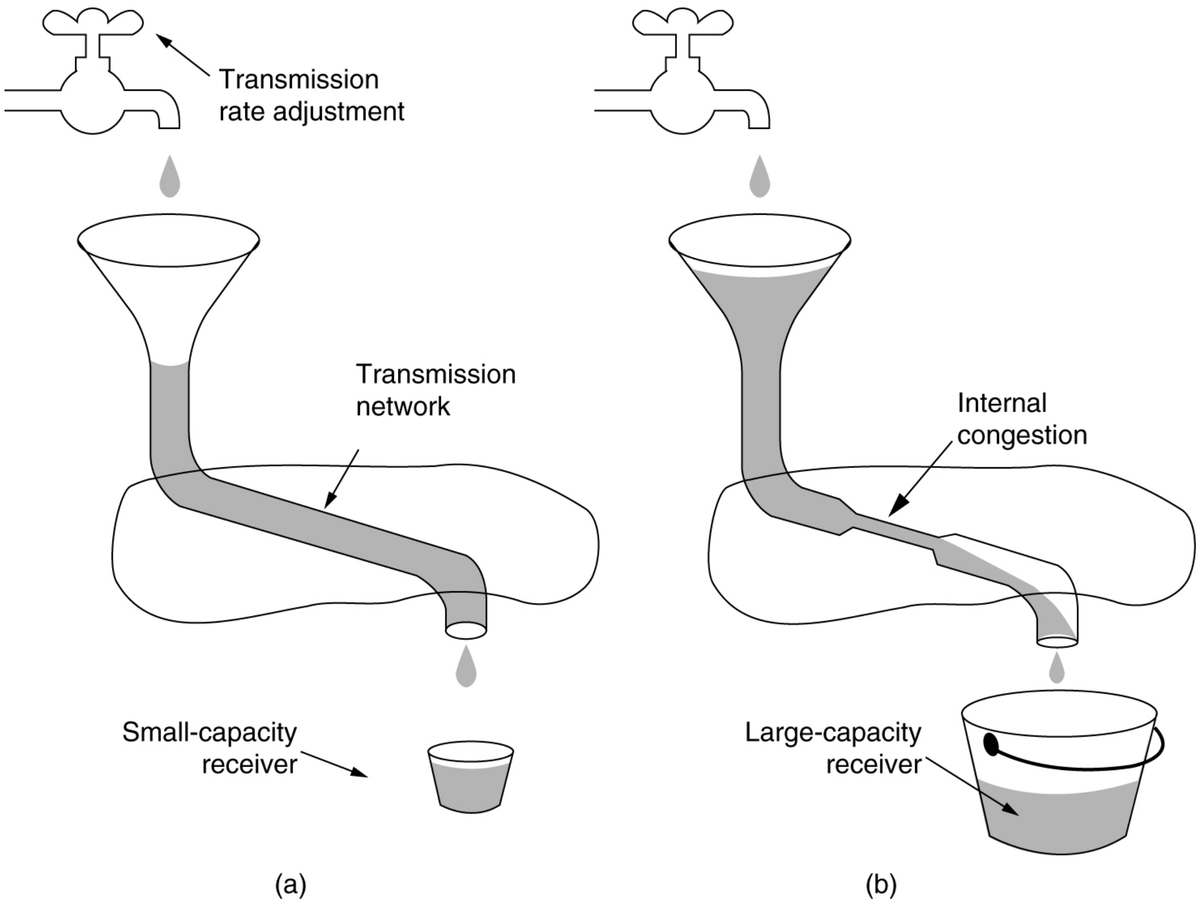


10 connections (fairly) share backbone bottleneck link R bits/sec

What is the per-connection end-end throughput?

- min (R_c, R_s, R/10)
- in practice: R_c or R_s is often bottleneck

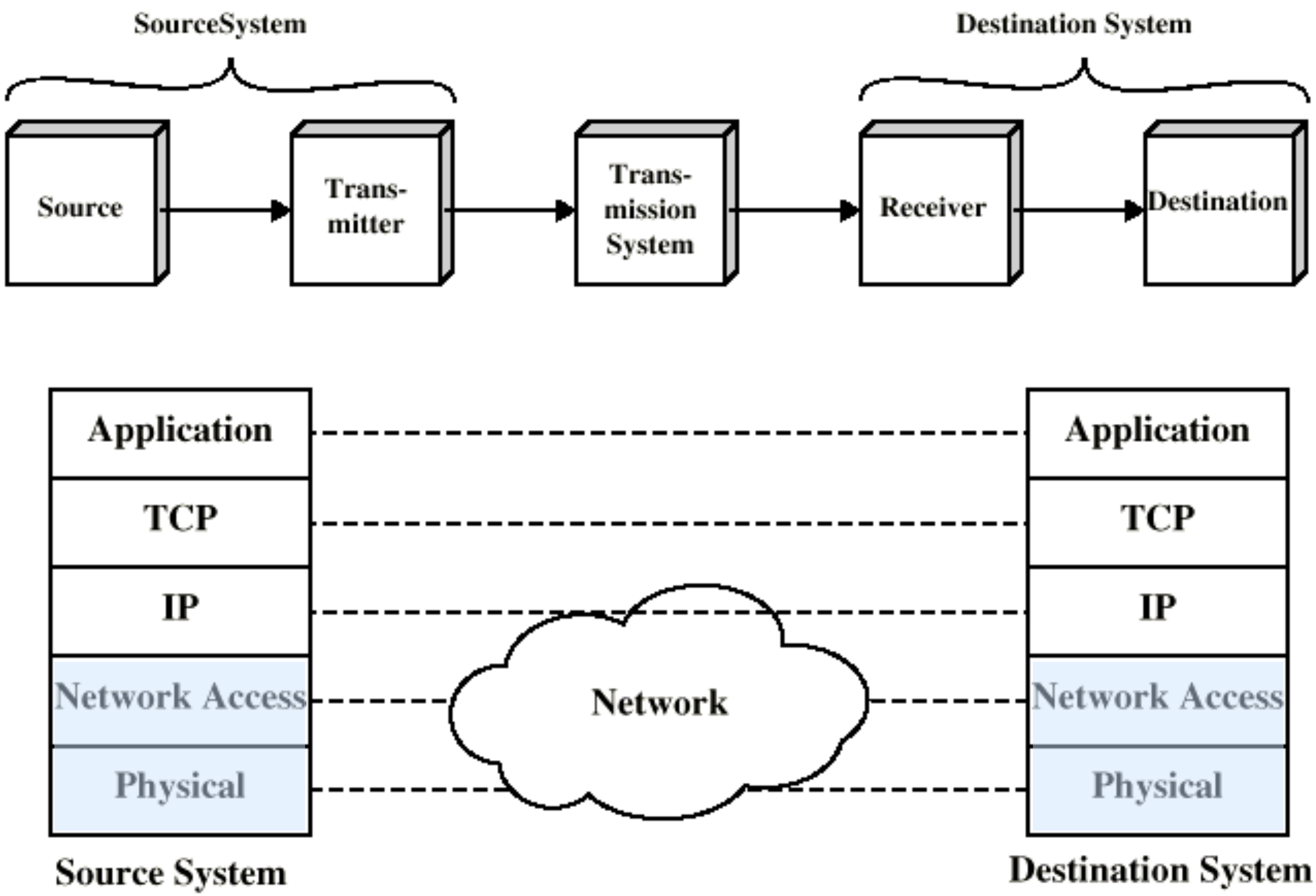
TCP Congestion Control



(a) A fast network feeding a low capacity receiver.

(b) A slow network feeding a high-capacity receiver.

TCP/IP Protocol Architecture Model



Medium Access Control Sublayer

Our goals understand principles behind :

- Channel Allocation Problem
- Multiple Access Protocols
- Ethernet
- Wireless LANs
- Broadband Wireless
- Bluetooth
- RFID
- Data Link Layer Switching

Revised: August 2011

Link Layer: Introduction

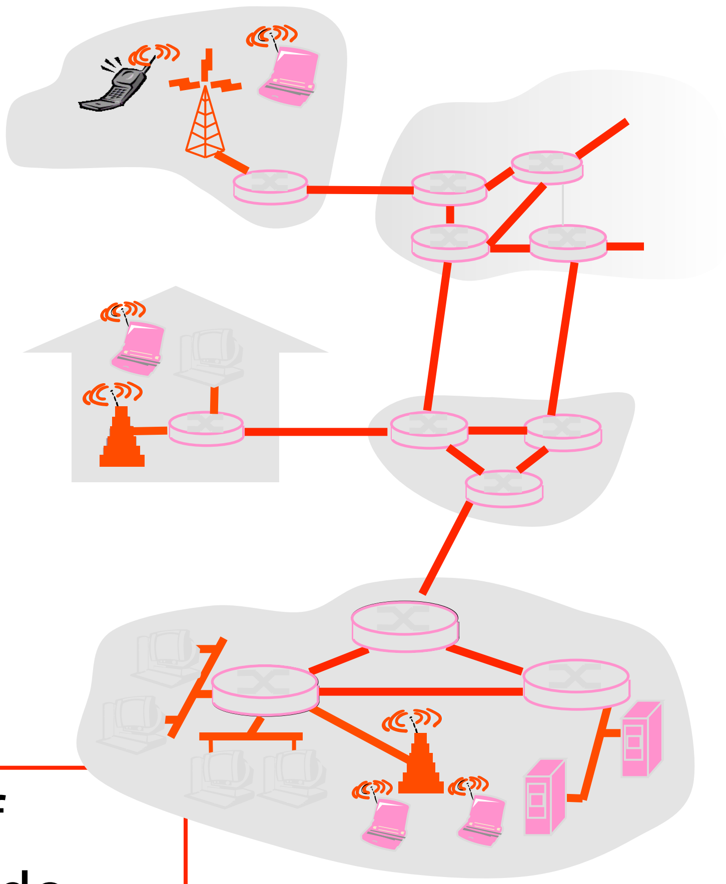
<u>Terminology:</u>

hosts and routers are nodes

communication channels that connect adjacent nodes along communication path are links

- wired links
- wireless links
- LANs

layer-2 packet is a frame, encapsulates datagram



data-link layer has responsibility of transferring datagram from one node to physically adjacent node over a link

Link layer: context

datagram transferred by different link protocols over different links:

- e.g., Ethernet on first link, frame relay on intermediate links, 802.11 on last link

each link protocol provides different services

- e.g., may or may not provide rdt over link

<u>transportation analogy</u>

trip from Princeton to Lausanne

- limo: Princeton to JFK
- plane: JFK to Geneva
- train: Geneva to Lausanne

tourist = datagram

transport segment = communication link

transportation mode = link layer protocol

travel agent = routing algorithm

Link Layer Services

framing, link access:

- encapsulate datagram into frame, adding header, trailer
- channel access if shared medium
- “MAC” addresses used in frame headers to identify source, dest
  - different from IP address!

reliable delivery between adjacent nodes

- we learned how to do this already
- seldom used on low bit-error link (fiber, some twisted pair)
- wireless links: high error rates

Link Layer Services (more)

flow control:

- pacing between adjacent sending and receiving nodes

error detection:

- errors caused by signal attenuation, noise.
- receiver detects presence of errors:
  - signals sender for retransmission or drops frame

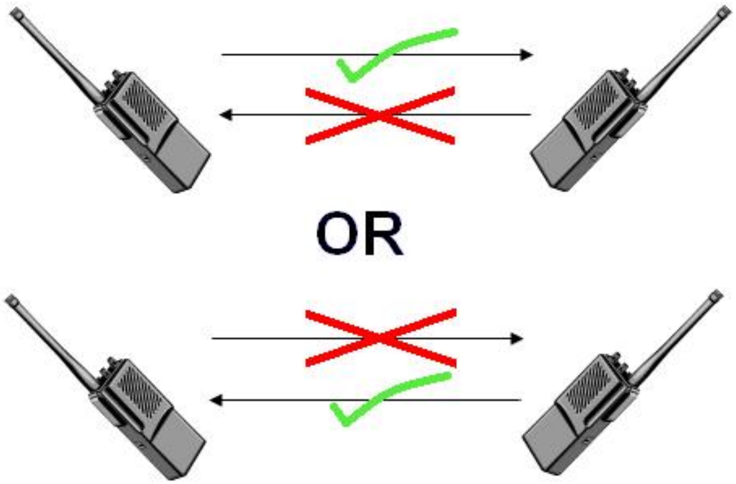
error correction:

- receiver identifies *and corrects* bit error(s) without resorting to retransmission

half-duplex and full-duplex

- with half duplex, nodes at both ends of link can transmit, but not at same time

Half-duplex and Full-duplex



Half-Duplex

Full-Duplex

Where is the link layer implemented?

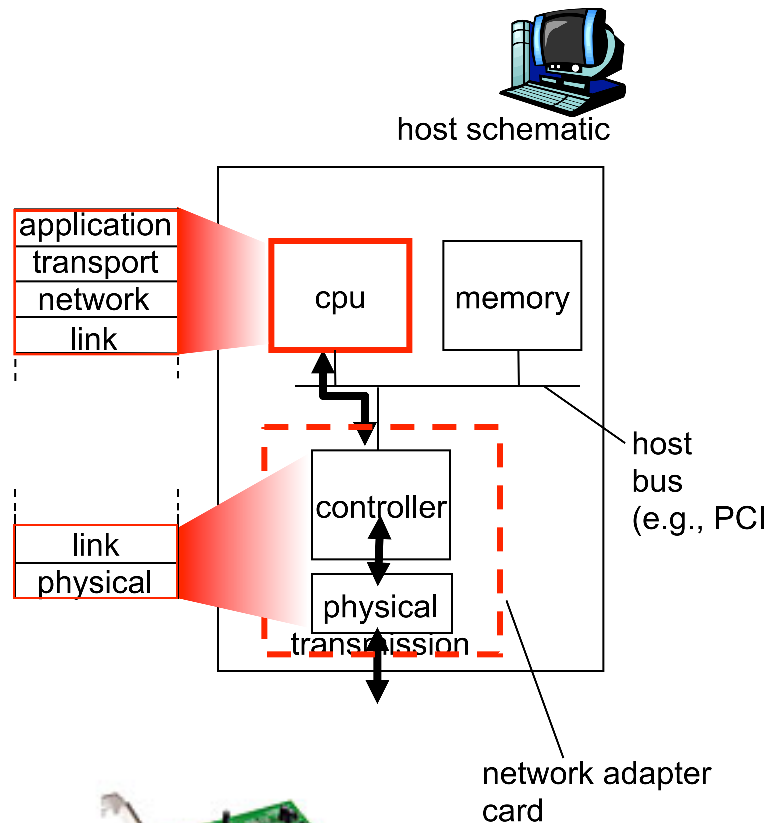
in each and every host

link layer implemented in "adaptor" (aka *network interface card* NIC)

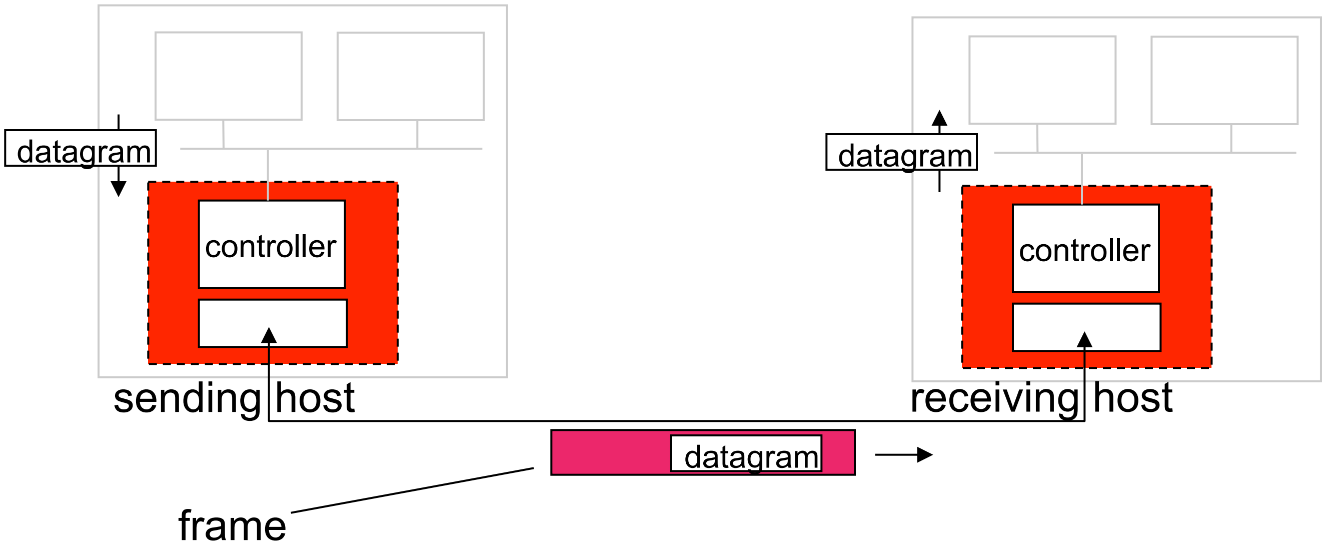
- Ethernet card, PCMCI card, 802.11 card
- implements link, physical layer

attaches into host's system buses

combination of hardware, software, firmware



Adaptors Communicating



sending side:

- encapsulates datagram in frame
- adds error checking bits, flow control, etc.

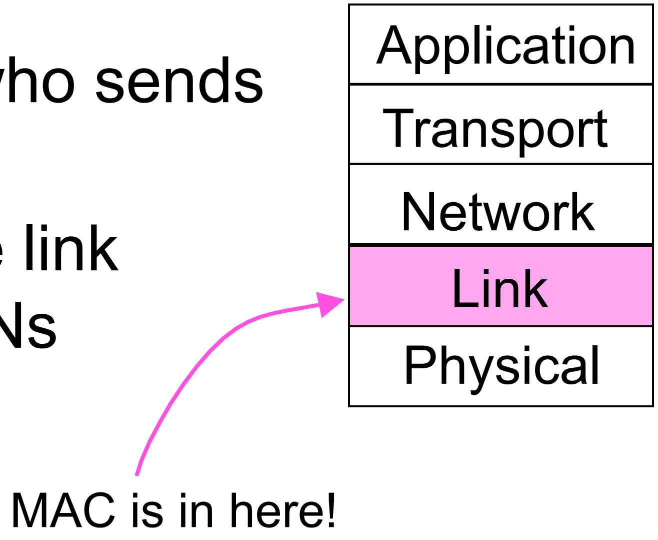
receiving side

- looks for errors, flow control, etc
- extracts datagram, passes to upper layer at receiving side

The MAC Sublayer

Responsible for deciding who sends next on a multi-access link

- An important part of the link layer, especially for LANs



Channel Allocation Problem (1)

For fixed channel and traffic from N users

- Divide up bandwidth using TDM, CDMA, etc.
- This is a static allocation, e.g., FM radio

This static allocation performs poorly for bursty traffic

- Allocation to a user will sometimes go unused

- TDM = Time Division Multiplexing
- CDMA = Code Division Multiple Access

Channel Allocation Problem (3)

Dynamic allocation gives the channel to a user when they need it. Potentially N times as efficient for N users.

Schemes vary with assumptions:

Station Model: The model consists of ***N*** independent stations (e.g., computers, telephones or personal communicators) each with a program or user that generates frames for transmission. Once a frame has been generated, the station is blocked and does nothing until the frame has been successfully transmitted.

Single Channel Assumption: A single channel is available for all communication. All station can transmit on it and all can receive from it;

Collision assumption: If two frames are transmitted simultaneously, they overlap in time and the resulting signal is garbled. This event is called a collision;

Channel Allocation Problem (4)

Dynamic allocation gives the channel to a user when they need it. Potentially N times as efficient for N users.

Schemes vary with assumptions:

Continuous time: Frame transmission can begin at any instant. There is no master clock dividing time into discrete intervals;

Slotted time: Time is divided into discrete intervals (slots). Frame transmissions always begin at the start of a slot. A slot may contain 0, 1 or more frames, corresponding to an idle slot, a successful transmission, or a collision, respectively;

Carrier Sense: Stations can tell if the channel is in use before trying to use it. If the channel is sensed as busy, no station will attempt to use it until goes idle;

No carrier sense: Stations cannot sense the channel before trying to use it. They just go ahead and transmit. Only later they determine whether the transmission was successful.

Channel Allocation Problem (2)

Dynamic allocation gives the channel to a user when they need it. Potentially N times as efficient for N users.

Schemes vary with assumptions:

Assumption	**Implication**
N Independent traffic (station)	Often not a good model, but permits analysis
Single channel	No external way to coordinate senders
Observable collisions	Needed for reliability; mechanisms vary
Continuous or slotted time	Slotting may improve performance
Carrier sense	Can improve performance if available

Example: Congestion in M2M over LTE

The expected number of M2M / MTC devices until 2020 is approximately 20 billions.

This devices are going to be applied in a wide range of applications.

To make this “dream become true” they will need an access network for exchange data/information.

In this context the cellular networks represents a good alternative of access network.

But there is a little problem... Cellular networks were projected for humans, not for machines!

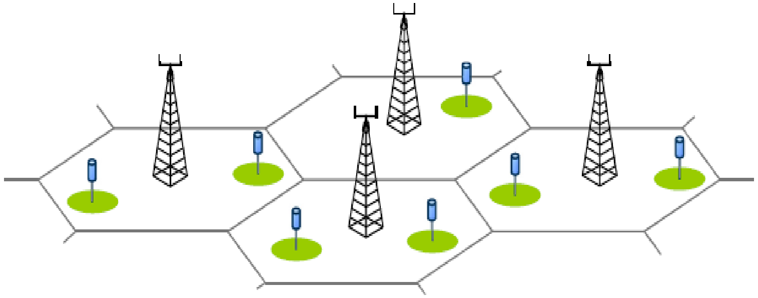
Example M2M: Cellular Networks

Used by mobile networks operators.

Designed for H2H and H2M types of communication.

Key features:

- Ubiquity
- Accessibility
- Security
- Designed for H2H communication



Technology:

GSM, UTMS, CDMA, LTE

Machine-to-Machine Communication

Machine-to-Machine (M2M) communications is a technology that enables one or more autonomous machines to communicate directly with one another without human intervention.

Its main characteristics are:

- Large number of simultaneously connected devices
- Small data volume transmissions
- Vastly diverse quality-of-service (QoS) requirements

Play important role on the Internet of Things (IoT)!

Example: Congestion in M2M over LTE

The congestion of MTC network usually happens in **radio Network** and **core network** because of mass concurrent signaling and data transmissions.

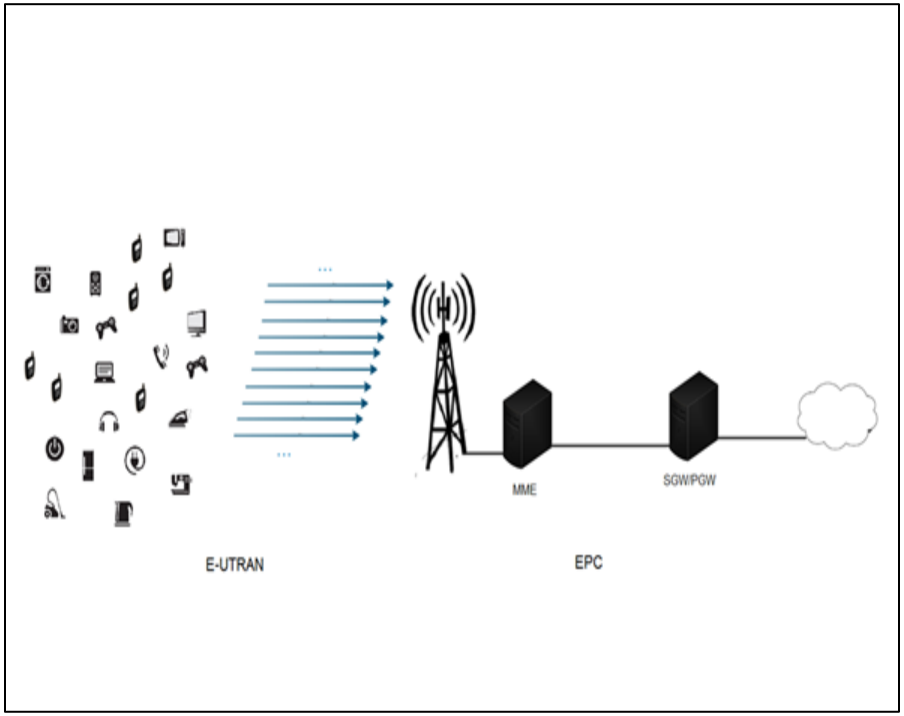
The Occurrence of congestion in LTE network:

- **Radio Access Network (RAN):** large number of devices requesting access to the network to enable \ modify \ disable a connection.

- **Core Network (CN):** excessive signaling flows or data (from several eNB) directed to the same element of the EPC (envolved Packet Core), for example, the S-GW and the MME (Mobility Management Entity).

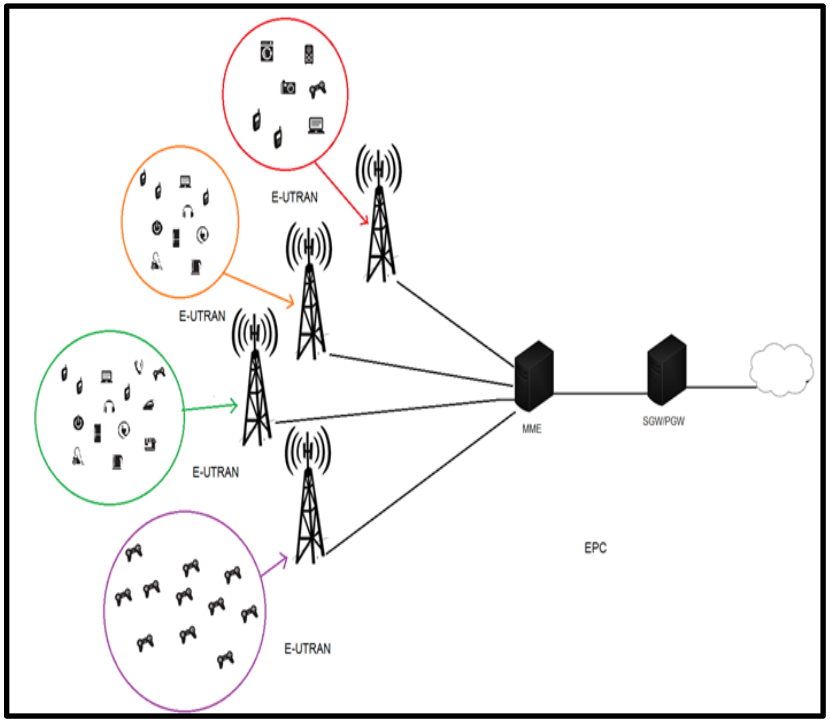
Example: Congestion in M2M over LTE

Radio Access Network (RAN)



RAN - Congestion

Core Network (CN)



EPC - Congestion

LTE

Home Subscriber Server

Figure 2: Basic EPS architecture with E-UTRAN access

Channel Allocation Problem

Multiple Access Links and Protocols

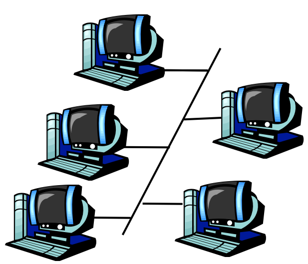
Two types of "links":

point-to-point

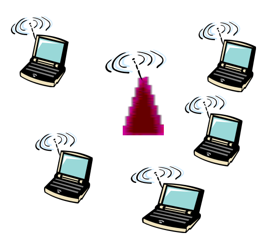
- PPP for dial-up access
- point-to-point link between Ethernet switch and host

broadcast (shared wire or medium)

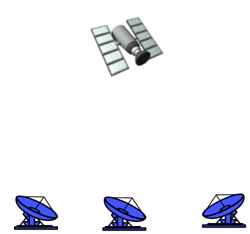
- old-fashioned Ethernet
- upstream HFC (Hybrid Fiber-Coaxial)
- 802.11 wireless LAN



shared wire (e.g., cabled Ethernet)



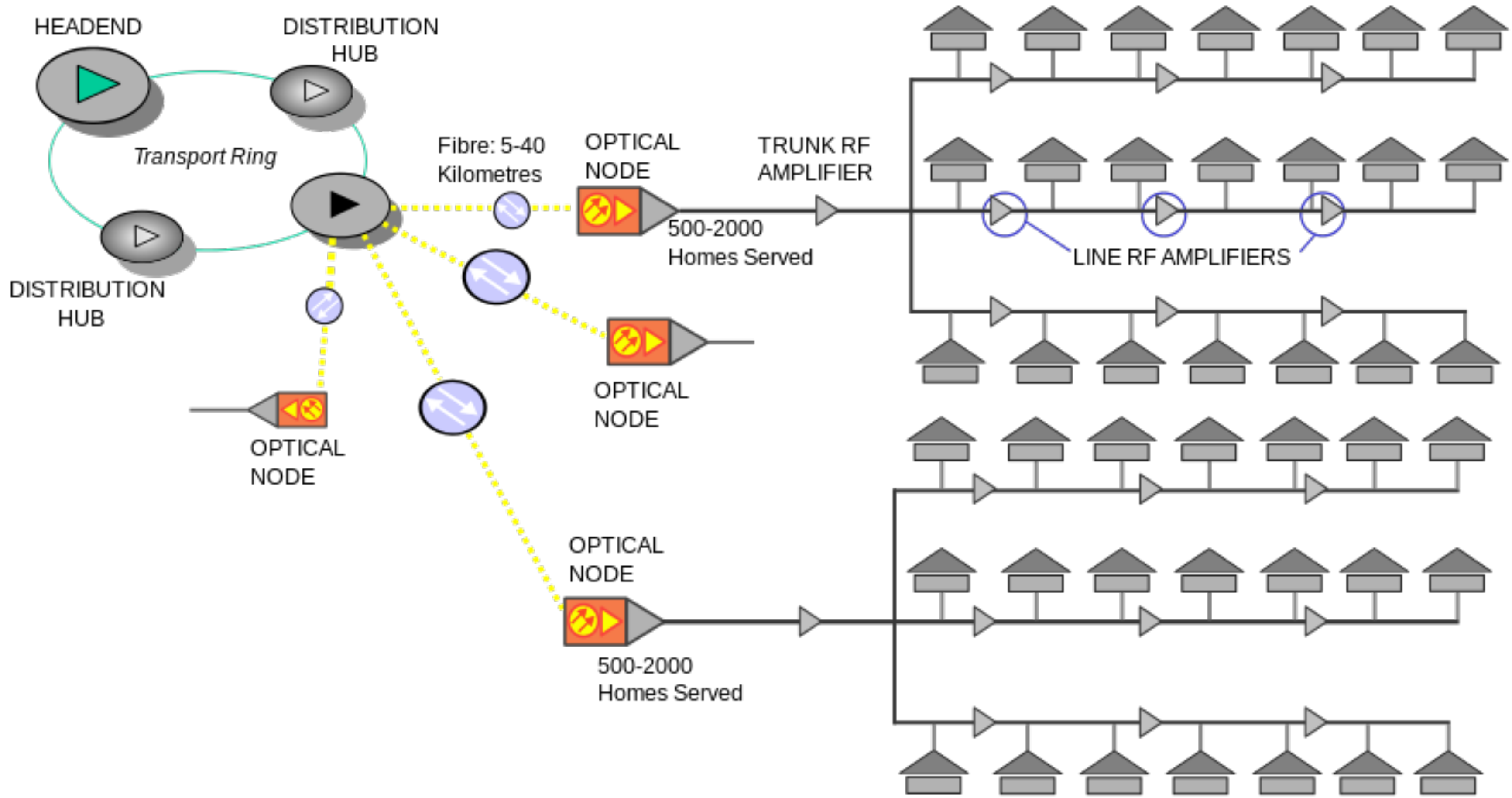
shared RF (Radio frequency) (e.g., 802.11 WiFi)



shared RF (satellite)

humans at a cocktail party (shared air, acoustical)

Hybrid Fiber-Coaxial



Multiple Access protocols

- single shared broadcast channel
- two or more simultaneous transmissions by nodes: interference
  - collision if node receives two or more signals at the same time

multiple access protocol

- distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit
- communication about channel sharing must use channel itself!
  - no out-of-band channel for coordination

Ideal Multiple Access Protocol

Broadcast channel of rate R bps

1. when one node wants to transmit, it can send at rate R.
2. when M nodes want to transmit, each can send at average rate R/M
3. fully decentralized:
   - no special node to coordinate transmissions
   - no synchronization of clocks, slots
4. simple

MAC Protocols: a taxonomy

Three broad classes:

Channel Partitioning

- divide channel into smaller “pieces” (time slots, frequency, code)
- allocate piece to node for exclusive use

Random Access

- channel not divided, allow collisions
- “recover” from collisions

“Taking turns”

- nodes take turns, but nodes with more to send can take longer turns

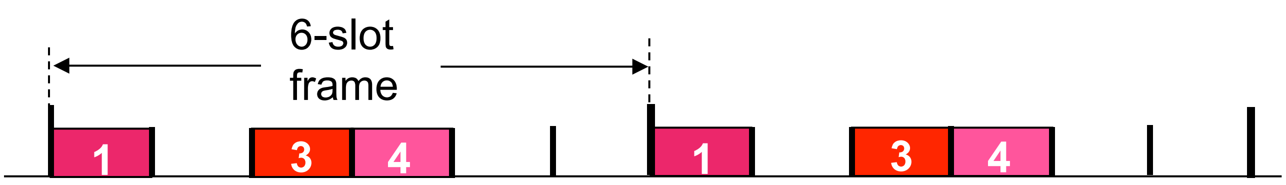
Multiple Access Protocols

- ALOHA »
- CSMA (Carrier Sense Multiple Access) »
- Collision-free protocols »
- Limited-contention protocols »
- Wireless LAN protocols »

Channel Partitioning MAC protocols: TDMA

TDMA: Time Division Multiple Access

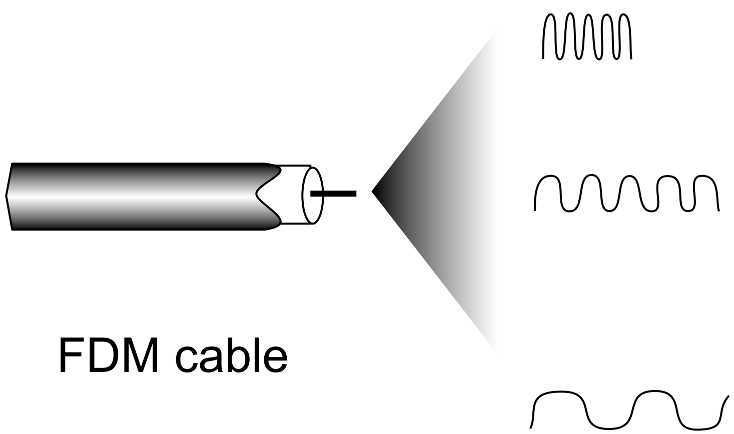
- access to channel in "rounds"
- each station gets fixed length slot (length = pkt trans time) in each round
- unused slots go idle
- example: 6-station LAN, 1,3,4 have pkt, slots 2,5,6 idle



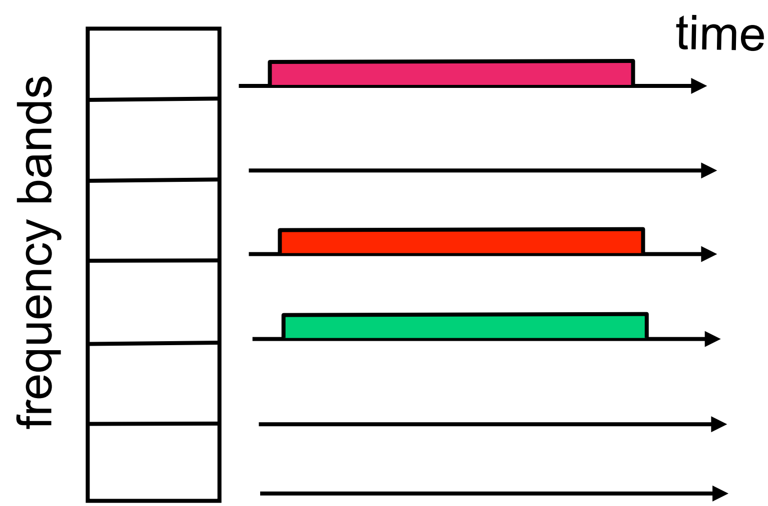
Channel Partitioning MAC protocols: FDMA

FDMA: Frequency Division Multiple Access

- channel spectrum divided into frequency bands
- each station assigned fixed frequency band
- unused transmission time in frequency bands go idle
- example: 6-station LAN, 1,3,4 have pkt, frequency bands 2,5,6 idle



FDM cable



Channel Allocation Problem (1)

For fixed channel and traffic from N users

- Divide up bandwidth using TDM, CDMA, etc.
- This is a static allocation, e.g., FM radio

This static allocation performs poorly for bursty traffic

- Allocation to a user will sometimes go unused

- TDM = Time Division Multiplexing
- CDMA = Code Division Multiple Access

Random Access Protocols

When node has packet to send

- transmit at full channel data rate R.
- no *a priori* coordination among nodes

two or more transmitting nodes ➜ "collision",

random access MAC protocol specifies:

- how to detect collisions
- how to recover from collisions (e.g., via delayed retransmissions)

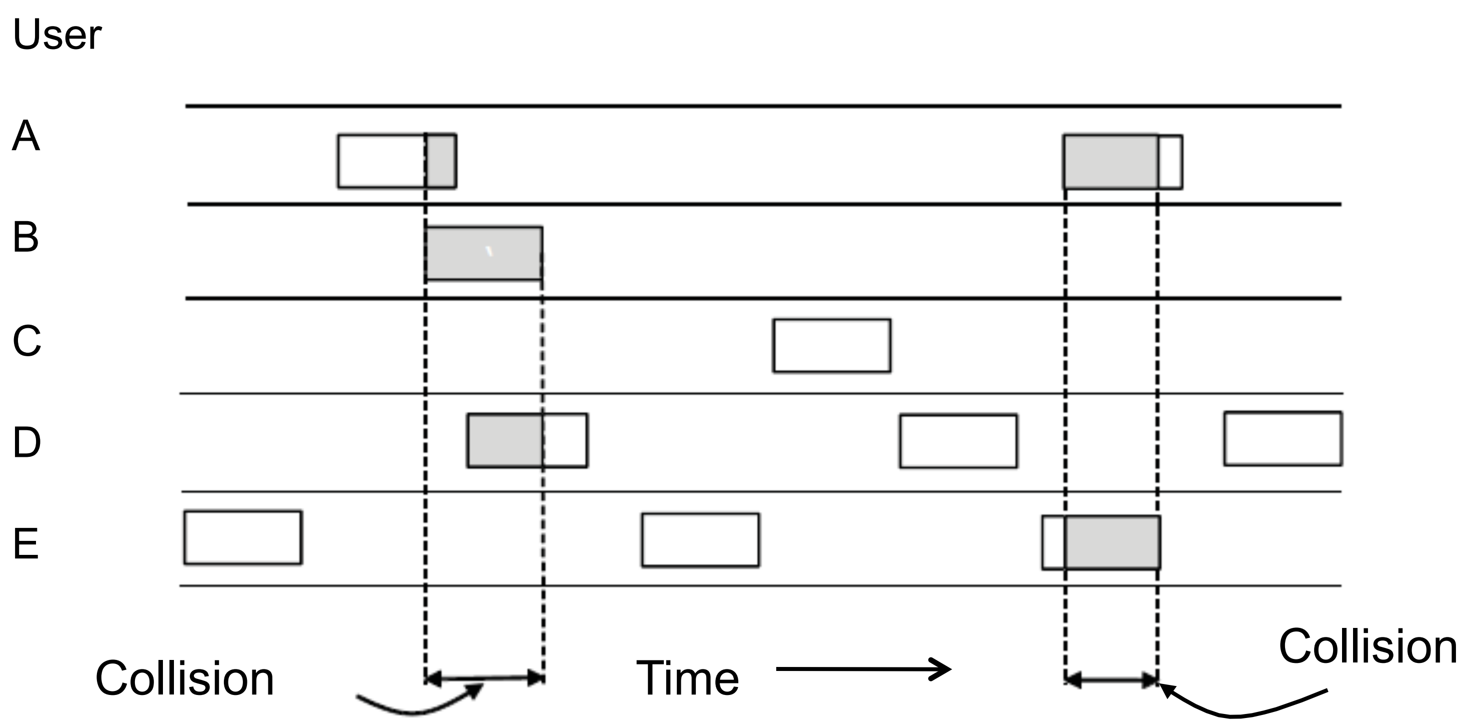
Examples of random access MAC protocols:

- ALOHA
- slotted ALOHA
- CSMA, CSMA/CD, CSMA/CA

ALOHA (1)

In pure ALOHA, users transmit frames whenever they have data; users retry after a random time for collisions

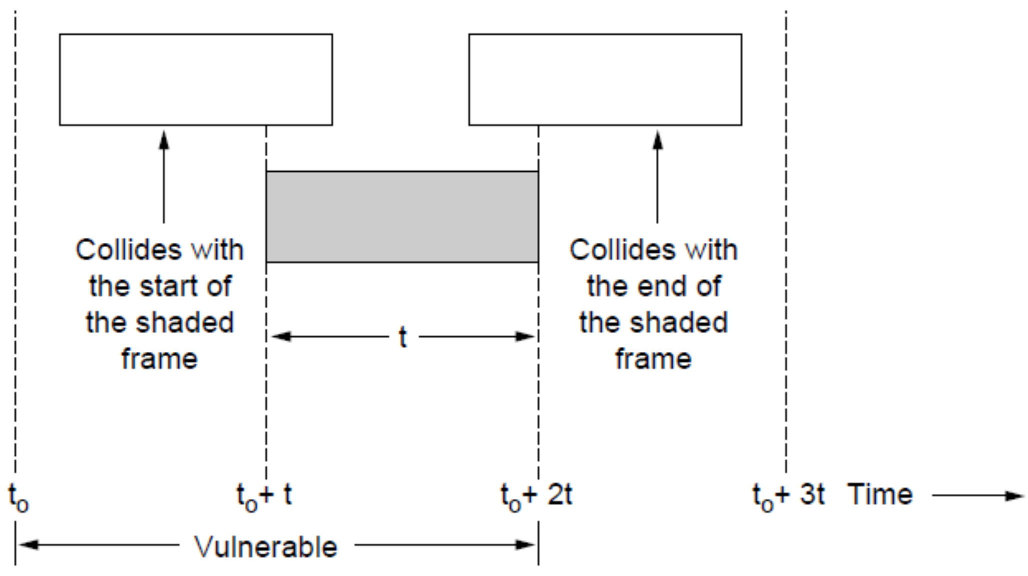
- Efficient and low-delay under low load



ALOHA (2)

Under what conditions will the shaded frame arrive undamaged?

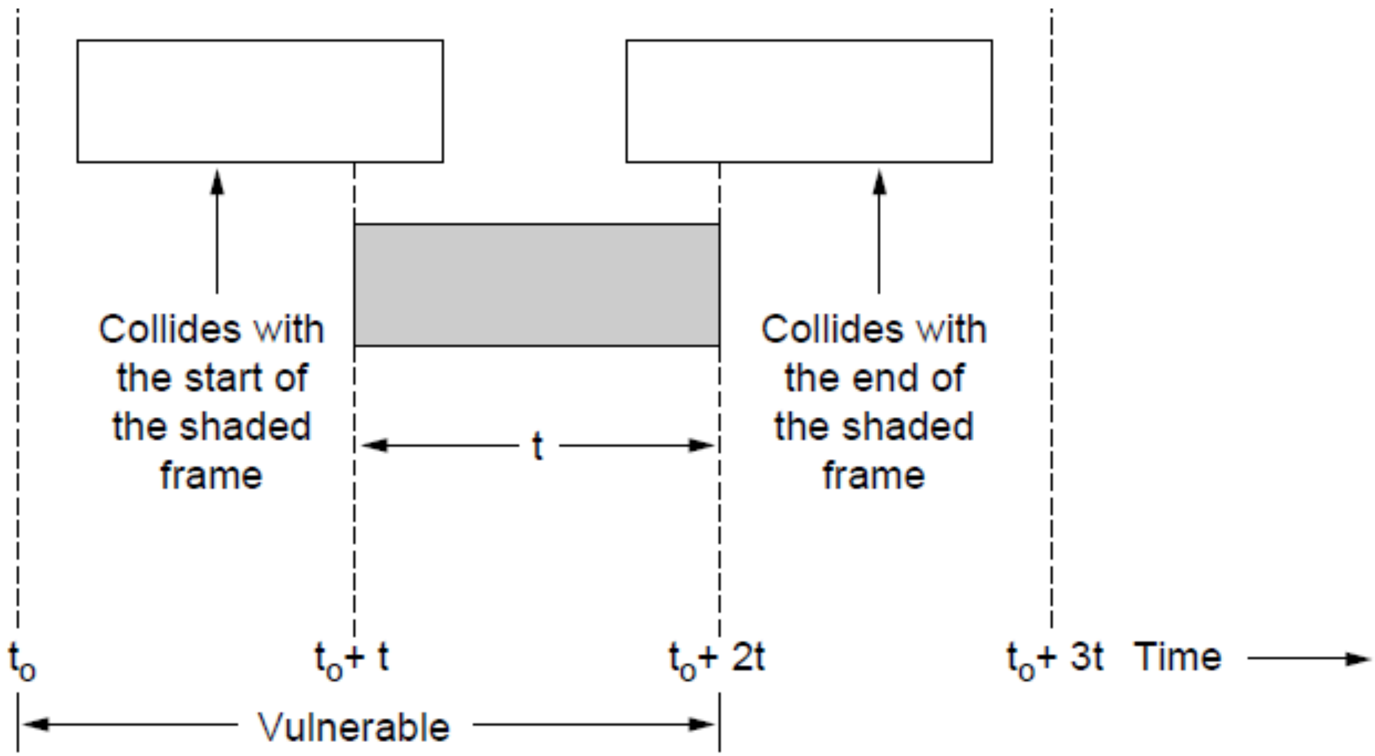
- Let t be the time required to send a frame
- If any other user has generated a frame between time t_0 and $t_0 + t$, the end of that frame will collide with the beginning of the shaded one
- Similarly, any other frame started between $t_0 + t$ and $t_0 + 2t$ will bump into the end of the shaded frame
- Since the pure ALOHA a station does not listen to the channel before transmitting, it has no way of knowing that another frame was already underway



ALOHA (2)

Collisions happen when other users transmit during a vulnerable period that is **twice the frame time**

- Synchronizing senders to slots can reduce collisions



Pure (unslotted) ALOHA

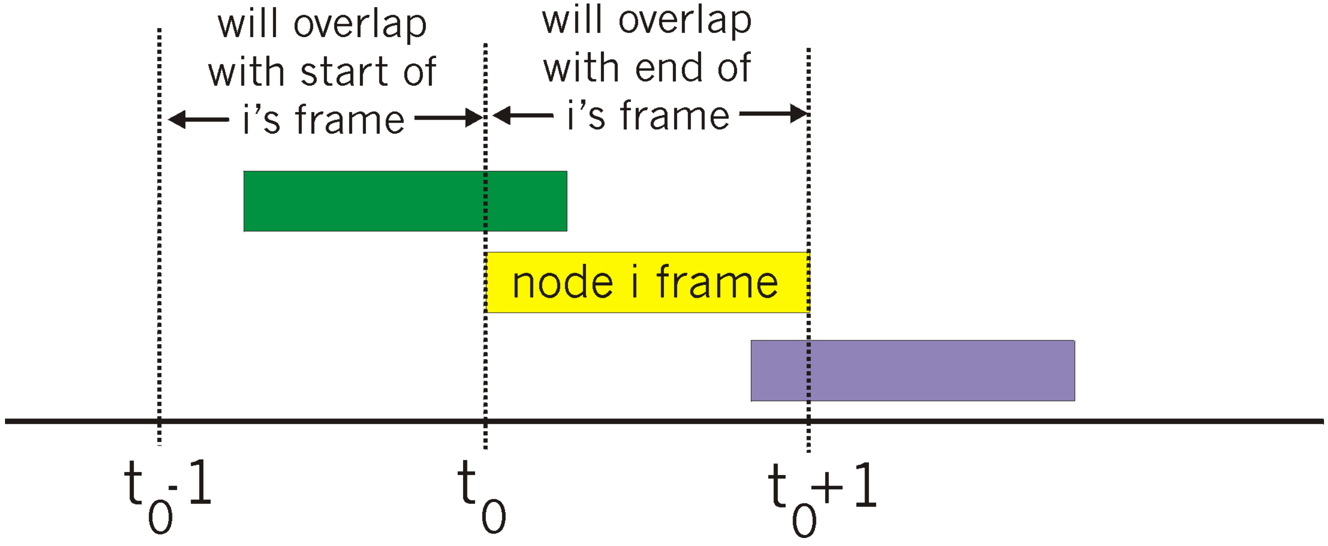
unslotted Aloha: simpler, no synchronization

when frame first arrives

- transmit immediately

collision probability increases:

- frame sent at t_0 collides with other frames sent in $[t_0-1, t_0+1]$



Slotted ALOHA

Assumptions:

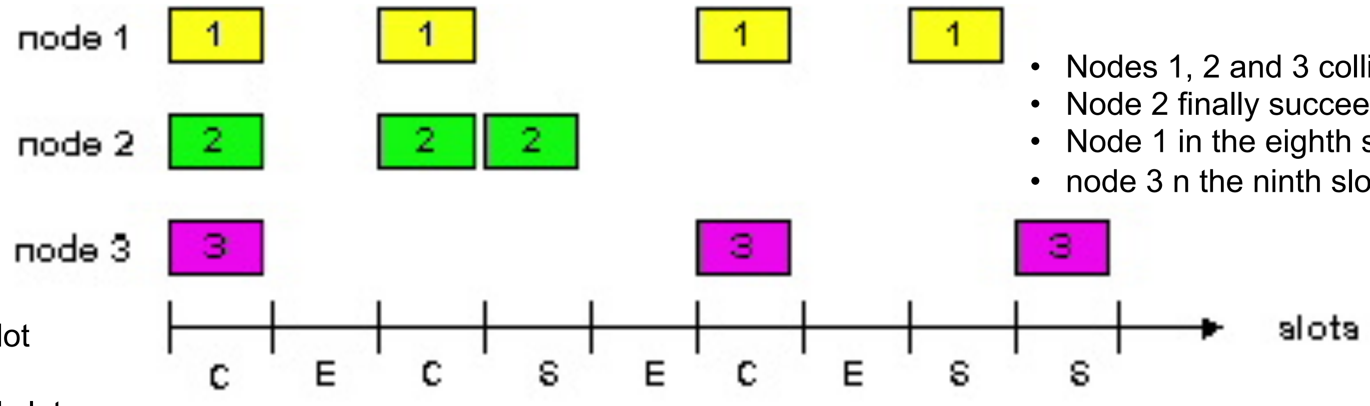
- all frames same size
- time divided into equal size slots (time to transmit 1 frame)
- nodes start to transmit only slot beginning
- nodes are synchronized
- if 2 or more nodes transmit in slot, all nodes detect collision

Operation:

when node obtains fresh frame, transmits in next slot

- *if no collision:* node can send new frame in next slot
- *if collision:* node retransmits frame in each subsequent slot with prob. p until success

Slotted ALOHA



- Nodes 1, 2 and 3 collide in the first slot.
- Node 2 finally succeeds in the fourth slot,
- Node 1 in the eighth slot, and
- node 3 n the ninth slot

Legends:
C = Collision slot
E = Empty slot
S = Successful slot

Pros

1. single active node can continuously transmit at full rate of channel
2. highly decentralized: only slots in nodes need to be in sync
3. simple

Cons

1. collisions, wasting slots
2. idle slots
3. clock synchronization

Slotted Aloha efficiency

Efficiency : long-run fraction of successful slots (many nodes, all with many frames to send)

suppose: N nodes with many frames to send, each transmits in slot with probability *p*

prob that given node has success in a slot = $p(1-p)^{N-1}$

prob that *any* node has a success = $Np(1-p)^{N-1}$

max efficiency: find p* that maximizes $Np(1-p)^{N-1}$

for many nodes, take limit of $Np^*(1-p^*)^{N-1}$ as N goes to infinity, gives:

Max efficiency = 1/e = .37

At best: channel used for useful transmissions 37% of time!

!

Pure Aloha efficiency

P(success by given node) = P(node transmits) ·

P(no other node transmits in $[p_0-1,p_0]$ ·

P(no other node transmits in $[p_0-1,p_0]$

$= p \cdot (1-p)^{N-1} \cdot (1-p)^{N-1}$

$= p \cdot (1-p)^{2(N-1)}$

… choosing optimum p and then letting n -> infty ...

= 1/(2e) = .18

At best: channel used for useful transmissions 18% of time!

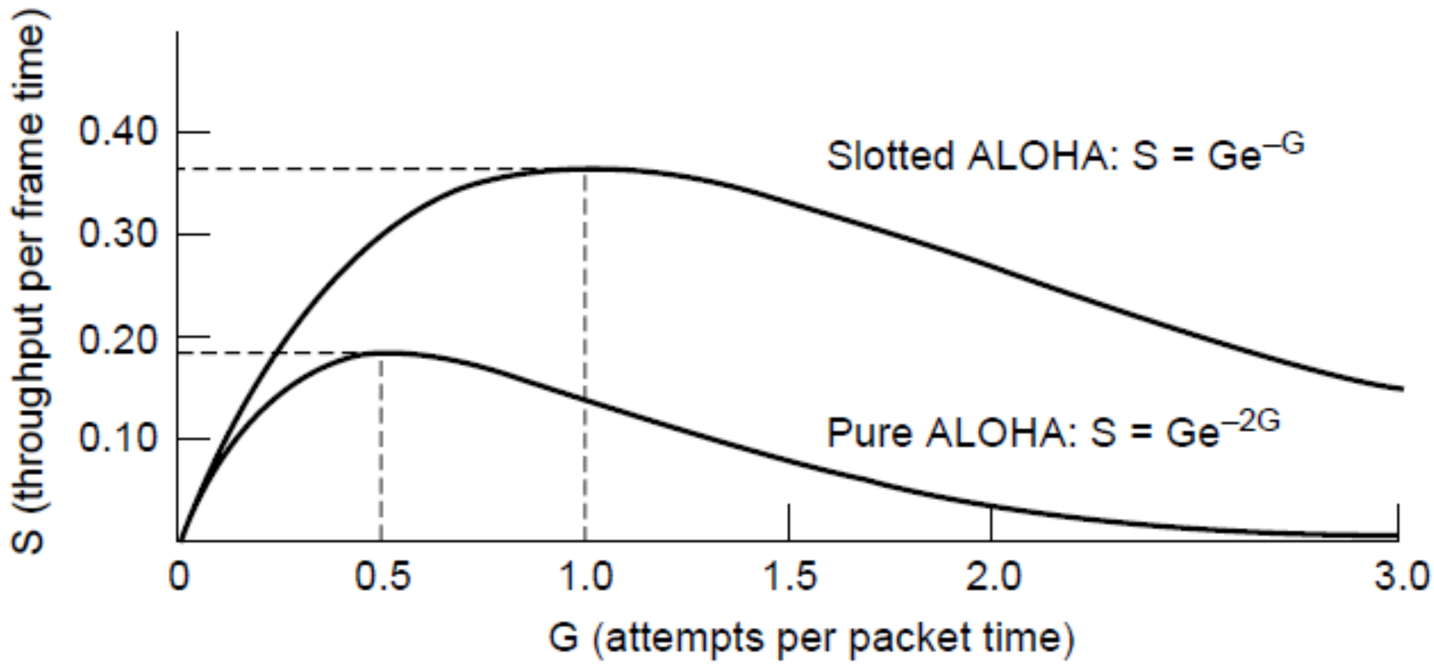
!

even worse than slotted Aloha!

ALOHA (3)

Slotted ALOHA is twice as efficient as pure ALOHA

- Low load wastes slots, high loads causes collisions
- Efficiency up to 1/e (37%) for random traffic models



MAC Protocols: a taxonomy

Three broad classes:

Channel Partitioning

- divide channel into smaller "pieces" (time slots, frequency, code)
- allocate piece to node for exclusive use

Random Access

- channel not divided, allow collisions
- "recover" from collisions

"Taking turns"

- nodes take turns, but nodes with more to send can take longer turns

CSMA (1)

CSMA improves on ALOHA by sensing the channel!

- User doesn't send if it senses someone else

Variations on what to do if the channel is busy:

- **1-persistent** (greedy) sends as soon as idle
- **Nonpersistent** waits a random time then tries again
- **p-persistent** sends with probability p when idle

CSMA (Carrier Sense Multiple Access)

CSMA: listen before transmit:

1. If channel sensed idle: transmit entire frame

2. If channel sensed busy, defer transmission: waits ("backs off") a random amount of time and then senses the channel.

- human analogy: don't interrupt others!

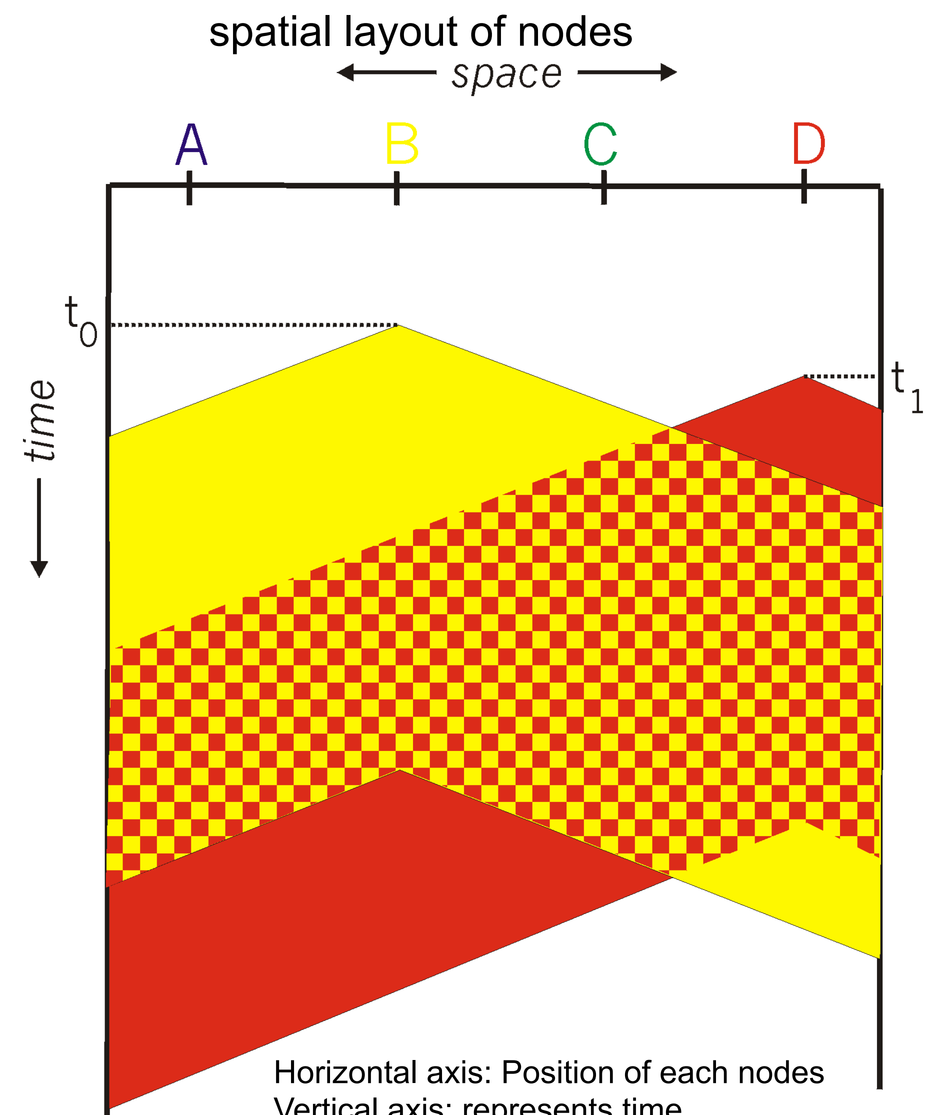
CSMA collisions

collisions can still occur:
propagation delay means two nodes may not hear each other's transmission

collision:
entire packet transmission time wasted

note:
role of distance & propagation delay in determining collision probability

Four nodes attached to a linear broadcast bus



Horizontal axis: Position of each nodes
Vertical axis: represents time

CSMA/CD (Collision Detection)

CSMA/CD: carrier sensing, deferral as in CSMA

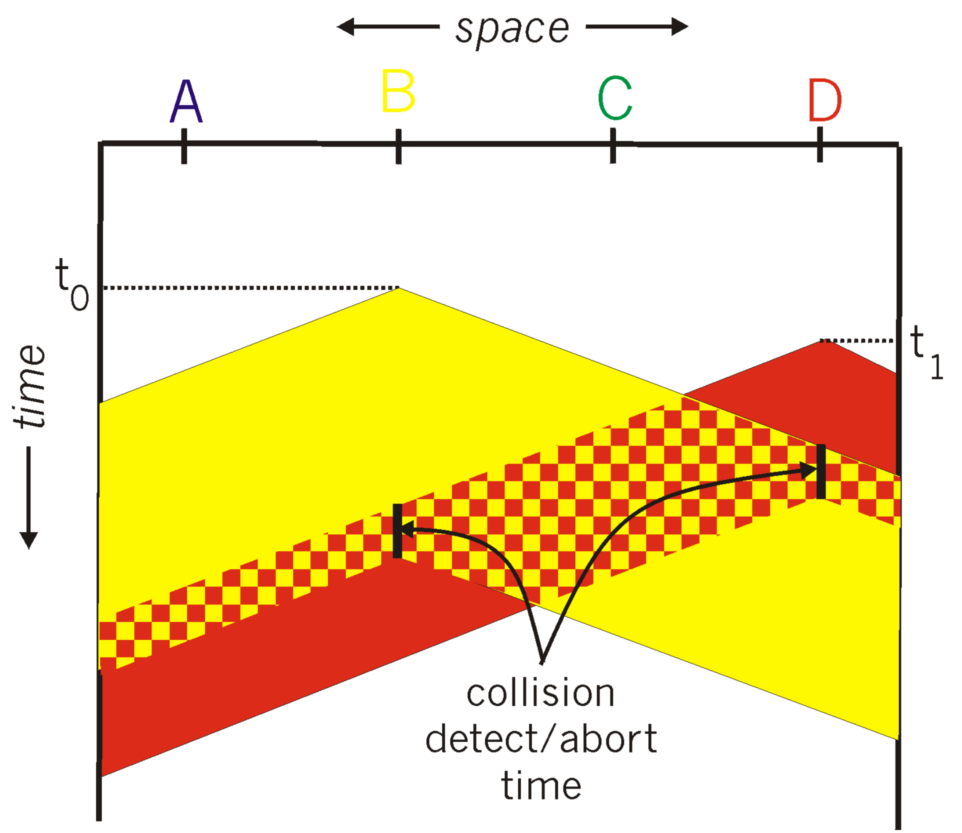
- If someone else begins talking at the same time, stop talking
- collisions *detected* within short time
- colliding transmissions aborted, reducing channel wastage

collision detection:

- easy in wired LANs: measure signal strengths, compare transmitted, received signals
- difficult in wireless LANs: received signal strength overwhelmed by local transmission strength

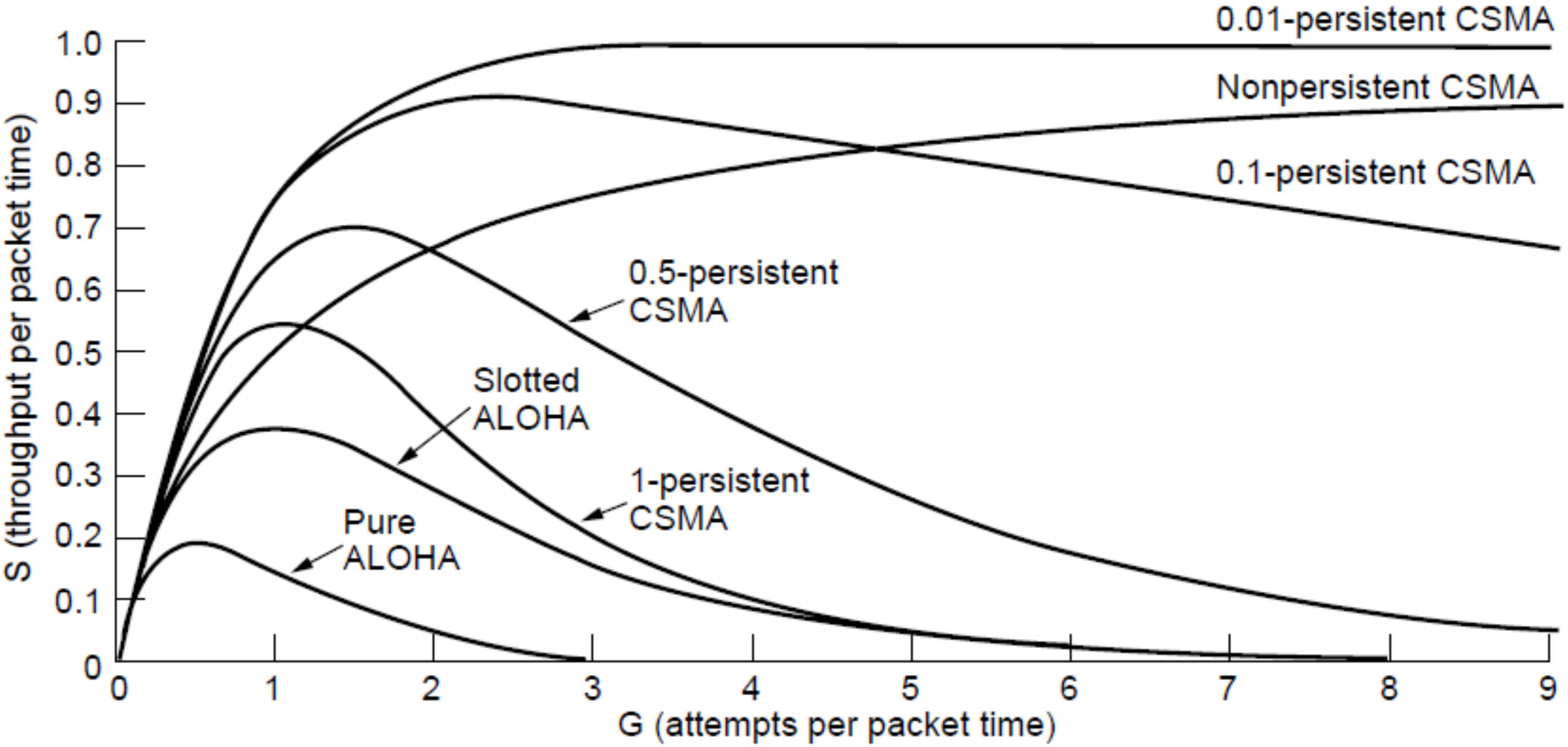
human analogy: the polite conversationalist

CSMA/CD collision detection



CSMA (2) – Persistence

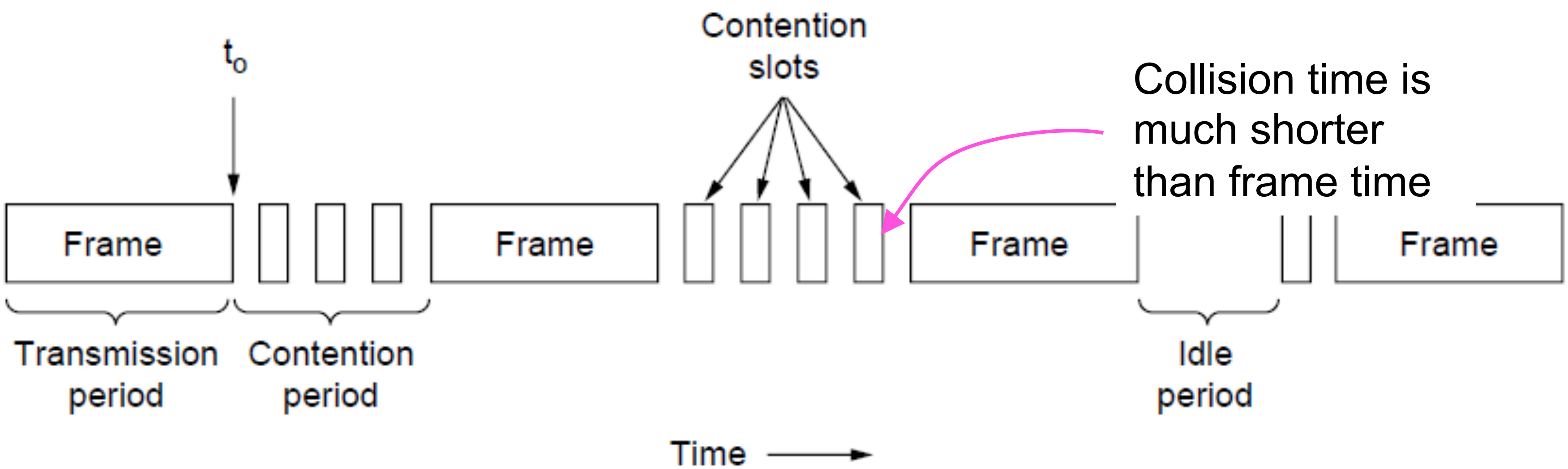
CSMA outperforms ALOHA, and being less persistent is better under high load



CSMA (3) – Collision Detection

CSMA/CD improvement is to detect/abort collisions

- Reduced contention times improve performance



CSMA/CD can be in one of three states: contention, transmission, or idle

Summary of MAC protocols

channel partitioning, by time, frequency or code

- Time Division, Frequency Division

random access (dynamic),

- ALOHA, S-ALOHA, CSMA, CSMA/CD
- carrier sensing: easy in some technologies (wire), hard in others (wireless)
- CSMA/CD used in Ethernet
- CSMA/CA used in 802.11

MAC Protocols: a taxonomy

Three broad classes:

Channel Partitioning

- divide channel into smaller “pieces” (time slots, frequency, code)
- allocate piece to node for exclusive use

Random Access

- channel not divided, allow collisions
- “recover” from collisions

“Taking turns”

- nodes take turns, but nodes with more to send can take longer turns