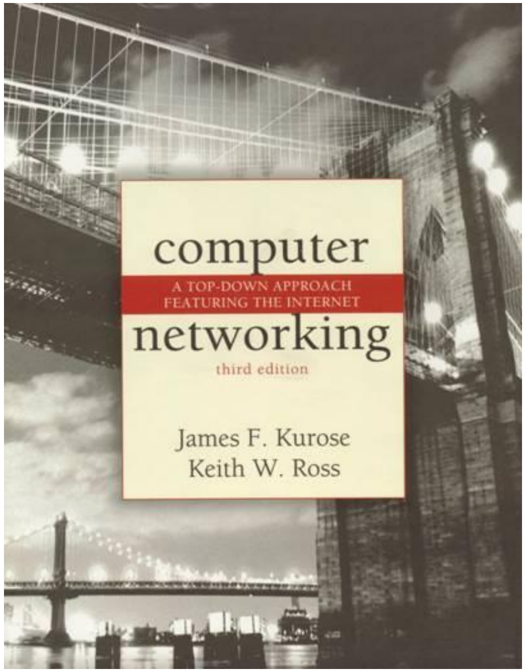
Chapter 1
Introduction



A note on the use of these ppt slides:

We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use, we only ask the following:

- If you use these slides (e.g., in a class) in substantially unaltered form, that you mention their source (after all, we'd like people to use our book!)
- If you post any slides in substantially unaltered form on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

Thanks and enjoy! JFK/KWR

Computer Networking: A Top Down Approach Featuring the Internet, 3rd edition.
Jim Kurose, Keith Ross
Addison-Wesley, July 2004.

Chapter 1: Introduction

Our goal:

- get "feel" and terminology
- more depth, detail *later* in course
- approach:
  - use Internet as example

Overview:

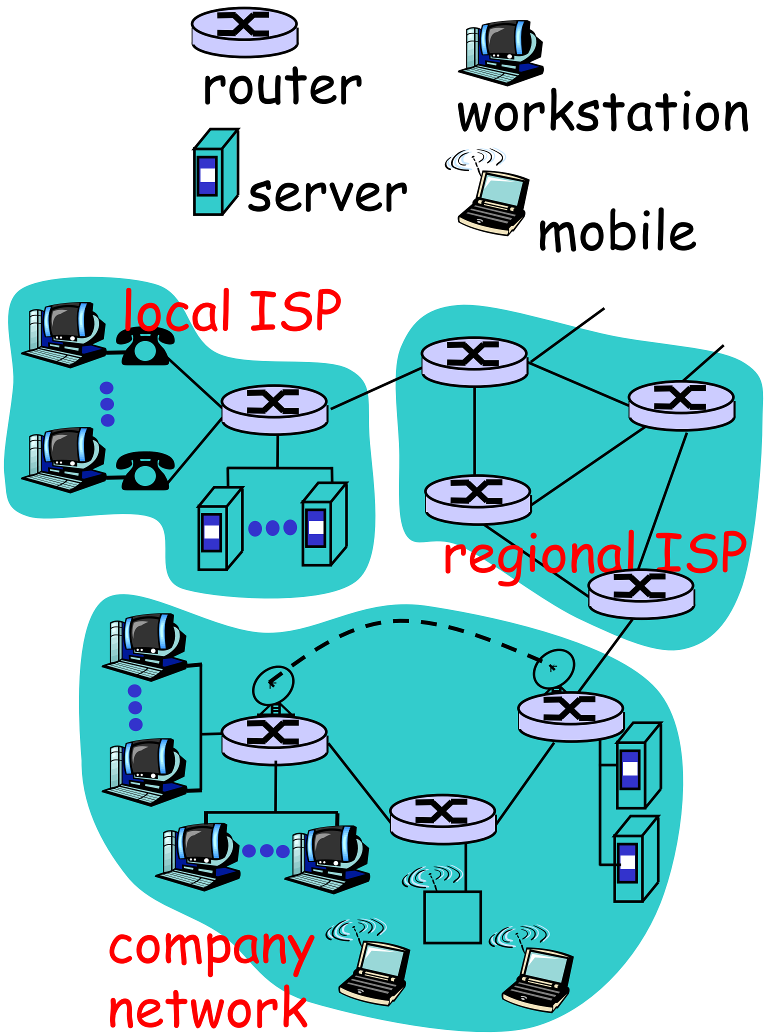
- what's the Internet
- what's a protocol?
- network edge
- network core
- access net, physical media
- Internet/ISP structure
- performance: loss, delay
- protocol layers, service models
- network modeling

Chapter 1: roadmap

1.1 What *is* the Internet?
1.2 Network edge
1.3 Network core
1.4 Network access and physical media
1.5 Internet structure and ISPs
1.6 Delay & loss in packet-switched networks
1.7 Protocol layers, service models
1.8 History

What's the Internet: "nuts and bolts" view

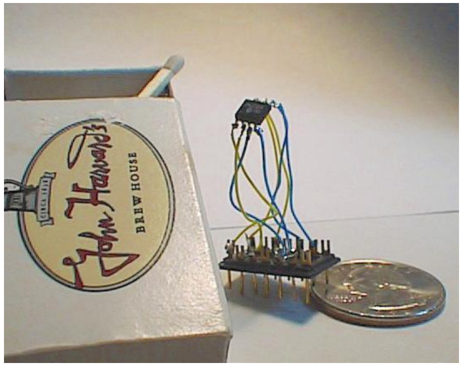
- millions of connected computing devices: *hosts = end systems*
- running *network apps*
- *communication links*
  - fiber, copper, radio, satellite
  - transmission rate = ***bandwidth***
- *routers:* forward packets (chunks of data)



“Cool” internet appliances

IP picture frame
http://www.ceiva.com/

Web-enabled toaster +
weather forecaster

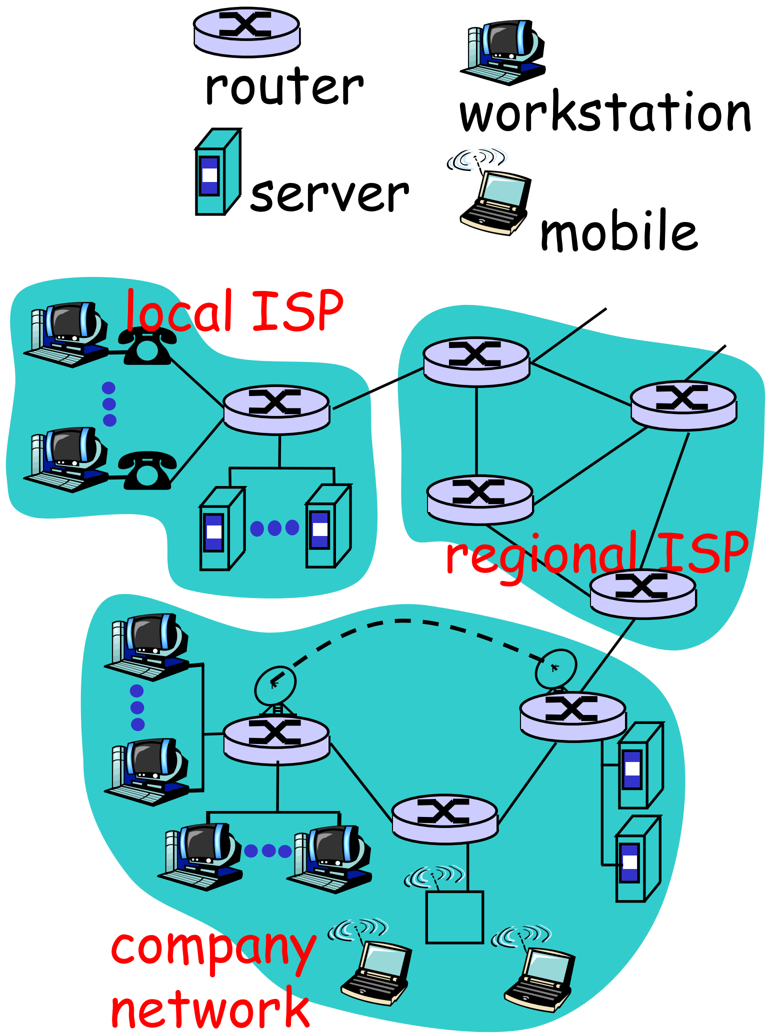


World’s smallest web server
http://www-ccs.cs.umass.edu/~shri/iPic.html

Internet phones

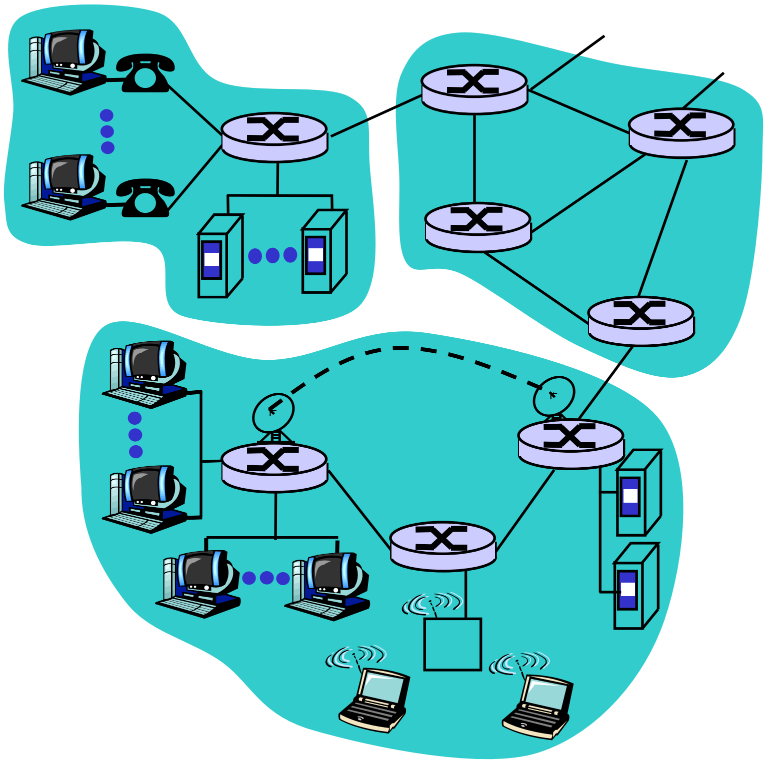
What's the Internet: "nuts and bolts" view

- *protocols* control sending, receiving of msgs
  - e.g., TCP, IP, HTTP, FTP, PPP
- *Internet:* "network of networks"
  - loosely hierarchical
  - public Internet versus private intranet
- Internet standards
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force



What's the Internet: a service view

- communication *infrastructure* enables distributed applications:
  - Web, email, games, e-commerce, file sharing
- communication services provided to apps:
  - Connectionless unreliable
  - connection-oriented reliable



What's a protocol?

human protocols:

- "what's the time?"
- "I have a question"
- introductions

… specific msgs sent

… specific actions taken when msgs received, or other events

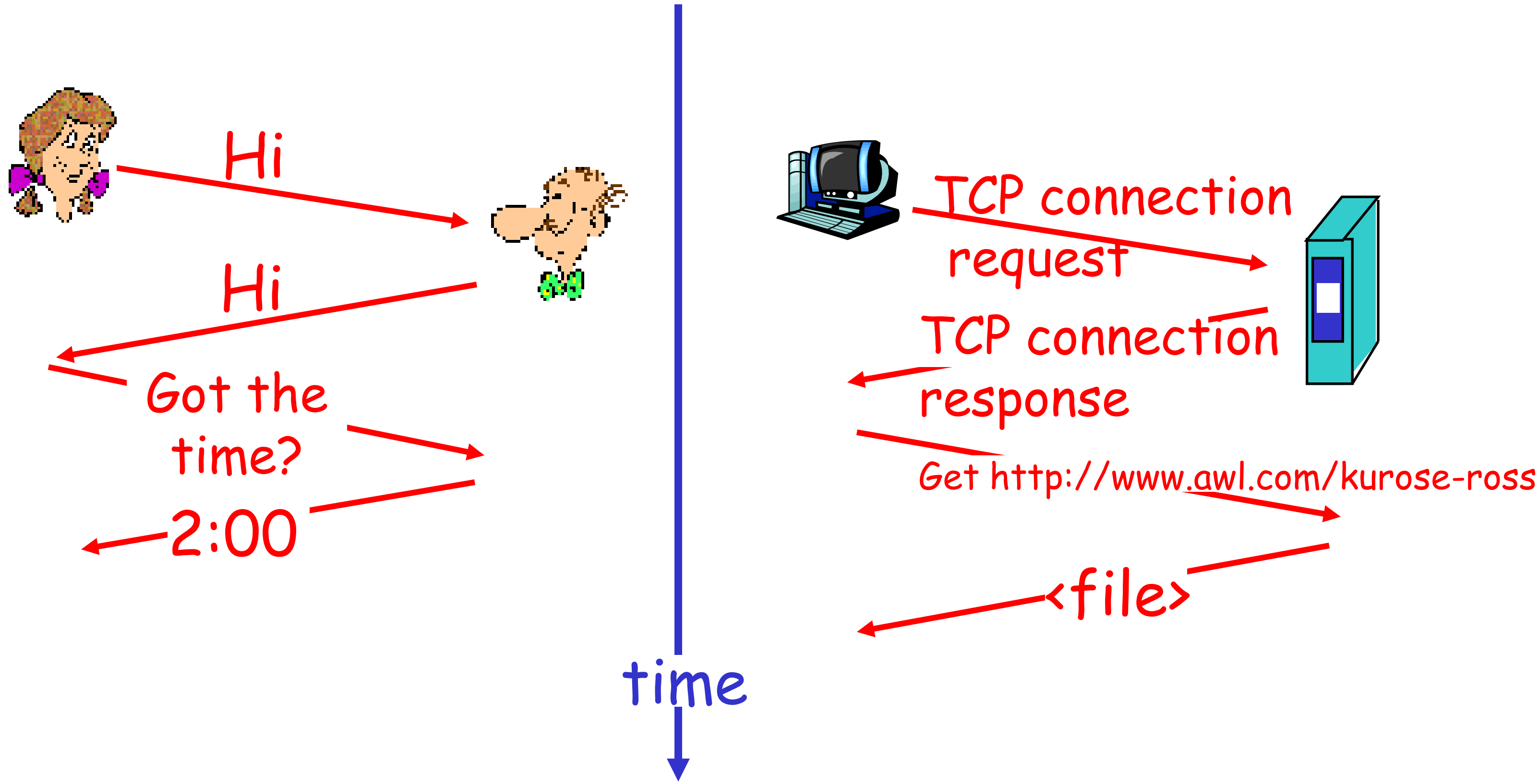
network protocols:

- machines rather than humans
- all communication activity in Internet governed by protocols

protocols define format, order of msgs sent and received among network entities, and actions taken on msg transmission, receipt

What's a protocol?

a human protocol and a computer network protocol:



Q: Other human protocols?

Chapter 1: roadmap

1.1 What *is* the Internet?

1.2 Network edge

1.3 Network core

1.4 Network access and physical media

1.5 Internet structure and ISPs

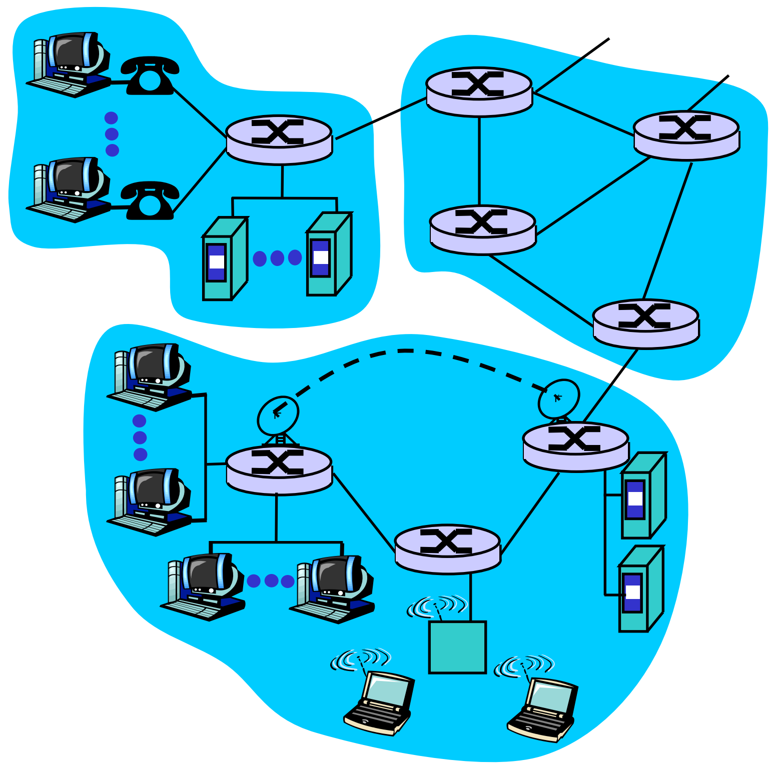
1.6 Delay & loss in packet-switched networks

1.7 Protocol layers, service models

1.8 History

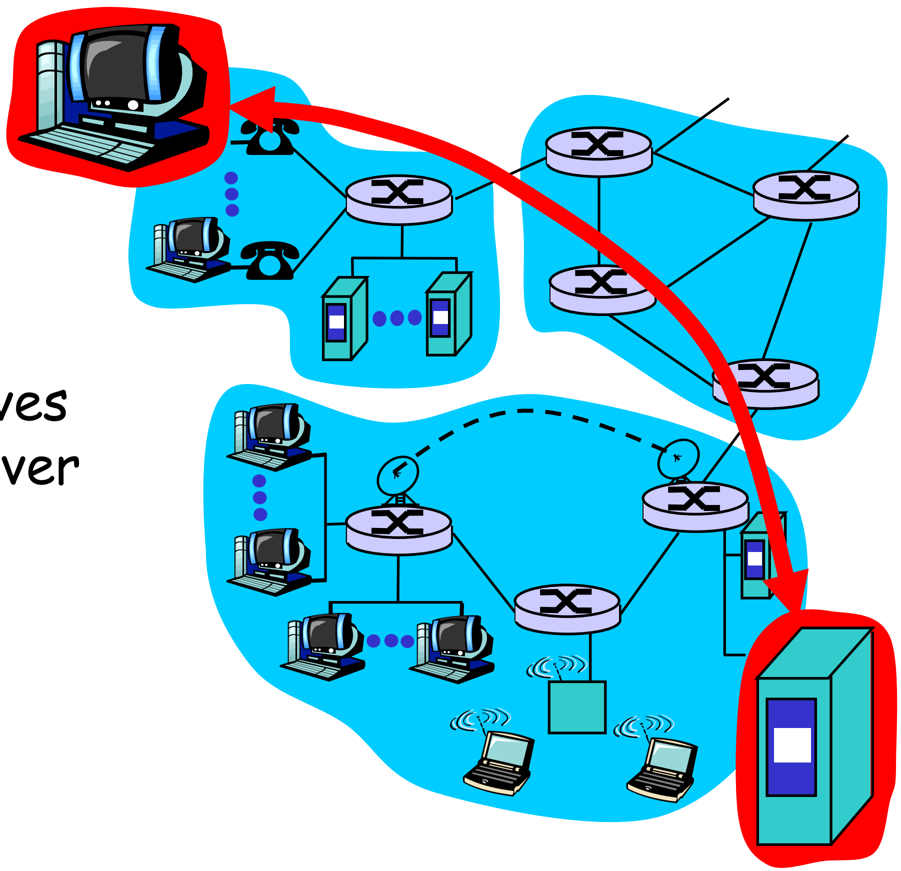
A closer look at network structure:

- network edge: applications and hosts
- network core:
  - routers
  - network of networks
- access networks, physical media: communication links



The network edge:

- end systems (hosts):
  - run application programs
  - e.g. Web, email
  - at “edge of network”
- client/server model
  - client host requests, receives service from always-on server
  - e.g. Web browser/server; email client/server
- peer-peer model:
  - minimal (or no) use of dedicated servers
  - e.g. Gnutella, KaZaA, Skype



Network edge: connection-oriented service

Goal: data transfer between end systems

- *handshaking:* setup (prepare for) data transfer ahead of time
  - Hello, hello back human protocol
  - *set up "state"* in two communicating hosts
- TCP - Transmission Control Protocol
  - Internet's connection-oriented service

TCP service [RFC 793]

- *reliable, in-order* byte-stream data transfer
  - loss: acknowledgements and retransmissions
- *flow control:*
  - sender won't overwhelm receiver
- *congestion control:*
  - senders "slow down sending rate" when network congested

Network edge: connectionless service

Goal: data transfer between end systems
- same as before!

- UDP - User Datagram Protocol [RFC 768]:
  - connectionless
  - unreliable data transfer
  - no flow control
  - no congestion control

App's using TCP:

- HTTP (Web), FTP (file transfer), Telnet (remote login), SMTP (email)

App's using UDP:

- streaming media, teleconferencing, DNS, Internet telephony

Chapter 1: roadmap

1.1 What *is* the Internet?

1.2 Network edge

1.3 Network core

1.4 Network access and physical media

1.5 Internet structure and ISPs

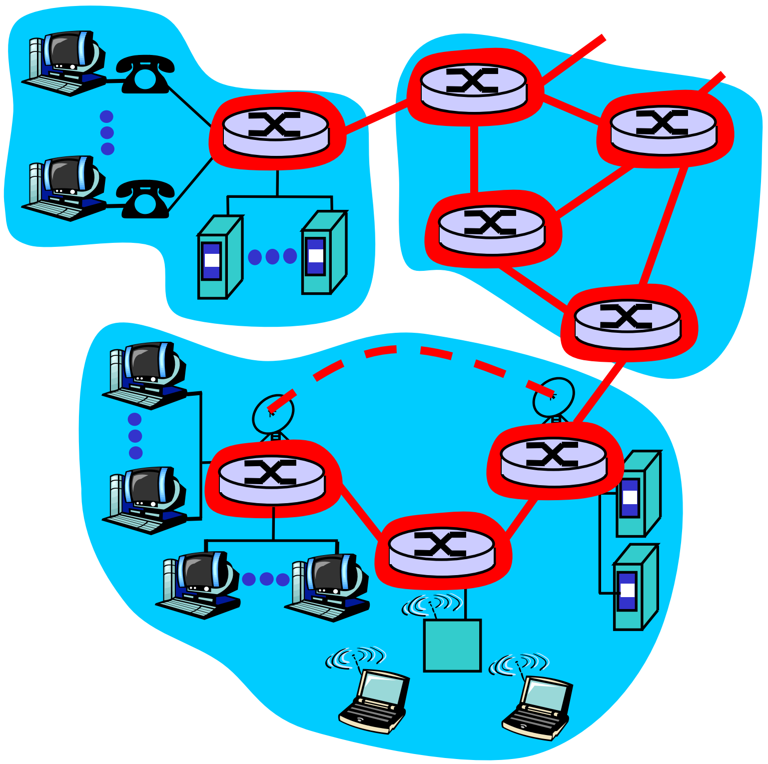
1.6 Delay & loss in packet-switched networks

1.7 Protocol layers, service models

1.8 History

The Network Core

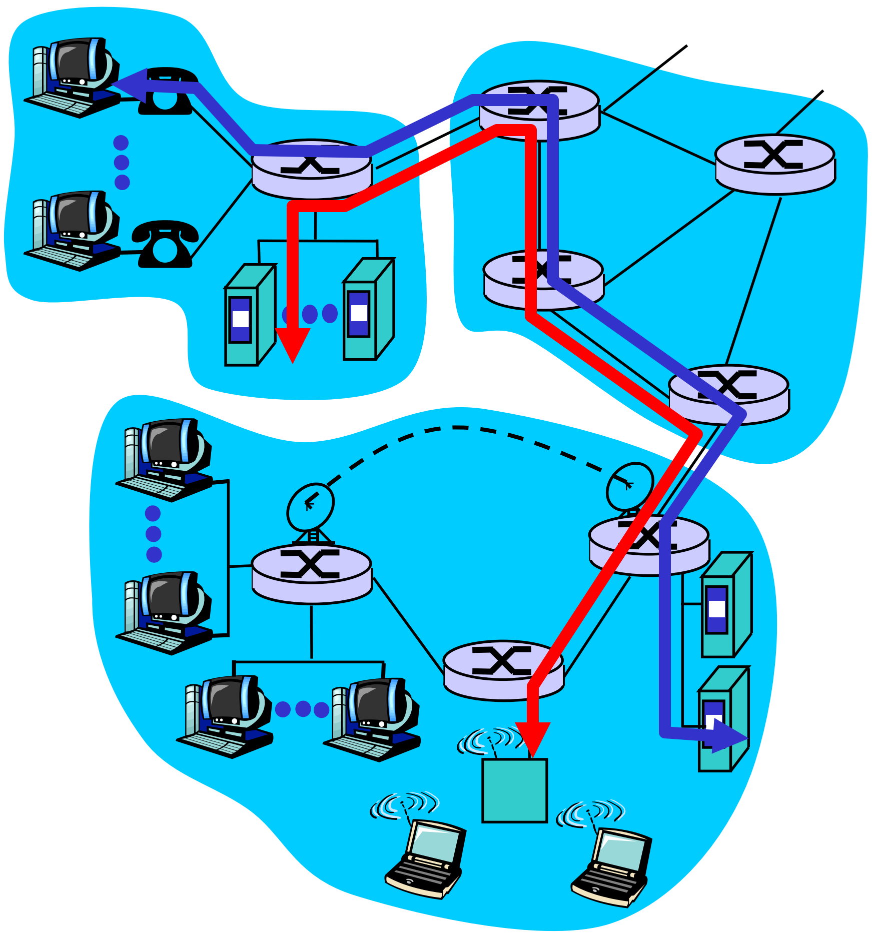
- mesh of interconnected routers
- *the* fundamental question: how is data transferred through net?
  - circuit switching: dedicated circuit per call: telephone net
  - packet-switching: data sent thru net in discrete "chunks"



Network Core: Circuit Switching

End-end resources reserved for "call"

- link bandwidth, switch capacity
- dedicated resources: no sharing
- circuit-like (guaranteed) performance
- call setup required



Network Core: Circuit Switching

network resources (e.g., bandwidth) divided into "pieces"

- pieces allocated to calls
- resource piece *idle* if not used by owning call *(no sharing)*

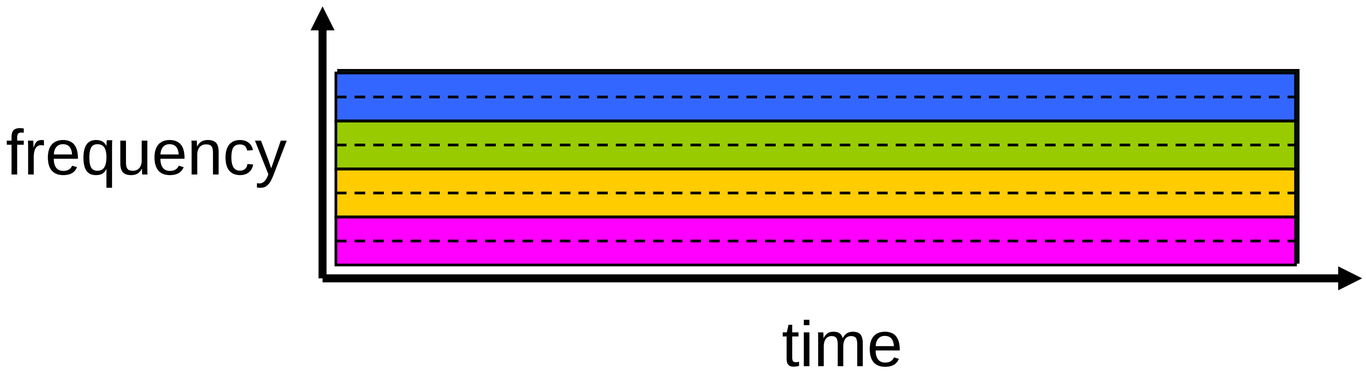
- dividing link bandwidth into "pieces"
  - frequency division
  - time division

Circuit Switching: FDM and TDM

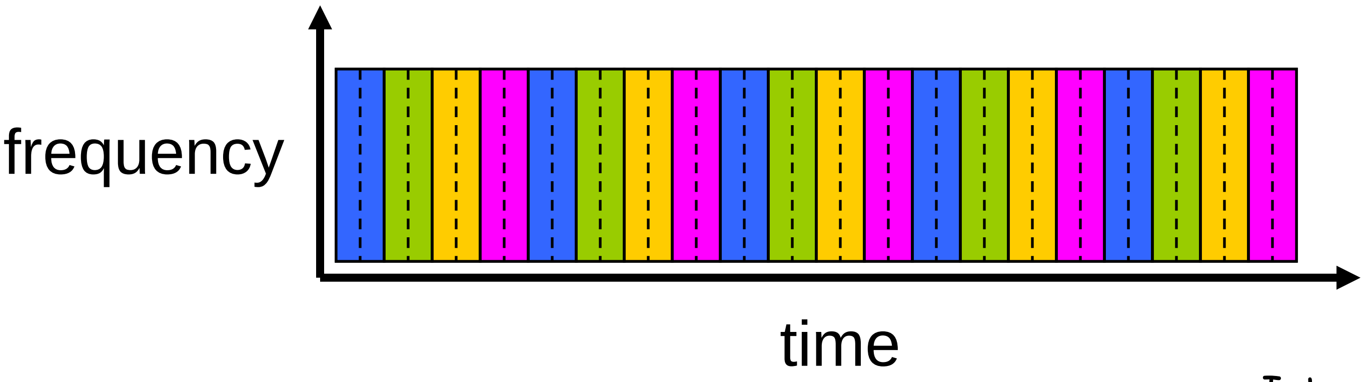
Example:

4 users

FDM



TDM



Numerical example

- How long does it take to send a file of 640,000 bits from host A to host B over a circuit-switched network?
  - All links are 1.536 Mbps
  - Each link uses TDM with 24 slots/sec
  - 500 msec to establish end-to-end circuit

Let's work it out!

Another numerical example

- How long does it take to send a file of 640,000 bits from host A to host B over a circuit-switched network?
  - All links are 1.536 Mbps
  - Each link uses FDM with 24 channels/frequencies
  - 500 msec to establish end-to-end circuit

Let's work it out!

Network Core: Packet Switching

each end-end data stream divided into *packets*

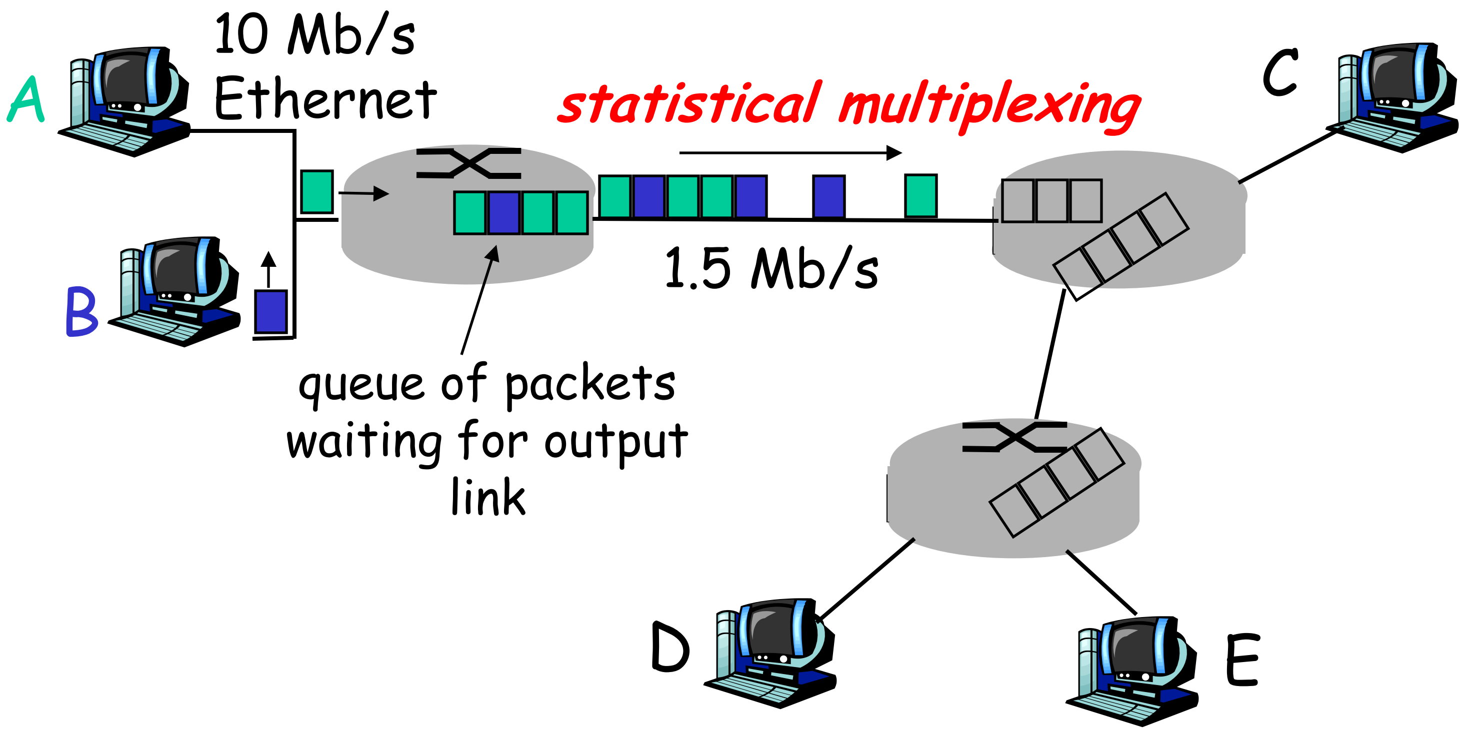
- user A, B packets *share* network resources
- each packet uses full link bandwidth
- resources used *as needed*

Bandwidth division into "pieces"
Dedicated allocation
Resource reservation

resource contention:

- aggregate resource demand can exceed amount available
- congestion: packets queue, wait for link use
- store and forward: packets move one hop at a time
  - Node receives complete packet before forwarding

Packet Switching: Statistical Multiplexing



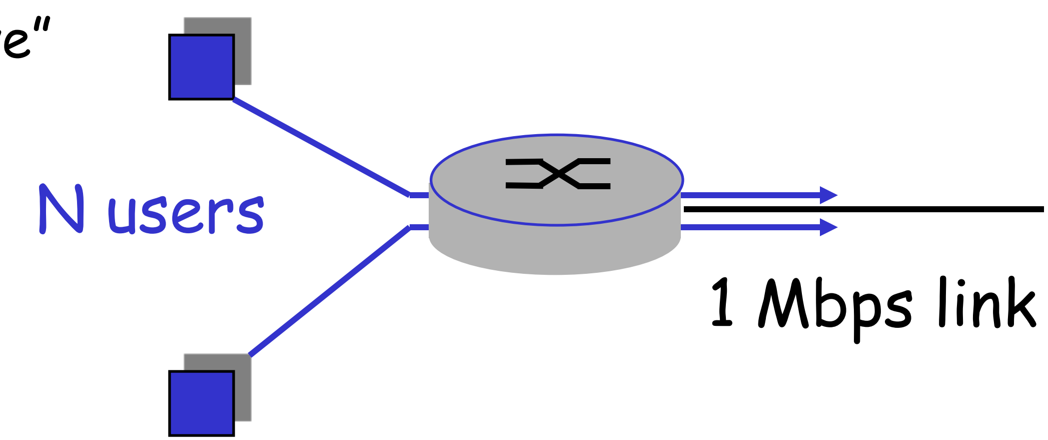
Sequence of A & B packets does not have fixed pattern, shared on demand ➔ ***statistical multiplexing***.

TDM: each host gets same slot in revolving TDM frame.

Packet switching versus circuit switching

Packet switching allows more users to use network!

- 1 Mb/s link
- each user:
  - 100 kb/s when "active"
  - active 10% of time
- circuit-switching:
  - 10 users
- packet switching:
  - with 35 users, probability > 10 active less than .0004



Q: how did we get value 0.0004?

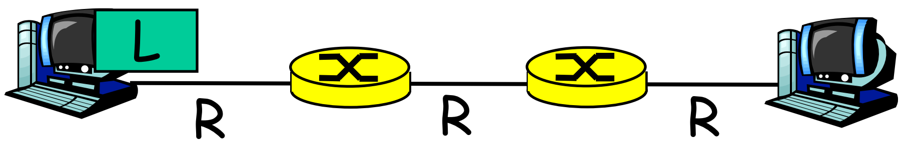
Packet switching versus circuit switching

Is packet switching a "slam dunk winner?"

- Great for bursty data
  - resource sharing
  - simpler, no call setup
- Excessive congestion: packet delay and loss
  - protocols needed for reliable data transfer, congestion control
- Q: How to provide circuit-like behavior?
  - bandwidth guarantees needed for audio/video apps
  - still an unsolved problem (chapter 7)

Q: human analogies of reserved resources (circuit switching) versus on-demand allocation (packet-switching)?

Packet-switching: store-and-forward



- Takes L/R seconds to transmit (push out) packet of L bits on to link or R bps
- Entire packet must arrive at router before it can be transmitted on next link: *store and forward*
- delay = 3L/R (assuming zero propagation delay)

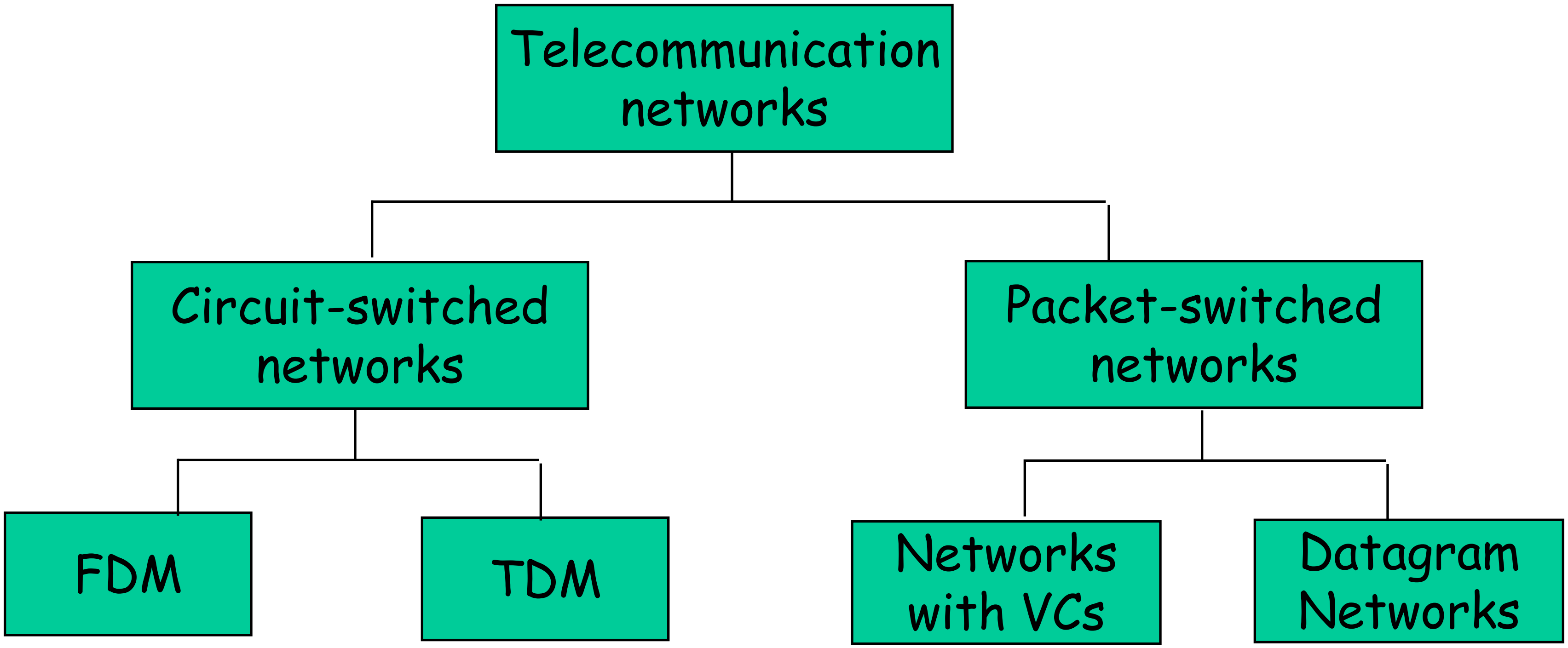
more on delay shortly …

Example:
- L = 7.5 Mbits
- R = 1.5 Mbps
- delay = 15 sec

Packet-switched networks: forwarding

- *Goal:* move packets through routers from source to destination
  - we'll study several path selection (i.e. routing) algorithms (chapter 4)
- datagram network:
  - *destination address* in packet determines next hop
  - routes may change during session
  - analogy: driving, asking directions
- virtual circuit network:
  - each packet carries tag (virtual circuit ID), tag determines next hop
  - fixed path determined at *call setup time*, remains fixed thru call
  - *routers maintain per-call state*

Network Taxonomy



- Datagram network is *not* either connection-oriented or connectionless.
- Internet provides both connection-oriented (TCP) and connectionless services (UDP) to apps.

Chapter 1: roadmap

1.1 What *is* the Internet?

1.2 Network edge

1.3 Network core

1.4 Network access and physical media

1.5 Internet structure and ISPs

1.6 Delay & loss in packet-switched networks

1.7 Protocol layers, service models

1.8 History

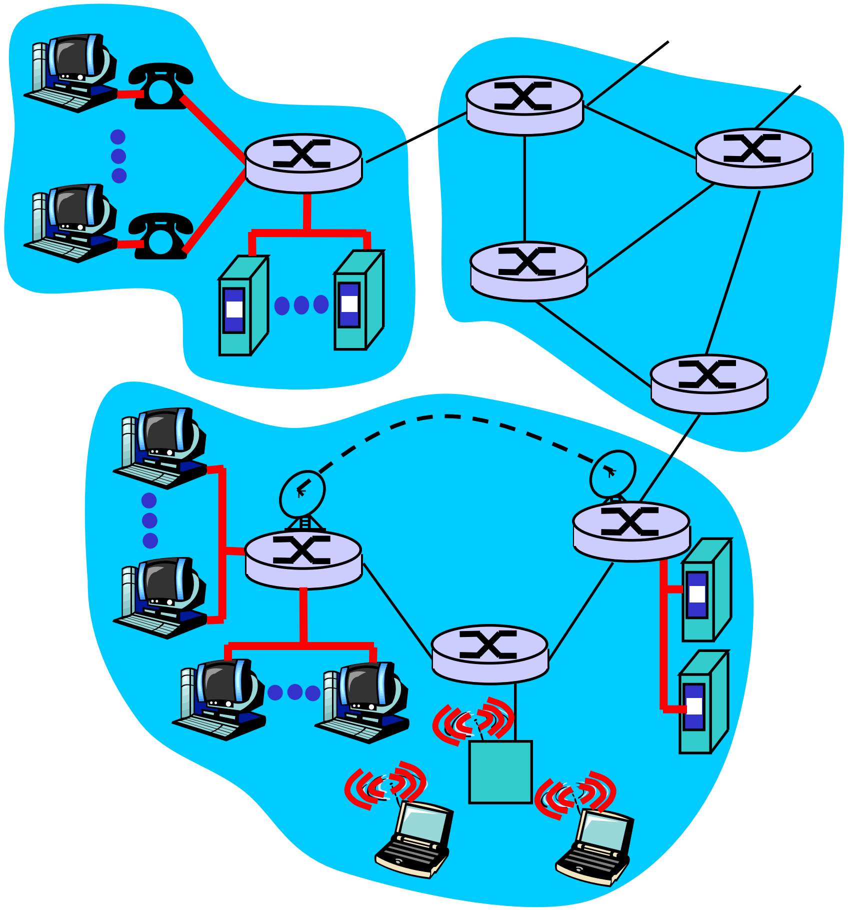
Access networks and physical media

Q: How to connect end systems to edge router?

- residential access nets
- institutional access networks (school, company)
- mobile access networks

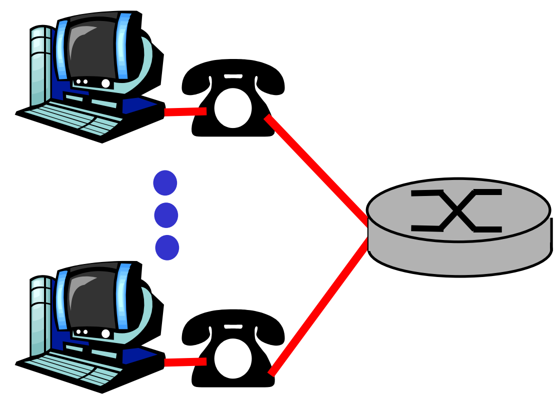
Keep in mind:

- bandwidth (bits per second) of access network?
- shared or dedicated?



Residential access: point to point access

- Dialup via modem
  - up to 56Kbps direct access to router (often less)
  - Can't surf and phone at same time: can't be "always on"



- ADSL: asymmetric digital subscriber line
  - up to 1 Mbps upstream (today typically < 256 kbps)
  - up to 8 Mbps downstream (today typically < 1 Mbps)
  - FDM: 50 kHz - 1 MHz for downstream

    4 kHz - 50 kHz for upstream

    0 kHz - 4 kHz for ordinary telephone

Residential access: cable modems

- HFC: hybrid fiber coax
  - asymmetric: up to 30Mbps downstream, 2 Mbps upstream
- network of cable and fiber attaches homes to ISP router
  - homes share access to router
- deployment: available via cable TV companies

Residential access: cable modems

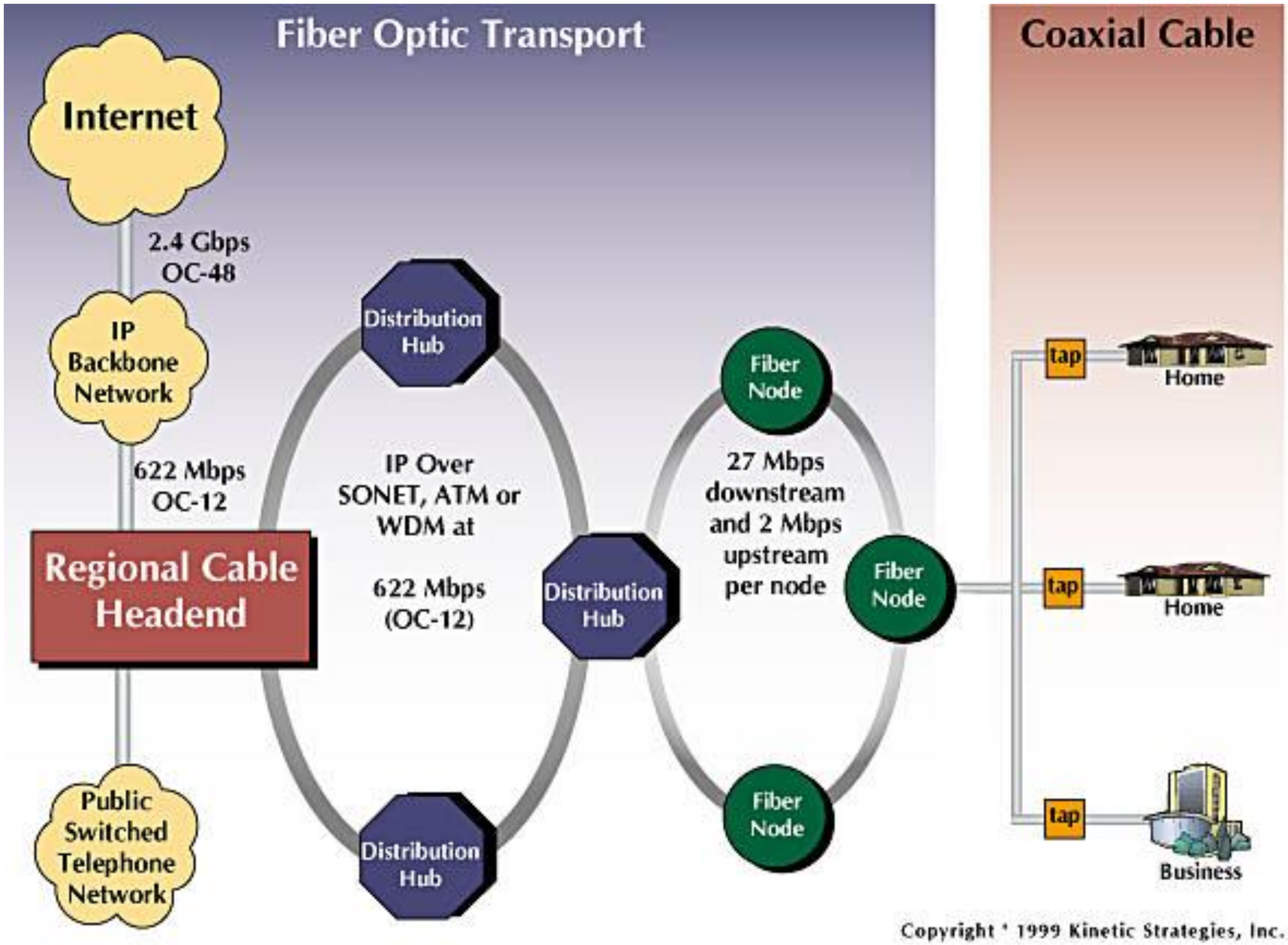
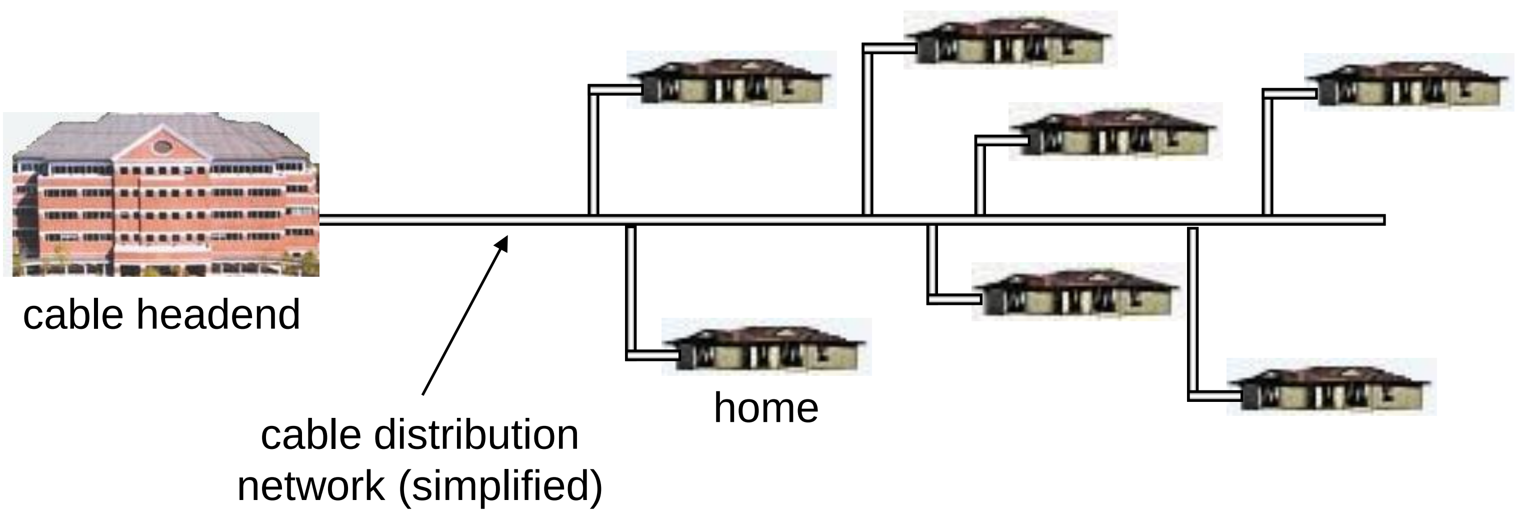


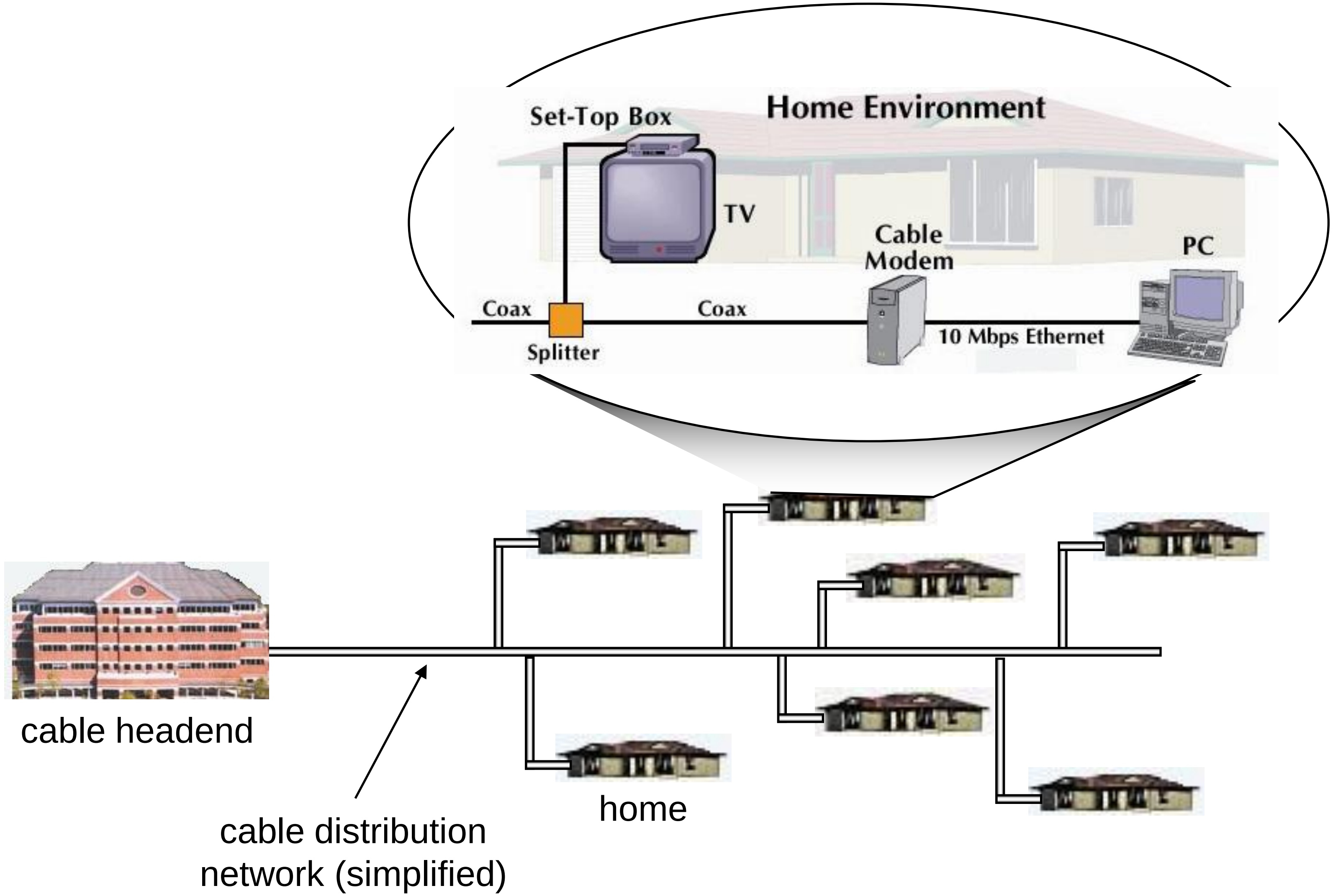
Diagram: http://www.cabledatacomnews.com/cmic/diagram.html

Cable Network Architecture: Overview

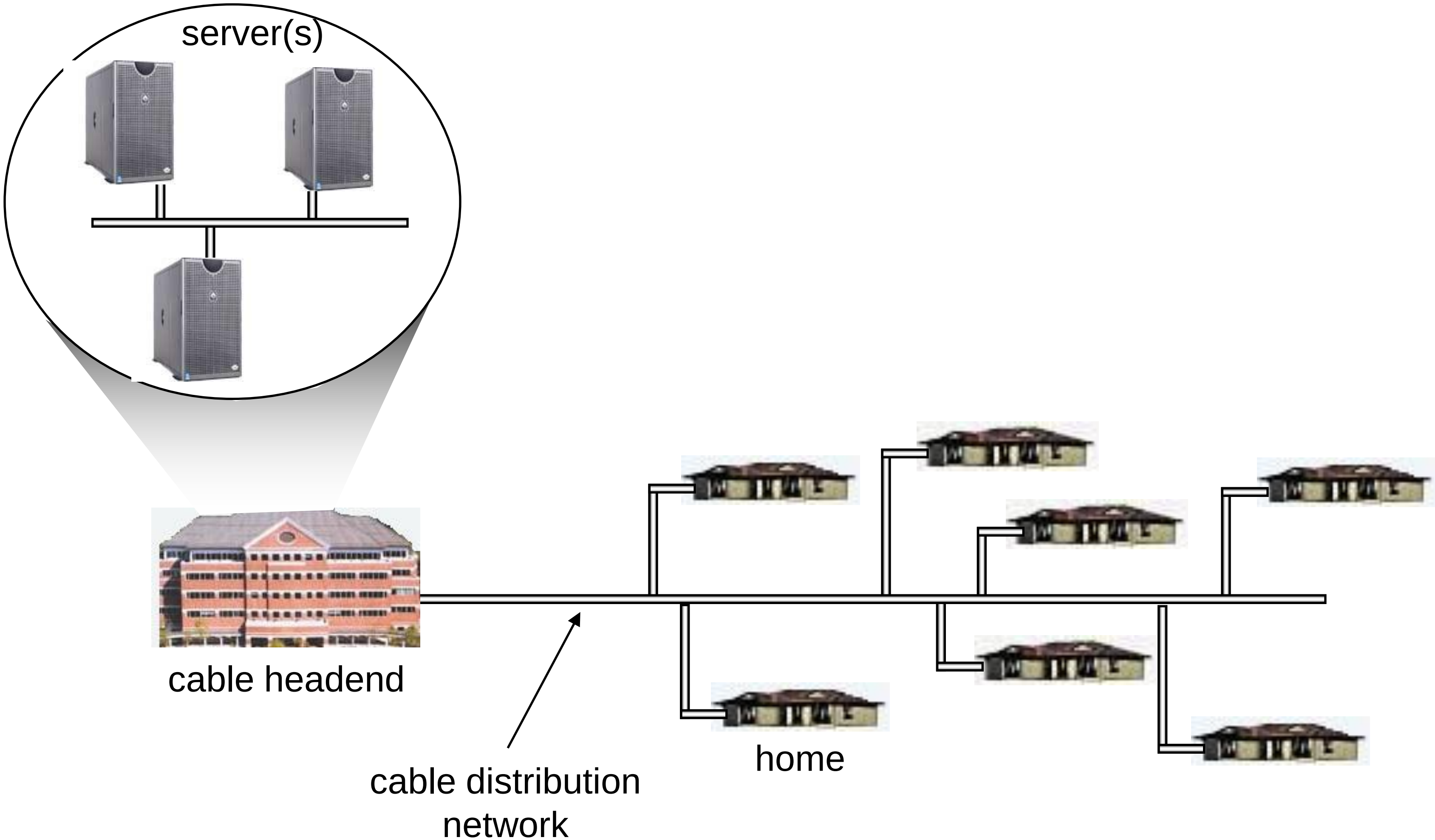
Typically 500 to 5,000 homes



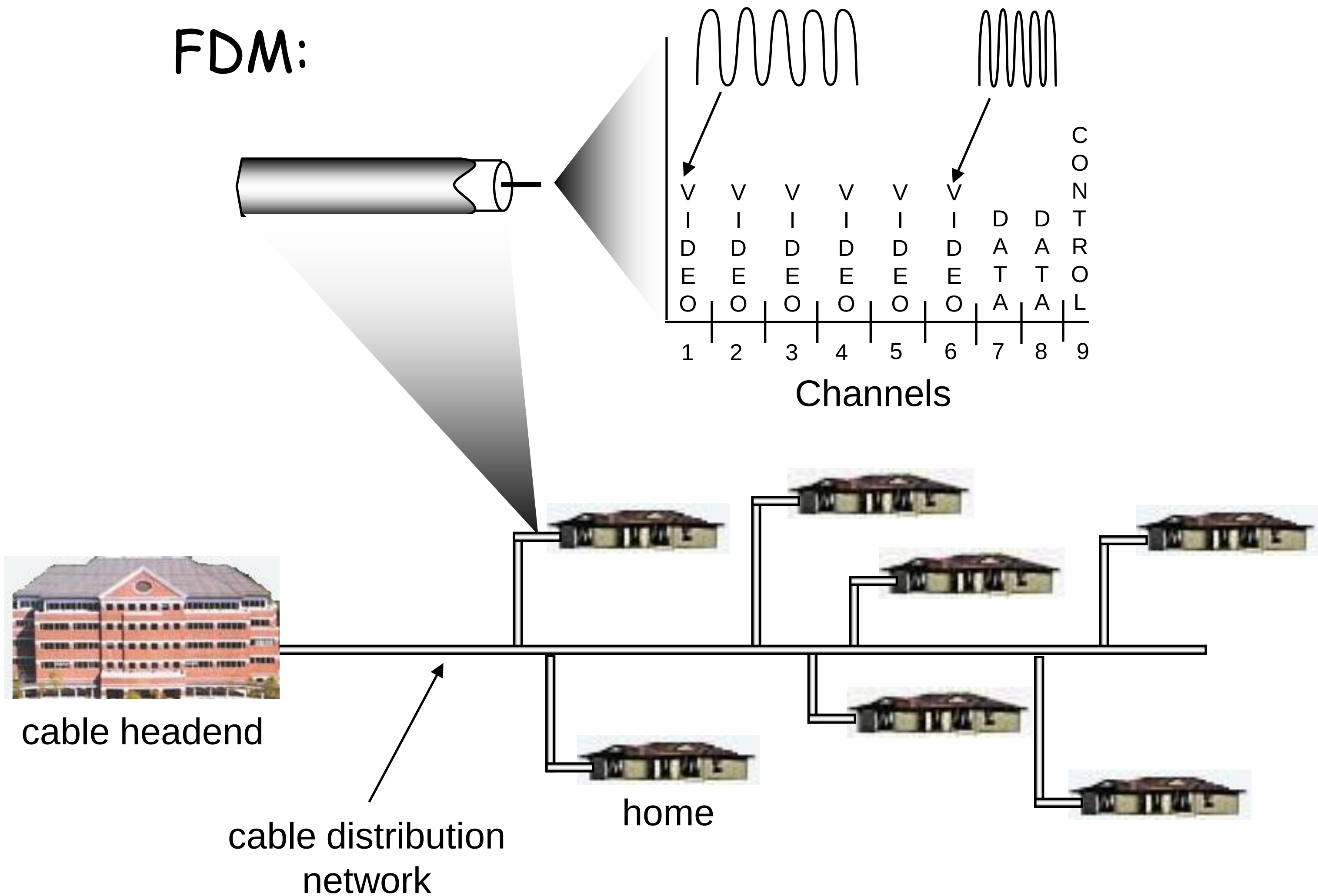
Cable Network Architecture: Overview



Cable Network Architecture: Overview

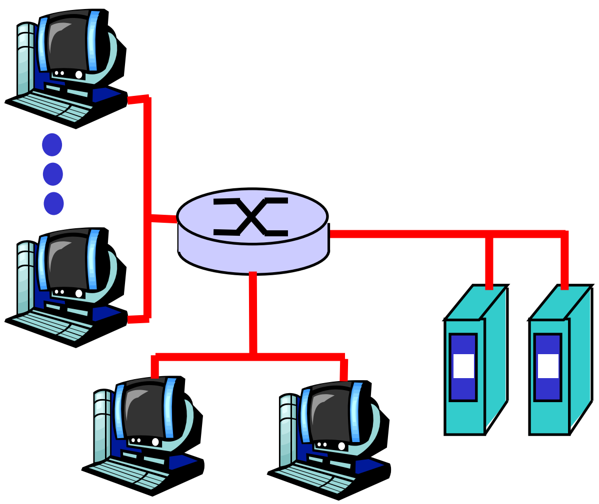


Cable Network Architecture: Overview



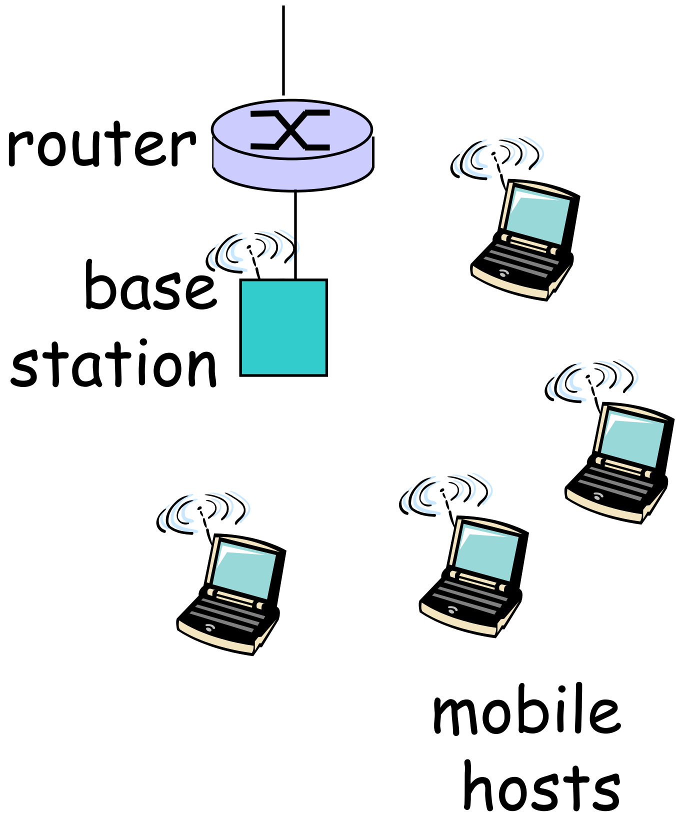
Company access: local area networks

- company/univ local area network (LAN) connects end system to edge router
- Ethernet:
  - shared or dedicated link connects end system and router
  - 10 Mbs, 100Mbps, Gigabit Ethernet
- LANs: chapter 5



Wireless access networks

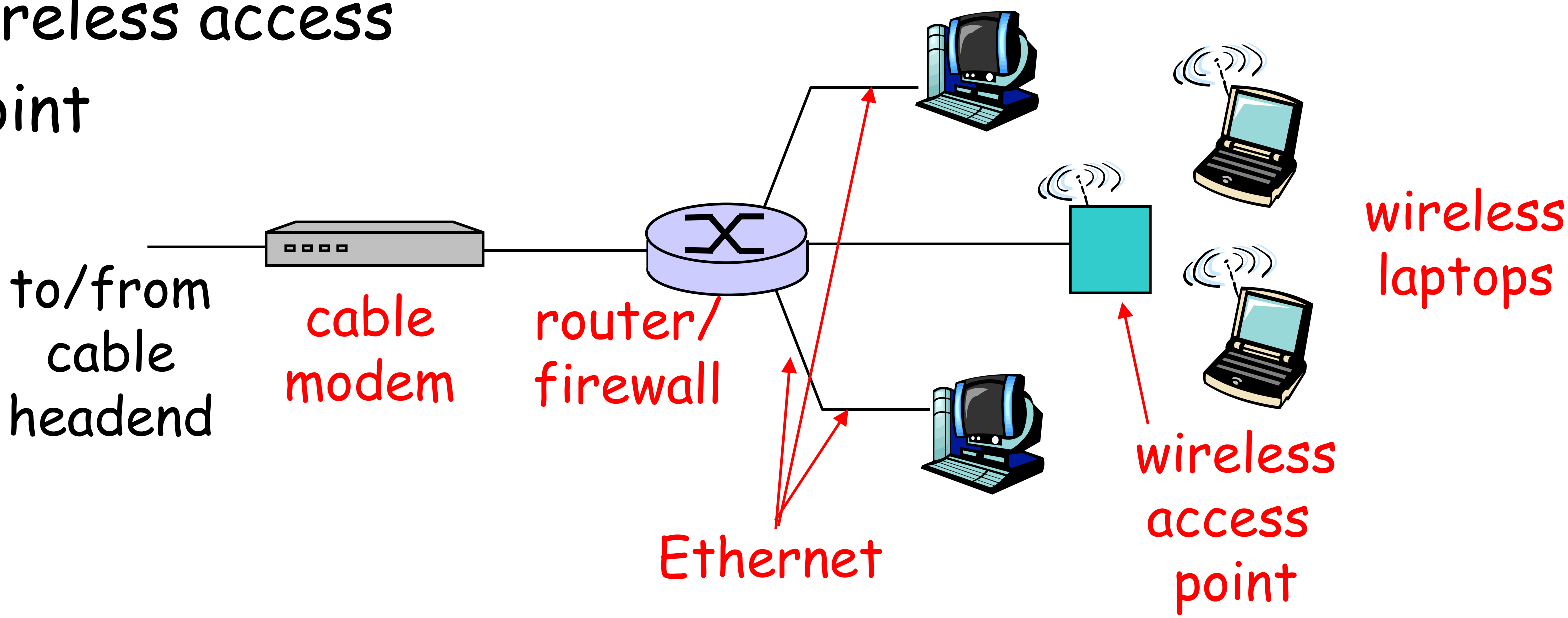
- shared *wireless* access network connects end system to router
  - via base station aka "access point"
- wireless LANs:
  - 802.11b (WiFi): 11 Mbps
- wider-area wireless access
  - provided by telco operator
  - 3G ~ 384 kbps
    - Will it happen??
  - WAP/GPRS in Europe



Home networks

Typical home network components:

- ADSL or cable modem
- router/firewall/NAT
- Ethernet
- wireless access point



Physical Media

- Bit: propagates between transmitter/rcvr pairs
- physical link: what lies between transmitter & receiver
- guided media:
  - signals propagate in solid media: copper, fiber, coax
- unguided media:
  - signals propagate freely, e.g., radio

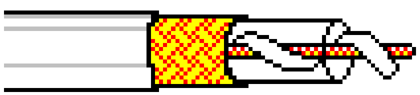
Twisted Pair (TP)

- two insulated copper wires
  - Category 3: traditional phone wires, 10 Mbps Ethernet
  - Category 5: 100Mbps Ethernet

Physical Media: coax, fiber

Coaxial cable:

- two concentric copper conductors
- bidirectional
- baseband:
  - single channel on cable
  - legacy Ethernet
- broadband:
  - multiple channels on cable
  - HFC



Fiber optic cable:

- glass fiber carrying light pulses, each pulse a bit
- high-speed operation:
  - high-speed point-to-point transmission (e.g., 10's-100's Gps)
- low error rate: repeaters spaced far apart ; immune to electromagnetic noise

Physical media: radio

- signal carried in electromagnetic spectrum
- no physical "wire"
- bidirectional
- propagation environment effects:
  - reflection
  - obstruction by objects
  - interference

Radio link types:

- terrestrial microwave
  - e.g. up to 45 Mbps channels
- LAN (e.g., Wifi)
  - 2Mbps, 11Mbps, 54 Mbps
- wide-area (e.g., cellular)
  - e.g. 3G: hundreds of kbps
- satellite
  - Kbps to 45Mbps channel (or multiple smaller channels)
  - 270 msec end-end delay
  - geosynchronous versus low altitude

Chapter 1: roadmap

1.1 What *is* the Internet?

1.2 Network edge

1.3 Network core

1.4 Network access and physical media

1.5 Internet structure and ISPs

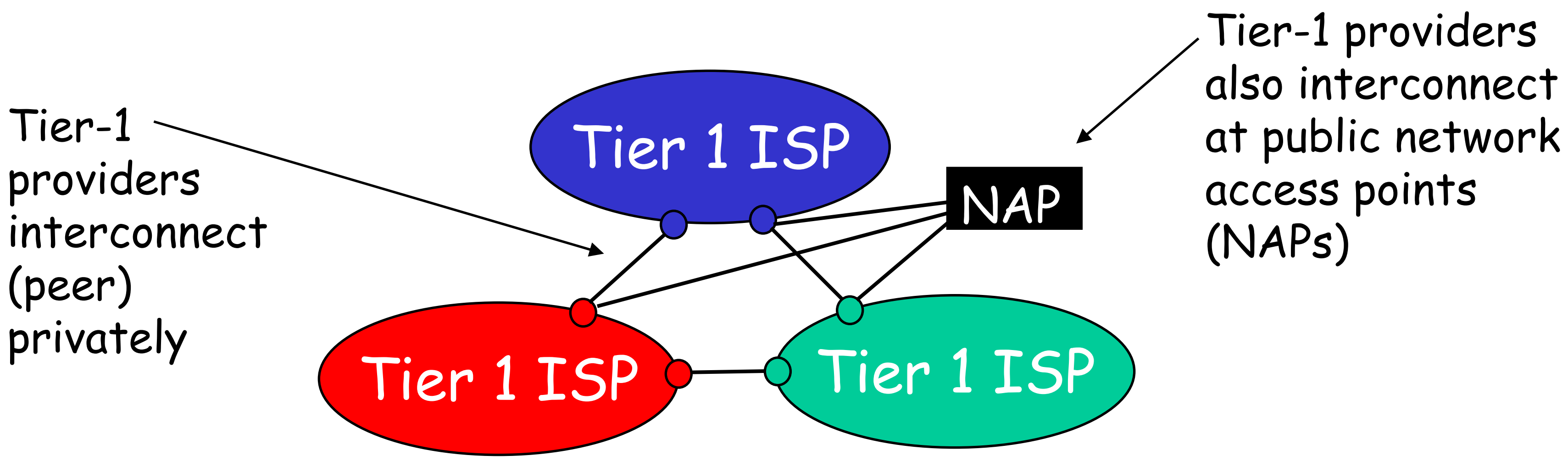
1.6 Delay & loss in packet-switched networks

1.7 Protocol layers, service models

1.8 History

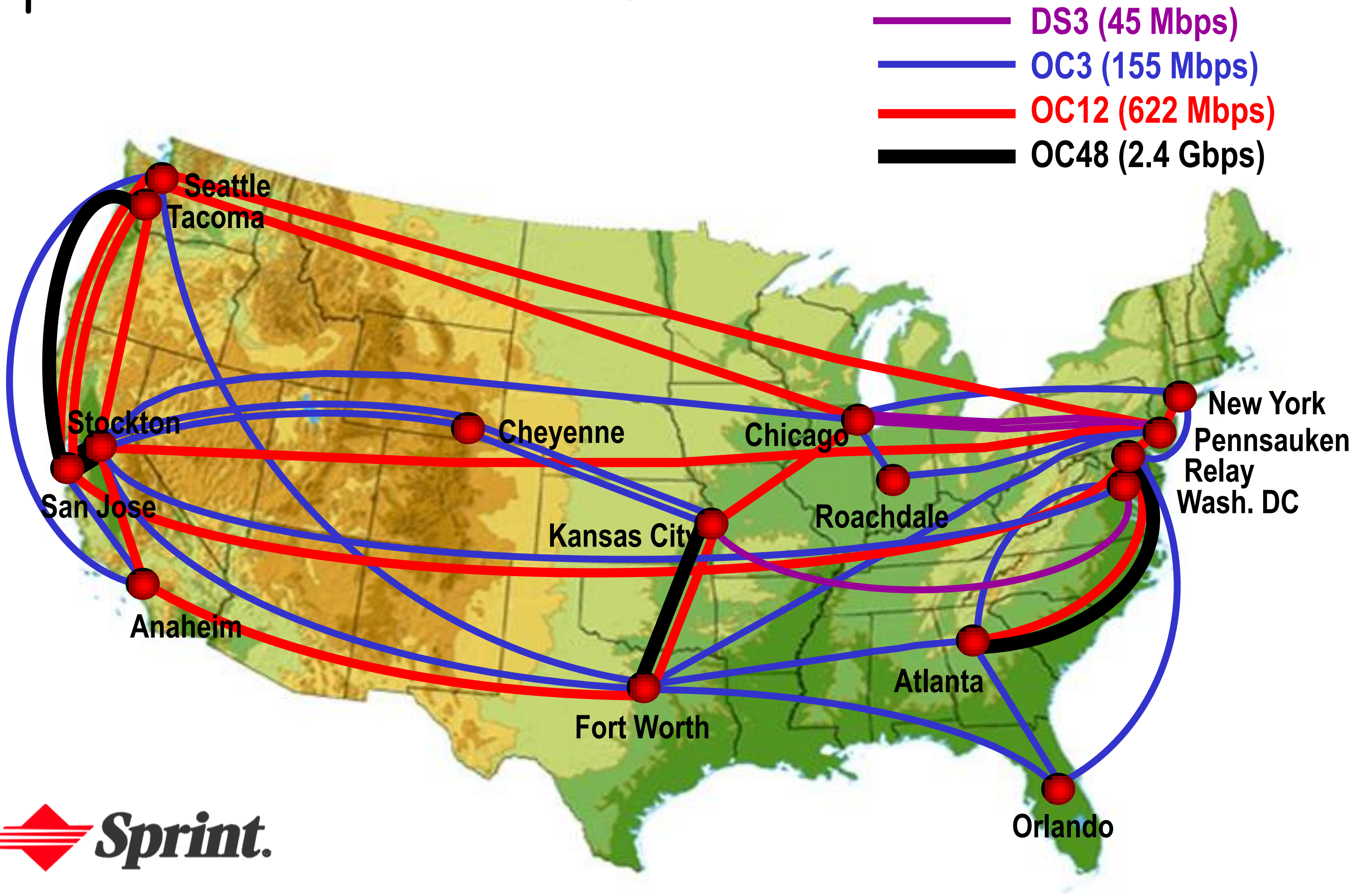
Internet structure: network of networks

- roughly hierarchical
- at center: "tier-1" ISPs (e.g., MCI, Sprint, AT&T, Cable and Wireless), national/international coverage
  - treat each other as equals



Tier-1 ISP: e.g., Sprint

Sprint US backbone network

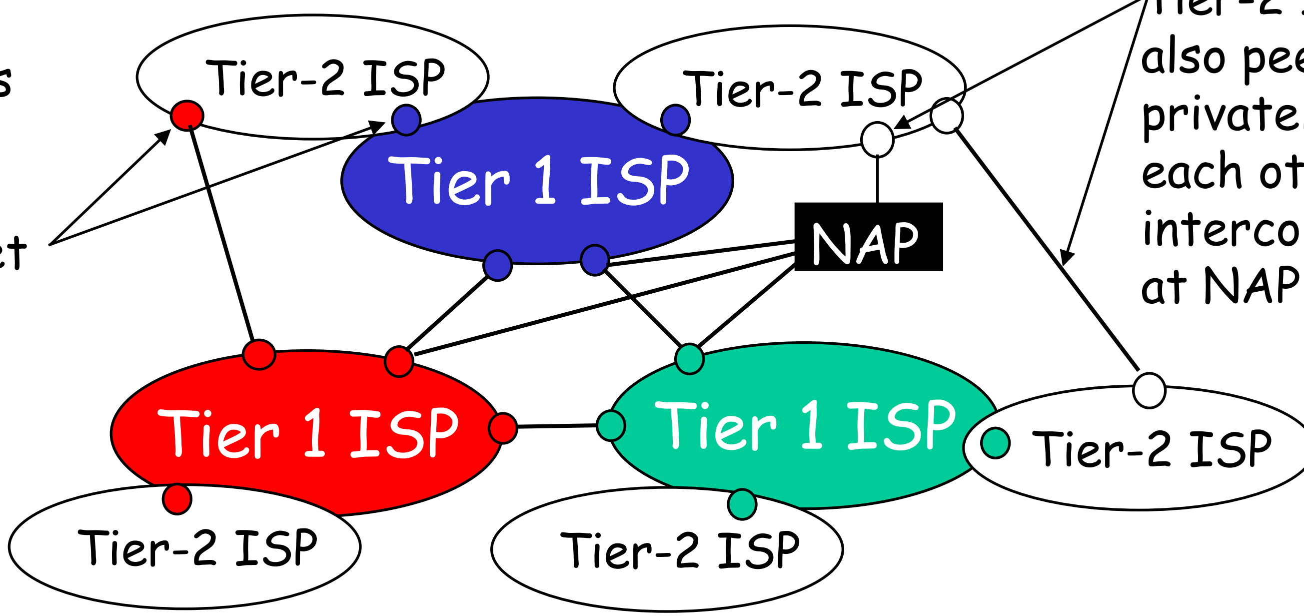


Internet structure: network of networks

- "Tier-2" ISPs: smaller (often regional) ISPs
  - Connect to one or more tier-1 ISPs, possibly other tier-2 ISPs

Tier-2 ISP pays tier-1 ISP for connectivity to rest of Internet

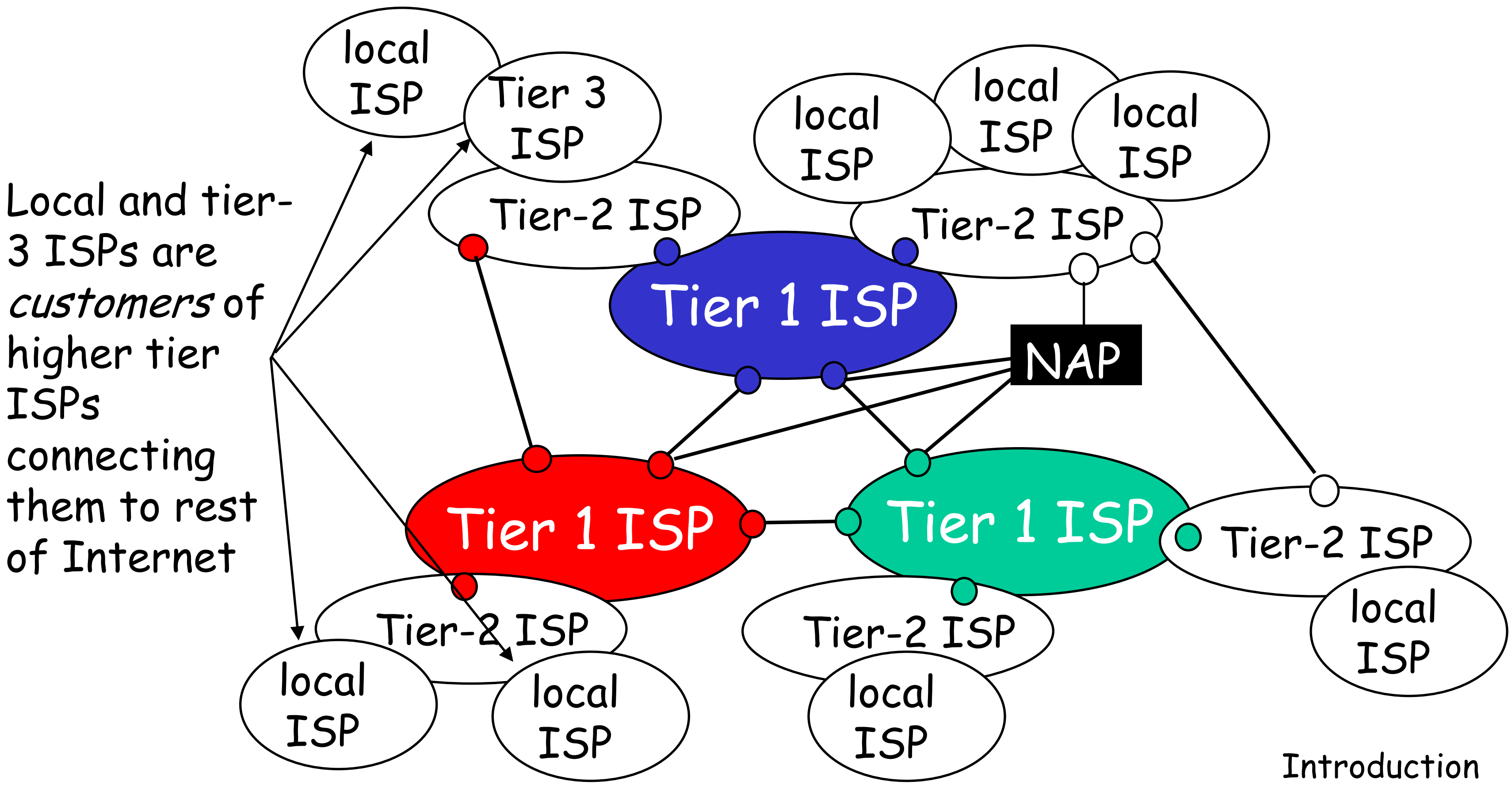
- tier-2 ISP is *customer* of tier-1 provider



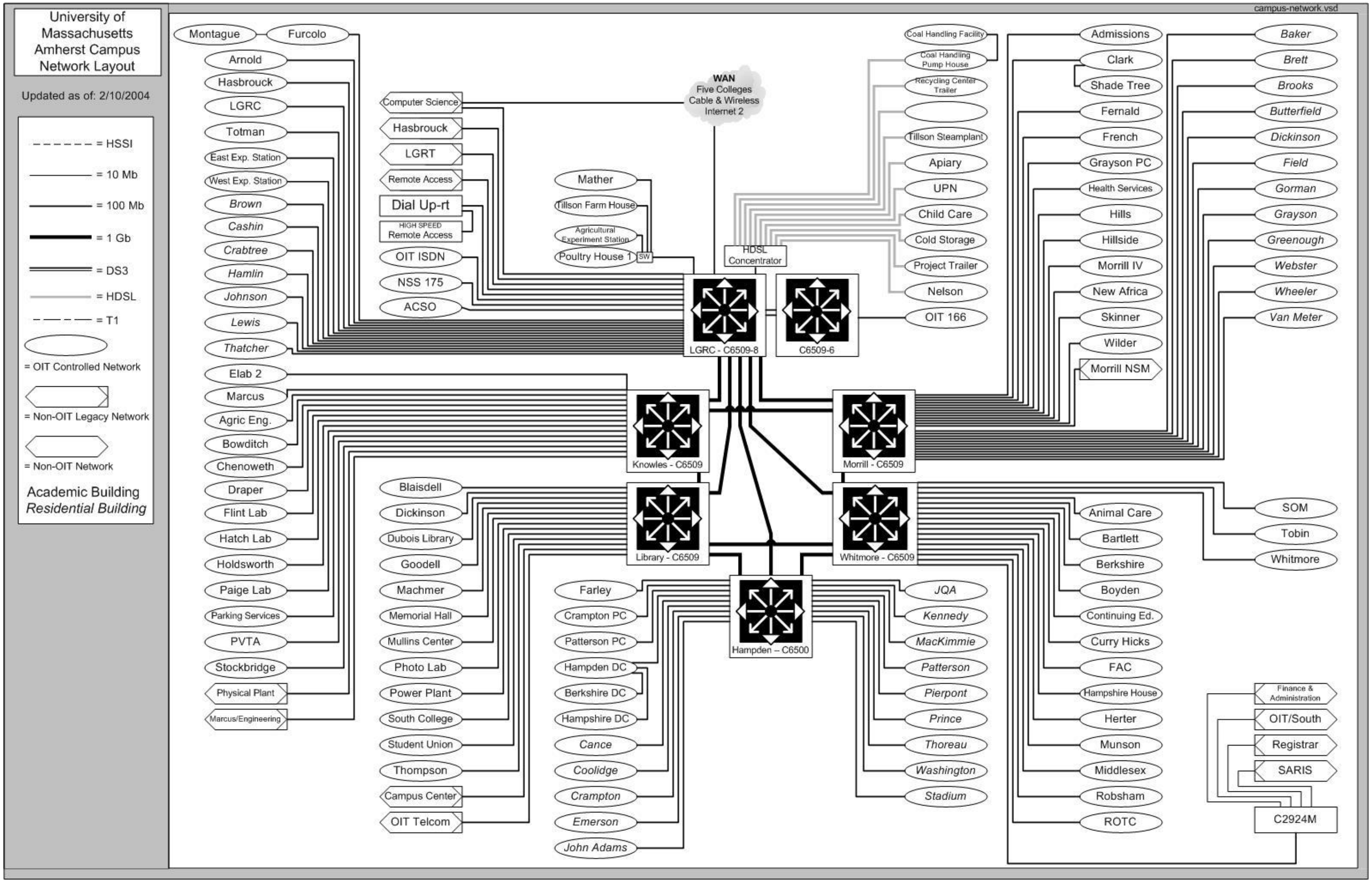
Tier-2 ISPs also peer privately with each other, interconnect at NAP

Internet structure: network of networks

- "Tier-3" ISPs and local ISPs
  - last hop ("access") network (closest to end systems)

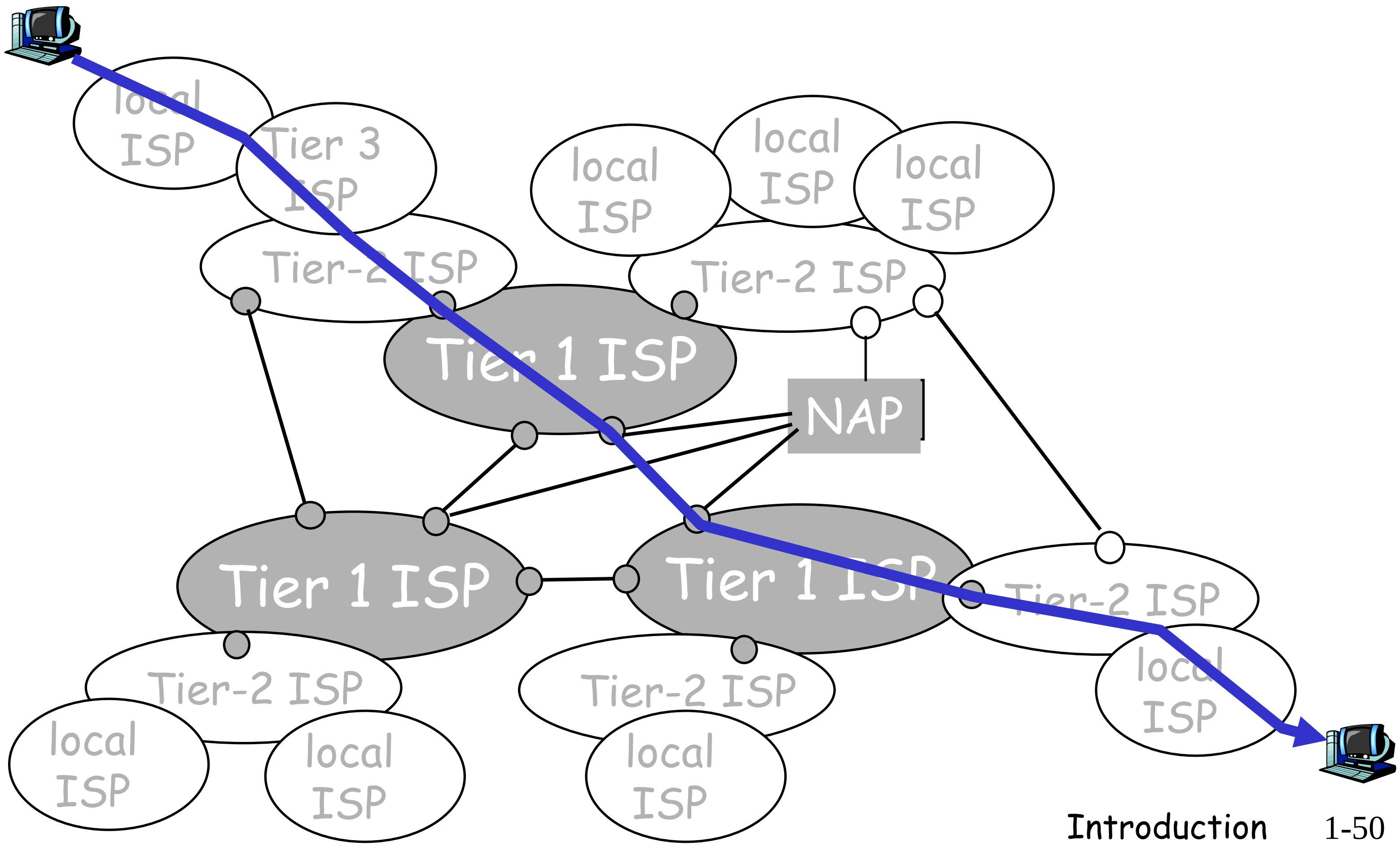


UMass Campus Network



Internet structure: network of networks

- a packet passes through many networks!



Chapter 1: roadmap

1.1 What *is* the Internet?

1.2 Network edge

1.3 Network core

1.4 Network access and physical media

1.5 Internet structure and ISPs

1.6 Delay & loss in packet-switched networks

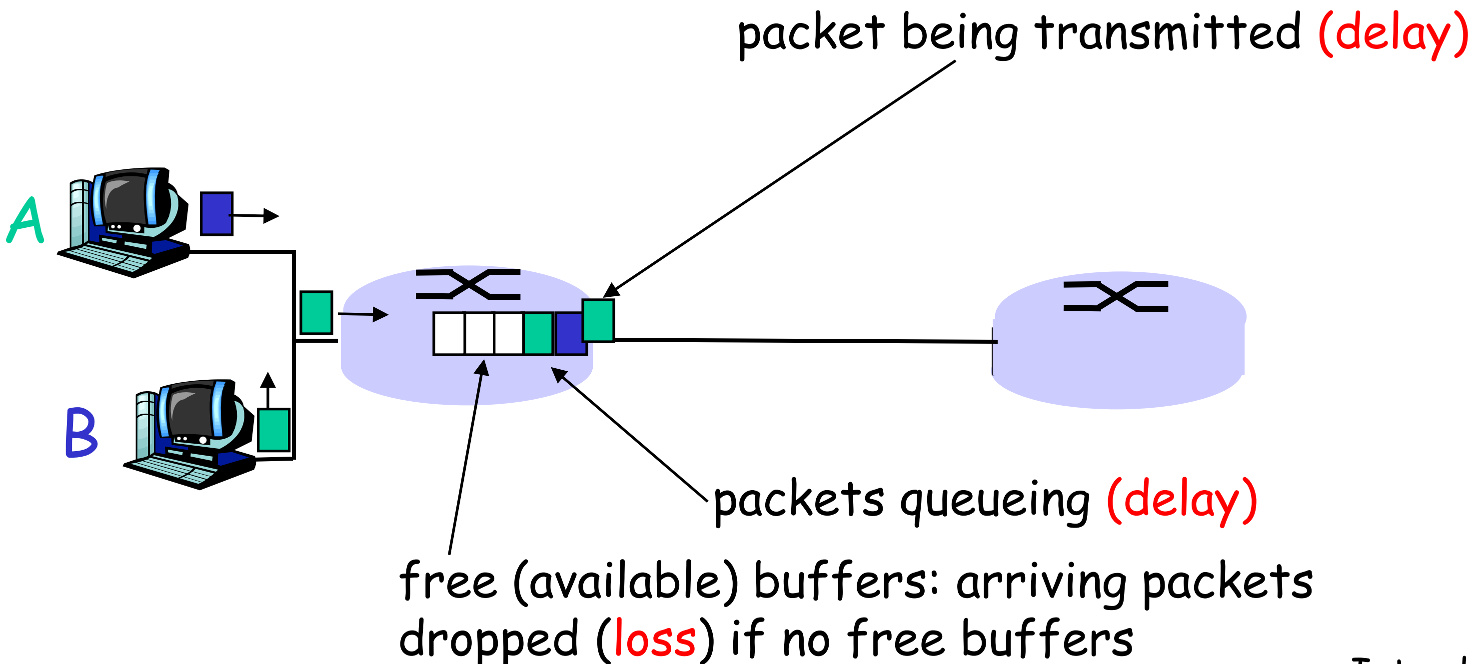
1.7 Protocol layers, service models

1.8 History

How do loss and delay occur?

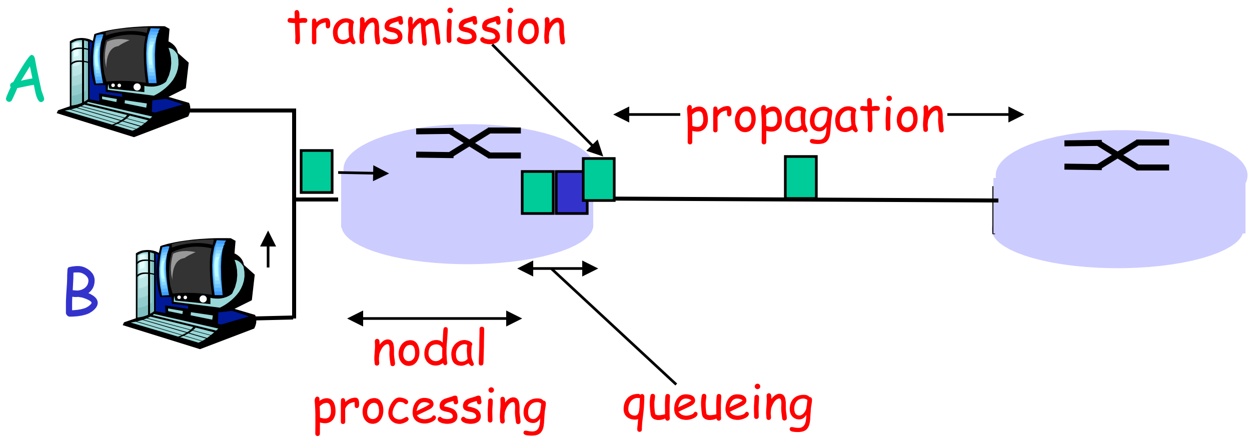
packets *queue* in router buffers

- packet arrival rate to link exceeds output link capacity
- packets queue, wait for turn



Four sources of packet delay

- 1. nodal processing:
  - check bit errors
  - determine output link
- 2. queueing
  - time waiting at output link for transmission
  - depends on congestion level of router



Delay in packet-switched networks

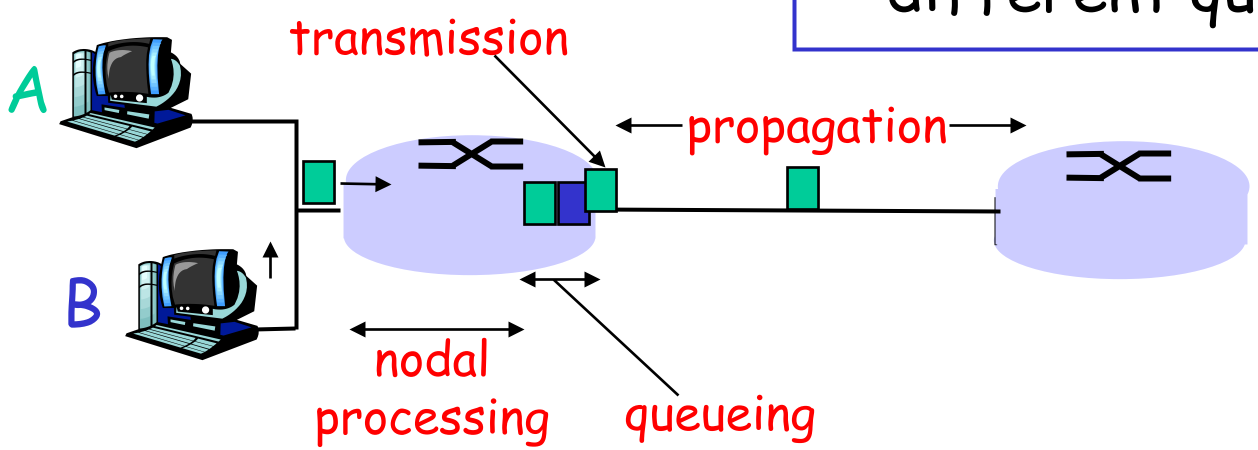
3. Transmission delay:

- R=link bandwidth (bps)
- L=packet length (bits)
- time to send bits into link = L/R

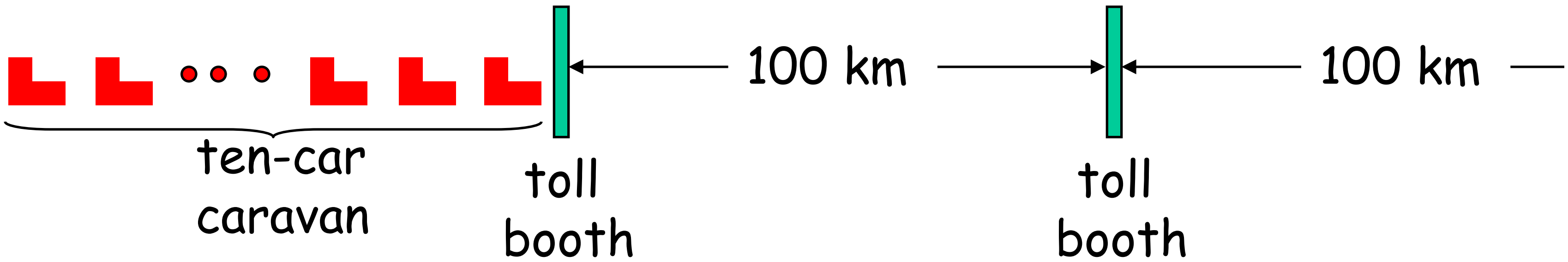
4. Propagation delay:

- d = length of physical link
- s = propagation speed in medium (~2x10^8 m/sec)
- propagation delay = d/s

Note: s and R are *very* different quantities!

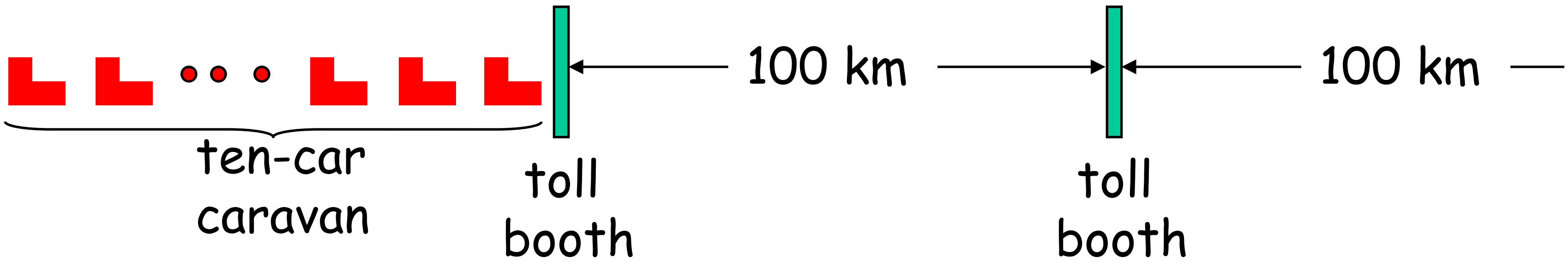


Caravan analogy



- Cars "propagate" at 100 km/hr
- Toll booth takes 12 sec to service a car (transmission time)
- car~bit; caravan ~ packet
- Q: How long until caravan is lined up before 2nd toll booth?
- Time to "push" entire caravan through toll booth onto highway = 12*10 = 120 sec
- Time for last car to propagate from 1st to 2nd toll both: 100km/(100km/hr)= 1 hr
- A: 62 minutes

Caravan analogy (more)



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

- Yes! After 7 min, 1st car at 2nd booth and 3 cars still at 1st booth.
- 1st bit of packet can arrive at 2nd router before packet is fully transmitted at 1st router!
  - See Ethernet applet at AWL Web site

Nodal delay

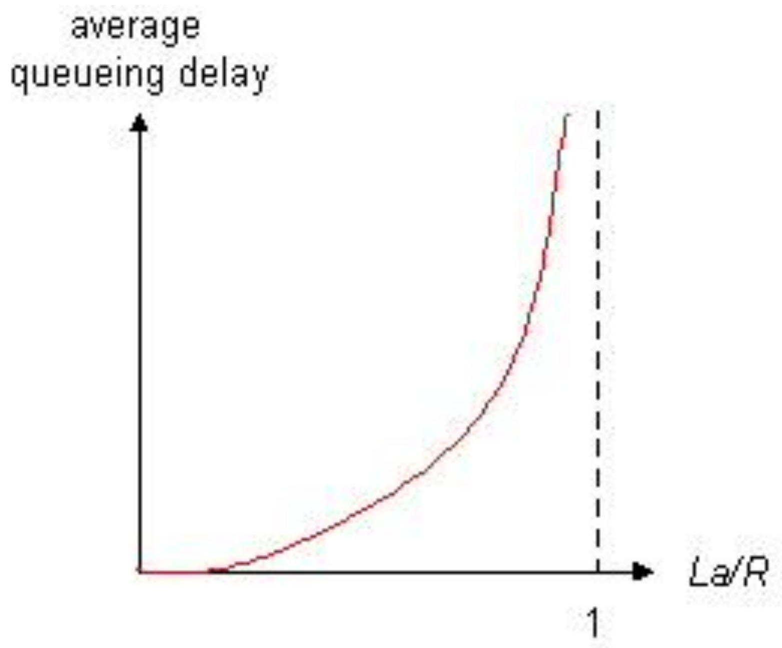
$$\boxed{d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}}$$

- d_{proc} = processing delay
  - typically a few microsecs or less
- d_{queue} = queuing delay
  - depends on congestion
- d_{trans} = transmission delay
  - = L/R, significant for low-speed links
- d_{prop} = propagation delay
  - a few microsecs to hundreds of msecs

Queueing delay (revisited)

- R=link bandwidth (bps)
- L=packet length (bits)
- a=average packet arrival rate

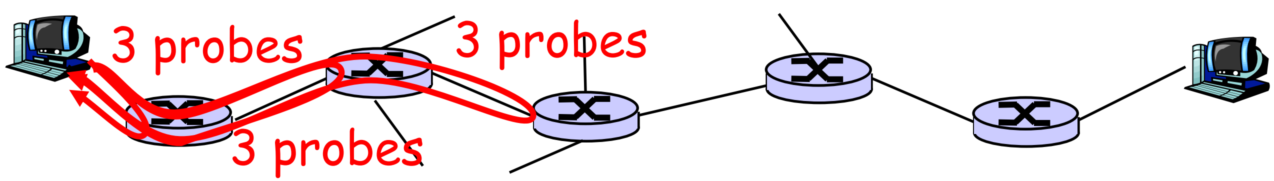
traffic intensity = La/R



- La/R ~ 0: average queueing delay small
- La/R -> 1: delays become large
- La/R > 1: more "work" arriving than can be serviced, average delay infinite!

"Real" Internet delays and routes

- What do "real" Internet delay & loss look like?
- **Traceroute program:** provides delay measurement from source to router along end-end Internet path towards destination. For all *i:*
  - sends three packets that will reach router *i* on path towards destination
  - router *i* will return packets to sender
  - sender times interval between transmission and reply.



"Real" Internet delays and routes

traceroute: gaia.cs.umass.edu to www.eurecom.fr

Three delay measurements from gaia.cs.umass.edu to cs-gw.cs.umass.edu

1 cs-gw (128.119.240.254) 1 ms 1 ms 2 ms
2 border1-rt-fa5-1-0.gw.umass.edu (128.119.3.145) 1 ms 1 ms 2 ms
3 cht-vbns.gw.umass.edu (128.119.3.130) 6 ms 5 ms 5 ms
4 jn1-at1-0-0-19.wor.vbns.net (204.147.132.129) 16 ms 11 ms 13 ms
5 jn1-so7-0-0-0.wae.vbns.net (204.147.136.136) 21 ms 18 ms 18 ms
6 abilene-vbns.abilene.ucaid.edu (198.32.11.9) 22 ms 18 ms 22 ms
7 nycm-wash.abilene.ucaid.edu (198.32.8.46) 22 ms 22 ms 22 ms
8 62.40.103.253 (62.40.103.253) 104 ms 109 ms 106 ms
9 de2-1.de1.de.geant.net (62.40.96.129) 109 ms 102 ms 104 ms
10 de.fr1.fr.geant.net (62.40.96.50) 113 ms 121 ms 114 ms
11 renater-gw.fr1.fr.geant.net (62.40.103.54) 112 ms 114 ms 112 ms
12 nio-n2.cssi.renater.fr (193.51.206.13) 111 ms 114 ms 116 ms
13 nice.cssi.renater.fr (195.220.98.102) 123 ms 125 ms 124 ms
14 r3t2-nice.cssi.renater.fr (195.220.98.110) 126 ms 126 ms 124 ms
15 eurecom-valbonne.r3t2.ft.net (193.48.50.54) 135 ms 128 ms 133 ms
16 194.214.211.25 (194.214.211.25) 126 ms 128 ms 126 ms
17 * * *
18 * * *
19 fantasia.eurecom.fr (193.55.113.142) 132 ms 128 ms 136 ms

trans-oceanic link

* means no response (probe lost, router not replying)

Packet loss

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

Chapter 1: roadmap

1.1 What *is* the Internet?
1.2 Network edge
1.3 Network core
1.4 Network access and physical media
1.5 Internet structure and ISPs
1.6 Delay & loss in packet-switched networks
1.7 Protocol layers, service models
1.8 History

Protocol "Layers"

Networks are complex!

- many "pieces":
  - hosts
  - routers
  - links of various media
  - applications
  - protocols
  - hardware, software

Question:

Is there any hope of *organizing* structure of network?

Or at least our discussion of networks?

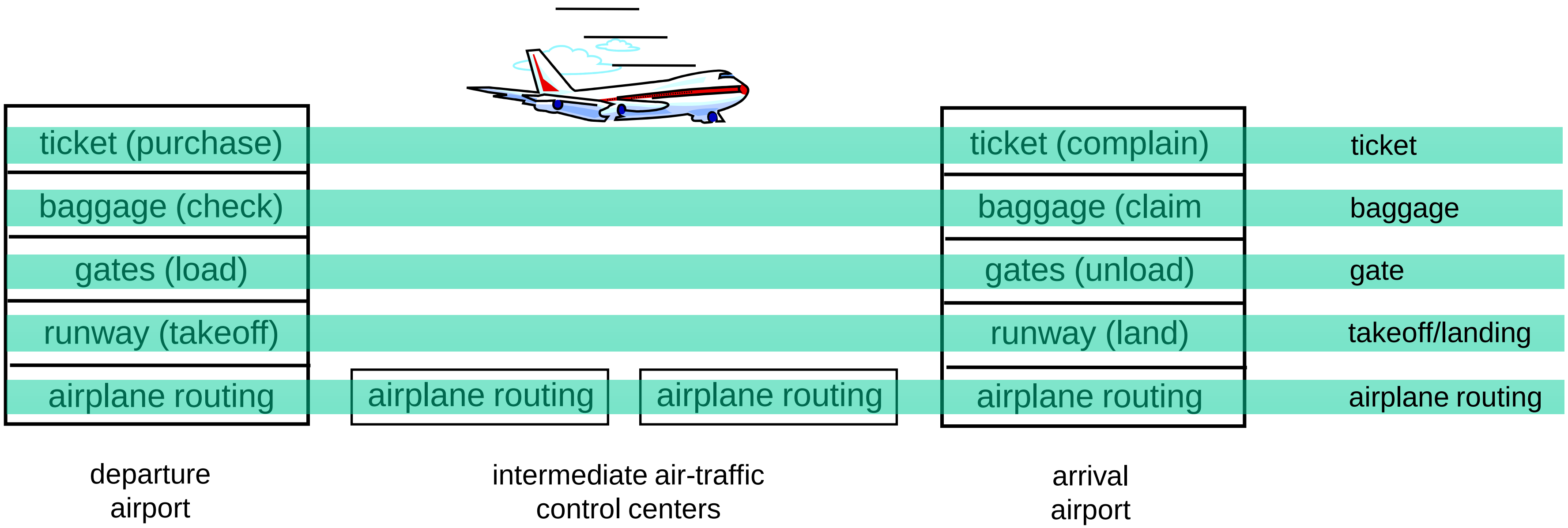
Organization of air travel

ticket (purchase)	ticket (complain)
baggage (check)	baggage (claim)
gates (load)	gates (unload)
runway takeoff	runway landing
airplane routing	airplane routing

airplane routing

- a series of steps

Layering of airline functionality



Layers: each layer implements a service

- via its own internal-layer actions
- relying on services provided by layer below

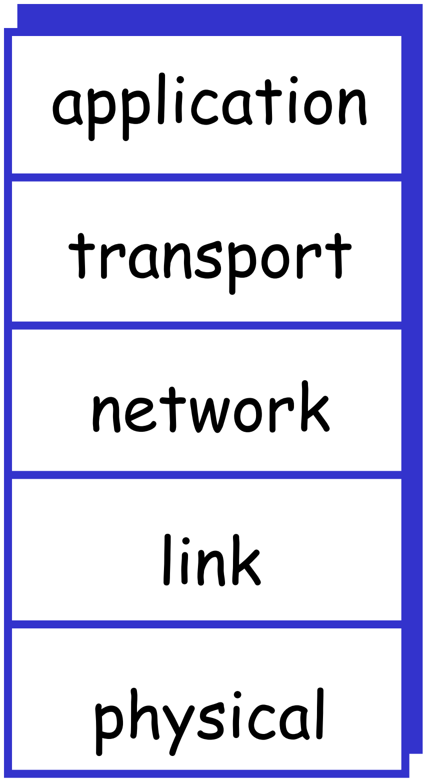
Why layering?

Dealing with complex systems:

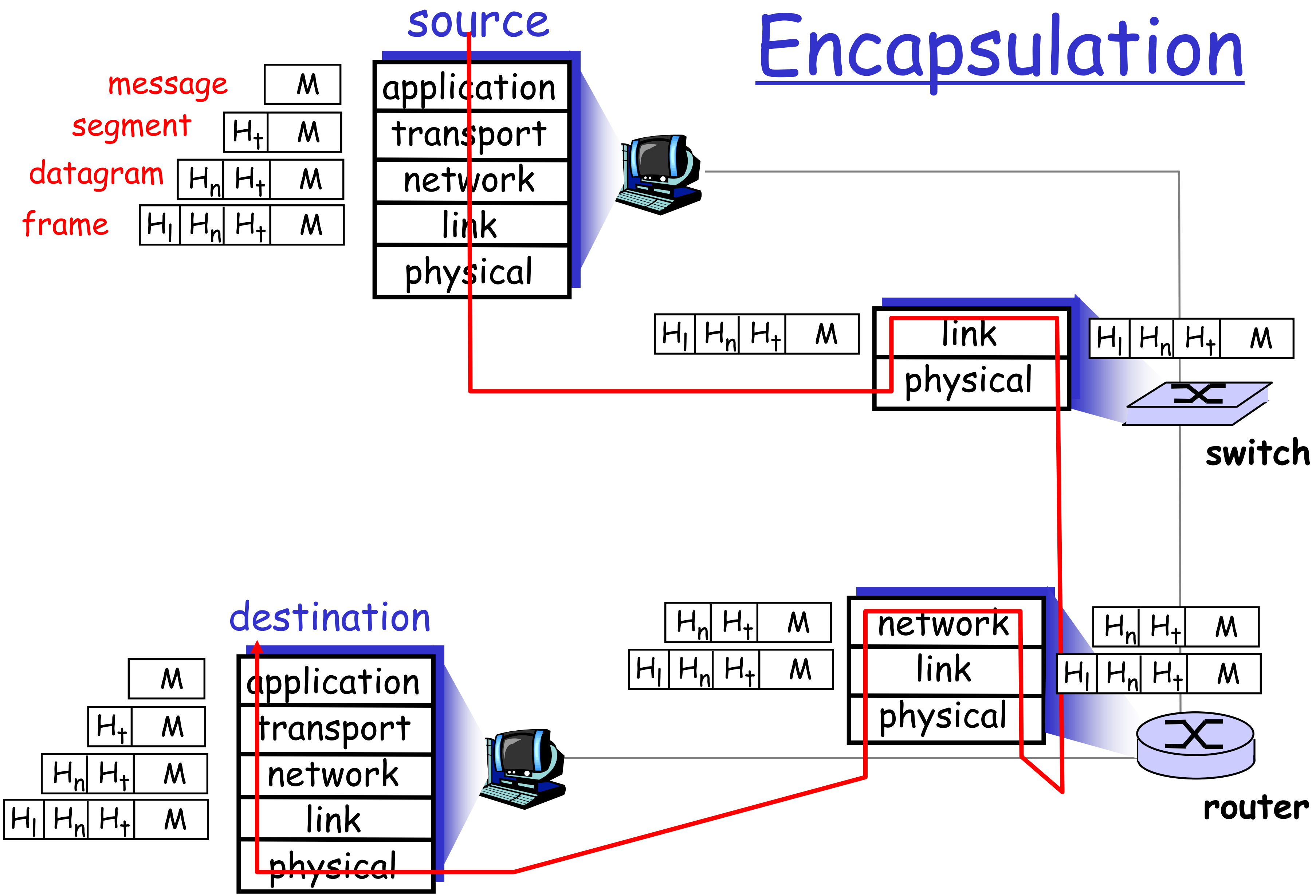
- explicit structure allows identification, relationship of complex system's pieces
  - layered reference model for discussion
- modularization eases maintenance, updating of system
  - change of implementation of layer's service transparent to rest of system
  - e.g., change in gate procedure doesn't affect rest of system
- layering considered harmful?

Internet protocol stack

- application: supporting network applications
  - FTP, SMTP, HTTP
- transport: host-host data transfer
  - TCP, UDP
- network: routing of datagrams from source to destination
  - IP, routing protocols
- link: data transfer between neighboring network elements
  - PPP, Ethernet
- physical: bits "on the wire"



Encapsulation



Chapter 1: roadmap

1.1 What *is* the Internet?

1.2 Network edge

1.3 Network core

1.4 Network access and physical media

1.5 Internet structure and ISPs

1.6 Delay & loss in packet-switched networks

1.7 Protocol layers, service models

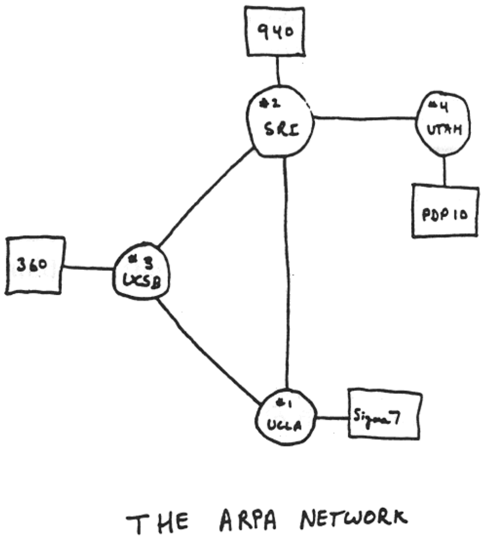
1.8 History

Internet History

1961-1972: Early packet-switching principles

- 1961: Kleinrock - queueing theory shows effectiveness of packet-switching
- 1964: Baran - packet-switching in military nets
- 1967: ARPAnet conceived by Advanced Research Projects Agency
- 1969: first ARPAnet node operational

- 1972:
  - ARPAnet public demonstration
  - NCP (Network Control Protocol) first host-host protocol
  - first e-mail program
  - ARPAnet has 15 nodes



Internet History

1972-1980: Internetworking, new and proprietary nets

- 1970: ALOHAnet satellite network in Hawaii
- 1974: Cerf and Kahn - architecture for interconnecting networks
- 1976: Ethernet at Xerox PARC
- ate70's: proprietary architectures: DECnet, SNA, XNA
- late 70's: switching fixed length packets (ATM precursor)
- 1979: ARPAnet has 200 nodes

Cerf and Kahn's internetworking principles:
- minimalism, autonomy - no internal changes required to interconnect networks
- best effort service model
- stateless routers
- decentralized control

define today's Internet architecture

Internet History

1980-1990: new protocols, a proliferation of networks

- 1983: deployment of TCP/IP
- 1982: smtp e-mail protocol defined
- 1983: DNS defined for name-to-IP-address translation
- 1985: ftp protocol defined
- 1988: TCP congestion control
- new national networks: Csnet, BITnet, NSFnet, Minitel
- 100,000 hosts connected to confederation of networks

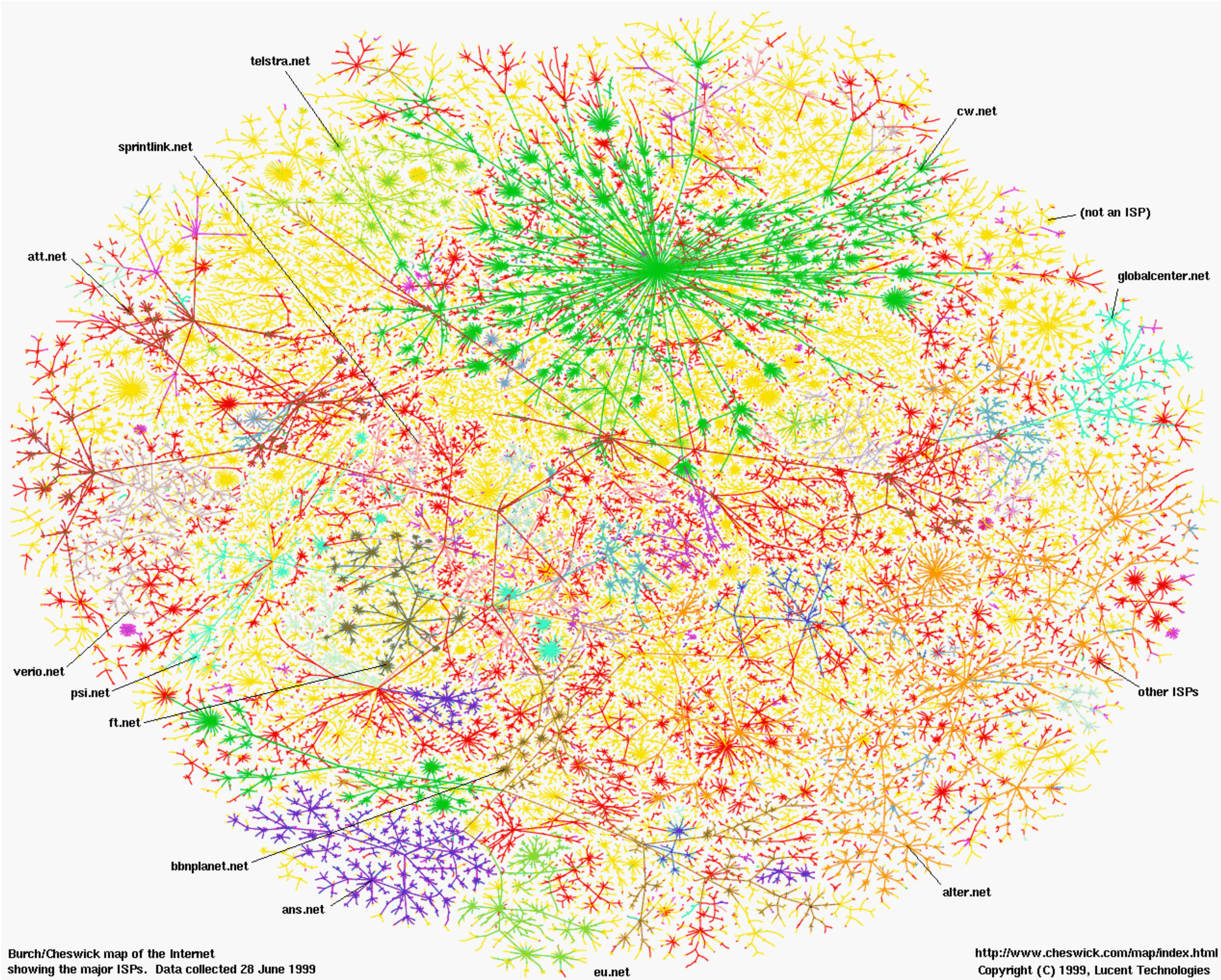
Internet History

1990, 2000's: commercialization, the Web, new apps

- Early 1990's: ARPAnet decommissioned
- 1991: NSF lifts restrictions on commercial use of NSFnet (decommissioned, 1995)
- early 1990s: Web
  - hypertext [Bush 1945, Nelson 1960's]
  - HTML, HTTP: Berners-Lee
  - 1994: Mosaic, later Netscape
  - late 1990's: commercialization of the Web

Late 1990's – 2000's:

- more killer apps: instant messaging, P2P file sharing
- network security to forefront
- est. 50 million host, 100 million+ users
- backbone links running at Gbps


telstra.net
sprintlink.net
att.net
cw.net
(not an ISP)
globalcenter.net
verio.net
psi.net
ft.net
other ISPs
bbnplanet.net
ans.net
alter.net
eu.net
Burch/Cheswick map of the Internet
showing the major ISPs. Data collected 28 June 1999
http://www.cheswick.com/map/index.html
Copyright (C) 1999, Lucent Technologies

Introduction: Summary

Covered a "ton" of material!

- Internet overview
- what's a protocol?
- network edge, core, access network
  - packet-switching versus circuit-switching
- Internet/ISP structure
- performance: loss, delay
- layering and service models
- history

You now have:

- context, overview, "feel" of networking
- more depth, detail *to follow!*

DATA LINK LAYER

Sunghyun Choi

Adopted from Prof. Saewoong Bahk's material

Data Link Layer: Introduction

- Efficient communication between two <u>adjacent</u> machines
- Providing a well defined service interface to the network layer
- Framing
- Error control (providing error-free channel)
- Flow control
- Medium access control (MAC) for broadcast channel

Services for the Network Layer

- Unacknowledged connectionless service
  - No connection establishment
  - No attempt to recover a frame
  - Appropriate for low error-rate links
- Acknowledged connectionless service
  - No connection establishment
  - Each frame sent is individually acknowledged
  - If ACK has not arrived within a specified time interval, it can be sent again

- Useful for unreliable channels

- Acknowledged connection-oriented service
  - Connection establishment
  - Each frame over the connection is numbered and acknowledged
  - Frame order conservation

Error detection and correction

- Extra bits must be appended to a frame to detect errors (redundancy)
- Single Parity Check
  - simplest
  - a single bit , “parity check” appended
  - This parity check bit is 1 if the number of 1’s in the bit string is odd, and 0 otherwise
  - single error detection (odd number of errors can be detected)

- Horizontal and vertical parity checks
  - 2-dimensional parity check (row and column)
  - If an even number of errors are confined to a single row, each of them can be detected by column parity checks
- Multi-parity checks
  - Let K be the length of the data string
  - Let L be the number of parity checks

- Each string of length K is mapped into a frame of length K+L
- Minimum distance of a code : the smallest number of bit positions in which two codewords differ
  - The smallest # of errors that can convert one code word into another
  - To detect d errors, need a distance d+1 code
  - To correct d errors, need a distance 2d+1 code

- Effectiveness of parity check codes
- Measures of effectiveness
  - Min. distance
  - Burst detecting capability
  - Probability that a completely random string is accepted as error free

- Length of a burst of errors: # of bits from the first error to the last, inclusive
- Burst detecting capability: the largest integer B s.t. the code can detect all burst of length B or less
- Completely random string of length K+L: each such string is received with probability $2^{-(K+L)}$, i.e., equal probability
  - Probability → $2^K / 2^{(K+L)} = 2^{-L}$
  - Error probability -> $2^{-L} * (2^K – 1) / 2^K$

- Cyclic Redundancy Check (CRC)
  - Most widely used for error detection
  - A k-bit string is regarded as the coefficient list for a polynomial with k terms ($D^{k-1} \ldots D^0$)

    (ex. 110001 : $D^5 + D^4 + D^0$)
  - Modulo 2 arithmetic (exclusive OR)
    - "addition" = "subtraction"

- Properties of N-bit cyclic codes
  - Linear: the sum of any two code words is another codeword
  - Cyclic: any cyclic shift of a code word is another codeword
    - If $X(D)$ is a code word in a cyclic code, then $D*X(D)$ modulo D^N-1 is also a code word.

- g(D): generator polynomial
  - Lowest degree monic (i.e., with highest coefficient = 1) polynomial that is a code word in a cyclic code
  - deg(g(D)) = L
- All code words in a cyclic code have the form a(D)*g(D)
  - deg(a(D)) < N-L = K (= # of data bits)

- Proof)
  - Assume a code word X(D)=a(D)*g(D) + r(D), then r(D) = X(D) + a(D)*g(D) is also a code word due to the linearity.
  - Contradiction because g(D) is the lowest degree code word.
- X(D) is a code word iff there exists a(D) of degree at most N-L-1 (=K-1) satisfying X(D)=a(D)*g(D)

- Data string: $s(D) = s_{K-1}D^{K-1}+...+1$
- generator polynomial:

  $g(D) = D^L + g_{L-1}D^{L-1} + ...+ 1$
- CRC polynomial:

  $c(D) = \text{Remainder}\ [\ s(D)D^L / g(D)\ ]$
- Let z(D) be the quotient, then

  $s(D)D^L = g(D)z(D) + c(D)$
- Frame to be transmitted

  $x(D)=g(D)z(D)=s(D)\ D^L - c(D)= s(D)\ D^L + c(D)$

– Let the received frame y(D)=x(D)+e(D), then e(D) is undetectable (e(D) is nonzero)

$$\updownarrow$$

e(D) = g(D)f(D)   for some nonzero polynomial f(D)

– ex) frame 1101011011, generator 10011

quotient 1100001010, remainder 1110

transmitted frame 1101011011 1110

- Effectiveness
  - Single errors: $e(D) = D^i \neq g(D)f(D)$ since $g(D)$ has at least two non-zero coefficients
  - Double errors: $e(D) = D^i + D^j = D^j (D^{i-j}-1)$ $(i>j)$
    - $D^{i-j}-1 / g(D) = D^n-1 / g(D)$
    - Smallest n s.t. D^n-1 is divisible by $g(D)$ cannot be larger than 2^L-1
    - If $g(D)$ is a primitive polynomial, then the smallest n $= 2^L-1$
    - If code word length $N \leq 2^L-1$, then all double errors can be detected

- Practically, $g(D) = (D+1)$(primitive polynomial of degree L-1)
  - All odd #'s of errors can be detected
- Min. distance ≥ 4

- Burst detecting capability = L
  - For $e(D) = g(D)\, z(D)$, $\deg(e(D)) \geq L$ since $\deg(g(D)) = L$
- Error probability
  - $(2^K - 1) / 2^{K+L}$

ARQ

- Retransmission strategy
- Assumptions
  - (a1) receiving DLC knows when frames start and when they end
  - (a2) all frames containing transmission errors are detected (← most unrealistic)
  - (a3) each transmitted frame is delayed by an arbitrary and variable time before arriving at the receiver and some frames may be lost and never received
  - (a4) those frames that arrive in the same order as transmitted

ARQ : Automatic Repeat reQuest

- Stop and Wait ARQ
  - sender sends one frame, then waits for ACK before proceeding
  - lost? error? timeout? duplicated?
- Simplex protocol for a noisy channel
  - stop and wait protocol with a timer
  - Receiver : Instead of using ACK, return the num. of the next packet awaited (may use piggybacking)
  - Inefficient use of communication links due to long waiting time

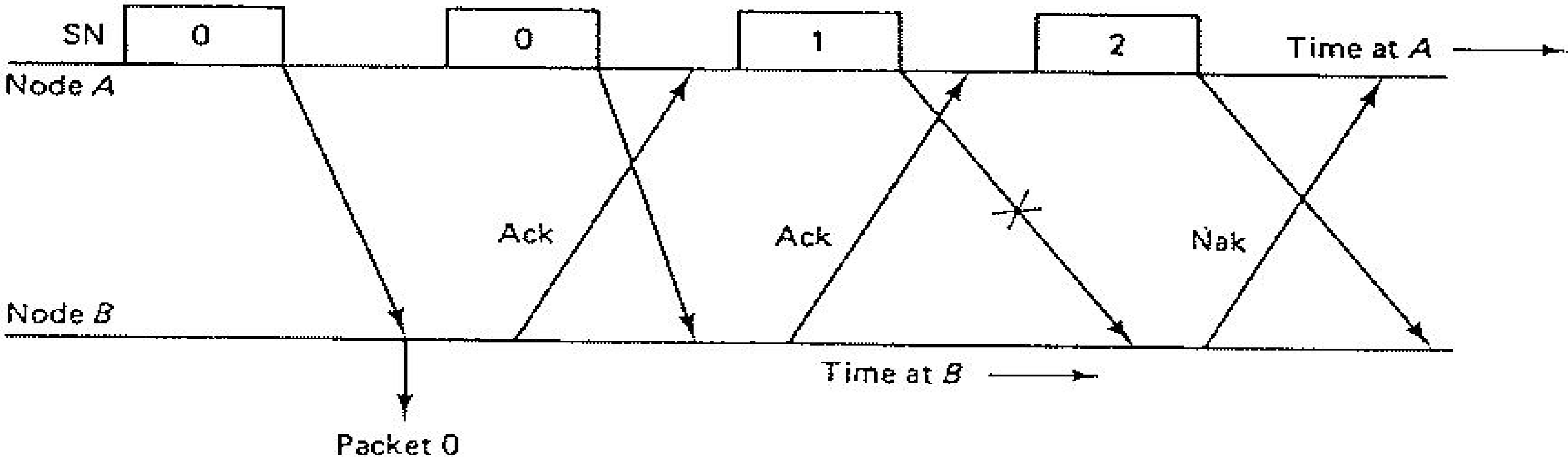


Figure 2.19 The trouble with unnumbered acks. If the transmitter at A times-out and sends packet 0 twice, node B can use the sequence numbers to recognize that packet 0 is being repeated. It must send an ack for both copies, however, and (since acks can be lost) the transmitter cannot tell whether the second ack is for packet 0 or 1.

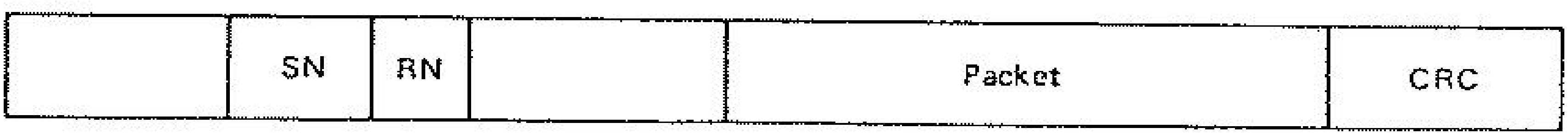


Figure 2.20 The header of a frame contains a field carrying the sequence number, SN, of the packet being transmitted. If piggybacking is being used, it also contains a field carrying the request number, RN, of the next packet awaited in the opposite direction.

- Algorithm at node A (transmitter)
  1. Set the integer variables SN to 0
  2. Accept a packet from the higher layer at A; if no packet is available, wait until it is; assign number SN to the new packet
  3. Transmit the SNth packet in a frame containing SN in the sequence number field.

4. If an error-free frame is received from B containing a request number RN greater than SN, increase SN to RN and go to step 2. If no such frame is received within some finite delay, go to step 3.

- Algorithm at node B (receiver)

1. Set the integer variable RN to 0 and repeat steps 2 and 3 forever

2. Whenever an error free frame is received from node A containing seq. num. SN equal to RN, release the received packet to the higher layer and increment RN.
3. An arbitrary time (within bounded delay) after receiving any error free frame from A, transmit a frame containing RN to node A.

- Correctness of ARQ
  - Safety: an algorithm is safe if it never produces an incorrect result; that is, it never releases a packet out of the correct order to the higher layer at the receiver end.
  - Liveness: an algorithm is live if it can continue to forever produce results, i.e., if it never enter a deadlock condition.

- Correctness of stop-and-wait ARQ
- Assumptions
  - All error frames are detected by CRC
  - Each frame is received error free with probability at least $q > 0$
  - The link is initially empty, i.e., SN = RN = 0

- Safety
  - SN = 0; RN = 0 → 1; SN = 0 → 1
- Liveness
  - At t1, node A first transmits packet i
  - At t2, this packet is received error-free and released to the higher layer at node B
  - At t3, SN at node A is increased to i+1
  - Demonstrate liveness by showing that t1 < t2 < t3, and t3 is finite
    - t3 is finite since q > 0

- Binary SN & RN suffice for stop-and-wait ARQ
  - Since infinite number of retransmission is allowed
  - SN & RN is determined using modulo 2 operation
  - ARQ operation can be described by a state diagram with four states, i.e., (SN,RN)=(0,0), (0,1), (1,1), (1,0)

Sliding Window Protocols : Go back n ARQ

- Go back n ARQ
  - Most widely accepted ARQ protocol such as for HDLC, SDLC, ADCCP, LAPB
  - Several successive packets (up to n packets) are sent without waiting for the next packet to be requested
  - SN_{min} : the smallest numbered packet that has not been ACKed
  - SN_{max} : the number of the next packet to be accepted from the higher layer

- Algorithm at node A (transmitter)
  1. Set the integer variables SN_{min} and SN_{max} to 0
  2. Do steps 3, 4, and 5 repeatedly in any order
  3. If $SN_{max} < SN_{min} + n$, and a packet is available from the higher layer, (1) accept a new packet into the DLC, (2) assign number SN_{max} to it, and (3) increment SN_{max}

4. If an error free frame received from B contains a request number RN greater than SN_{min}, increase SN_{min} to RN
5. If $SN_{min} < SN_{max}$ and no frame is currently in transmission, choose some number SN, $SN_{min} \le SN < SN_{max}$; transmit the SNth packet with a bounded delay between successive transmission of packet SN_{min} when SN_{min} does not change

- Algorithm at node B (receiver)
  1. Set the integer variable RN to 0 and repeat steps 2 and 3 forever
  2. Whenever an error free frame is received from node A containing seq. num. SN equal to RN, release the received packet to the higher layer and increment RN.
  3. An arbitrary time (within bounded delay) after receiving any error free frame from A, transmit a frame containing RN to node A.
- Go back n with modulus m > n
  – Assuming frames do not get out of order on the links

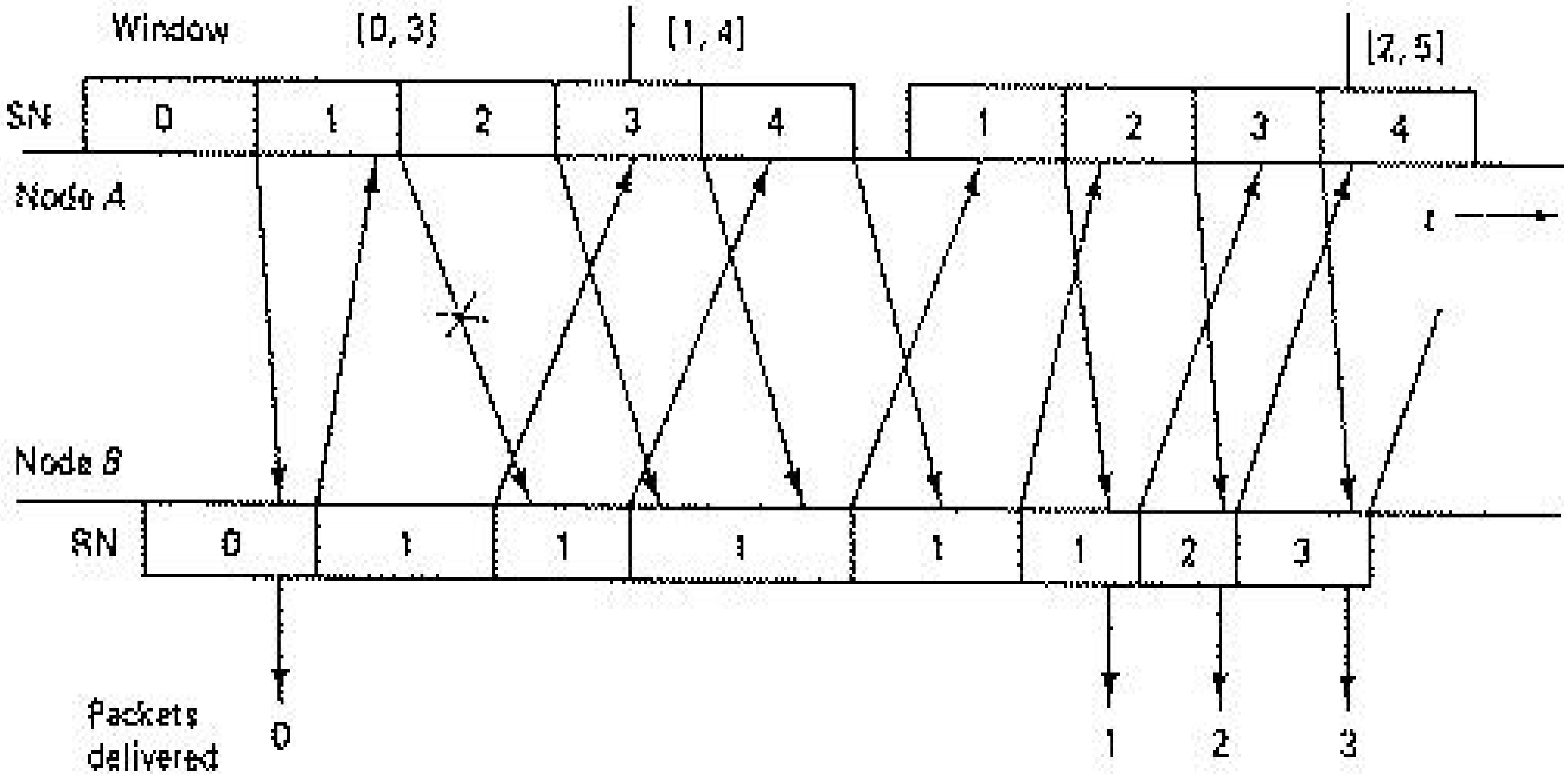


Figure 2.25 Effect of a transmission error on go back 4. Packet 1 is received in error at B, and node B continues to request packet 1 in each reverse frame until node A transmits its entire window, times-out, and goes back to packet 1.

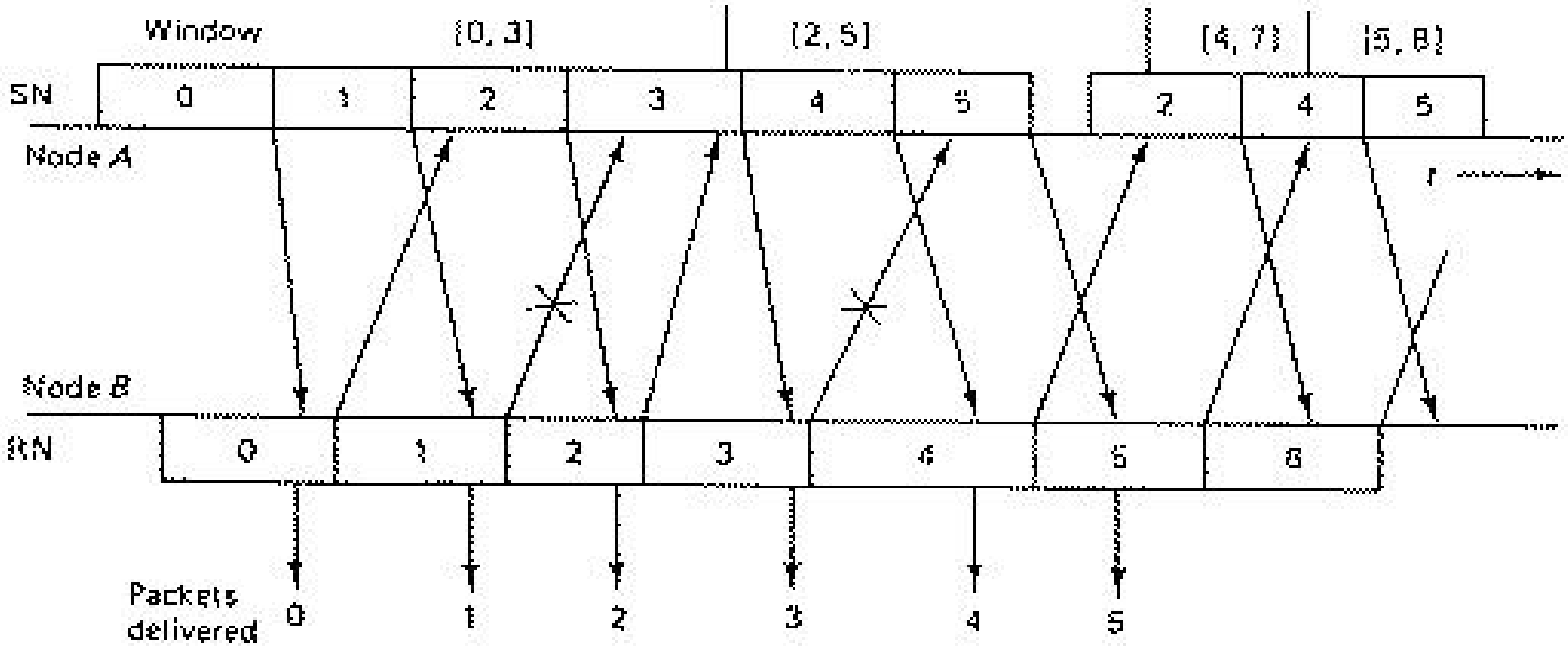


Figure 2.26 Effect of transmission errors in the reverse direction for go back 4. The first error frame, carrying $RN = 1$, causes no problem since it is followed by an error-free frame carrying $RN = 2$ and this frame reaches A before packet number 3, (*i.e.*, the last packet in the current window at A) has completed transmission and before a time-out occurs. The second error frame, carrying $RN = 3$, causes retransmissions since the following reverse frame is delayed until after node A sends its entire window and times-out. This causes A to go back and retransmit packet 2.

- Efficiency of go back n implementations
  - Retransmissions or delays waiting for time-outs due to
    - Errors in either forward or reverse direction
    - Longer frames in the feedback than in the forward direction
  - When frame lengths are exponentially distributed, probability p that a frame is not acked by the time the window is exhausted:
    - $p = (1/2)^n$ (from Prob. 2.23)

Sliding Window Protocols : Selective Repeat ARQ

- The receiver store all the correct frames following the bad one, and request retransmissions from A only for incorrectly received frames

  cf) Go back n : the receiver refuses to accept any frame except the awaiting one

- Selective Repeat with modulus $m \geq 2n$
  - Assuming frames do not get out of order on the links
  - Node A generates a packet with SN at t1, and node B receives it at t2
    - $SN_{min}(t1) \leq SN \leq SN_{min}(t1) + n - 1$
    - $SN_{min}(t1) \leq RN(t2) \leq SN_{min}(t1) + n$
    - $RN(t2) - n \leq SN \leq RN(t2) + n - 1$
    - Accordingly, $m \geq 2n$ since node B has to distinguish values of SN in the entire range

- Throughput: for frame error probability p, expected number of packets delivered to B per frame from A to B is bounded by
  - Th $\leq$ 1 – p: achieved with ideal selective repeat
  - ARQ attempts to achieve this bound
- For go back n, throughput becomes (Prob. 2.26)
  - Th $\leq$ (1-p) / (1+p*beta)
  - Beta: expected number of frames in a round-trip delay interval

- Window and storage issues
  - Transmitter need to idle upon n/beta time erroneous transmissions of the head-of-the-window frame
  - For small n, such idle chance becomes higher
  - For large n, the storage for receiver becomes an issue

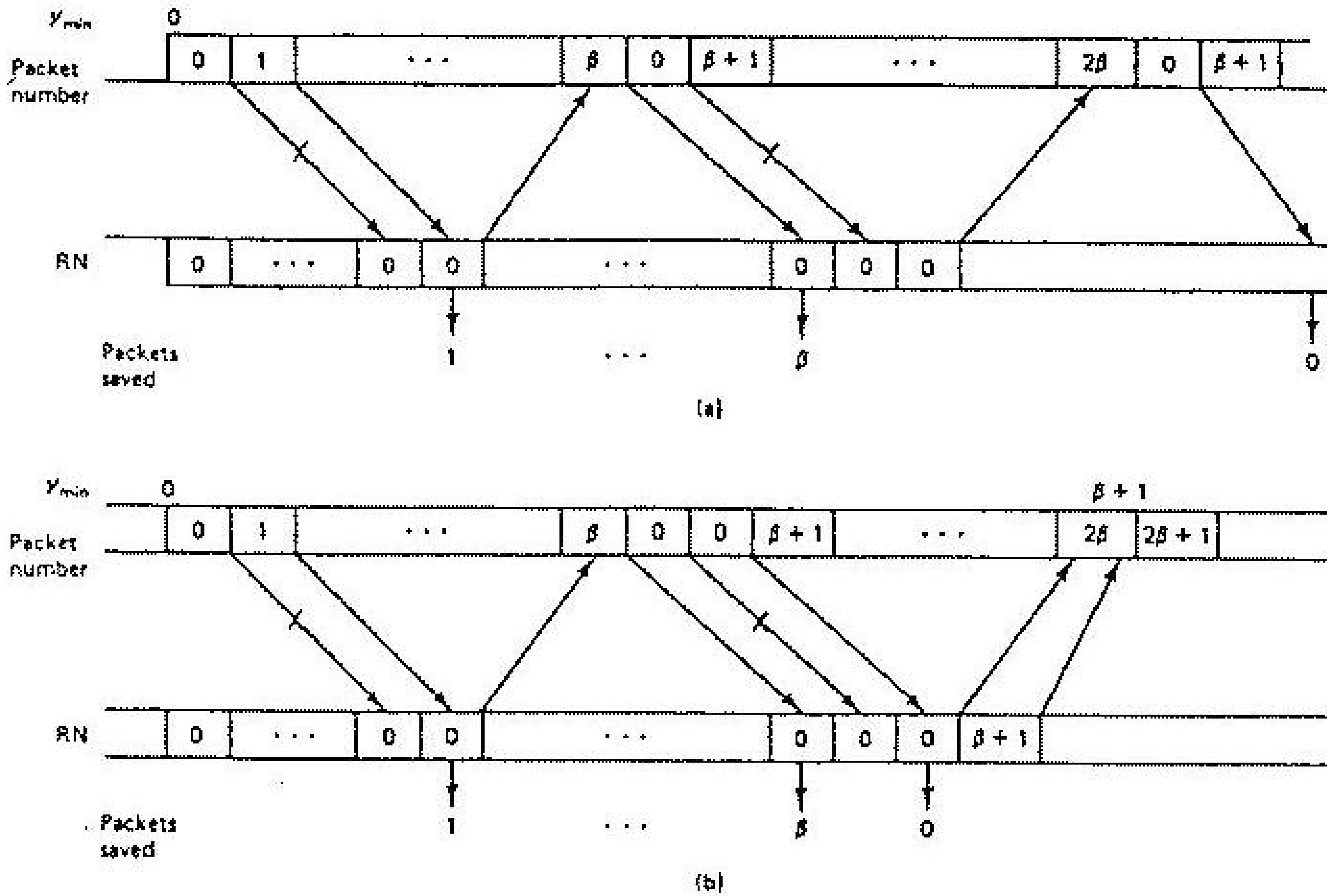


Figure 2.32 Selective repeat ARQ with $n = 2\beta + 2$ and receiver storage for $\beta + 1$ packets. (a) Note the wasted transmissions if a given packet (0) is transmitted twice with errors. (b) Note that this problem is cured, at the cost of one extra frame, if the second transmission of packet 0 is doubled. Feedback contains not only RN but additional information on accepted packets.

Framing

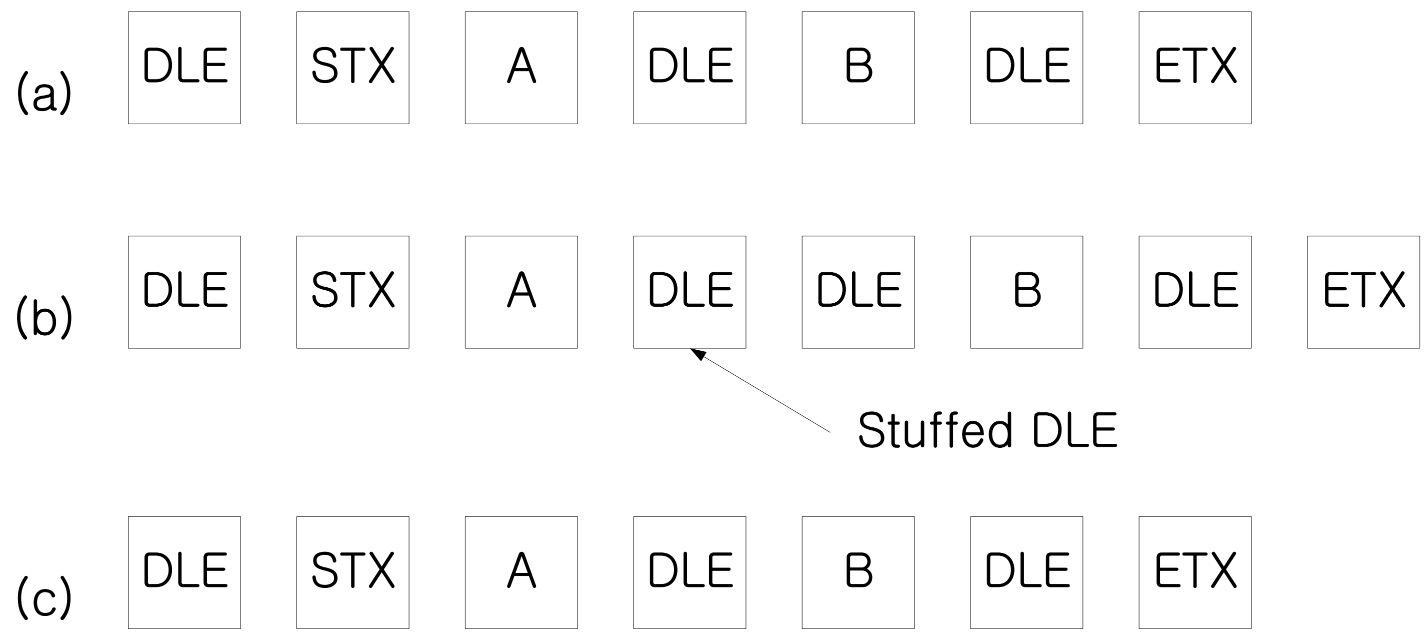
- The data link layer breaks the bit stream up into discrete frames
- How to mark the start and end of each frame?

1. Starting and ending characters with character stuffing
   - Each frame starts with DLE STX and ends with DLE ETX (ASCII character)

Data Link Escape, Start of TeXt, End of TeXt

- Character Stuffing : inserting DLE before each accidental DLE in the data
- Closely tied to 8-bit character system

2. Starting and ending flags with bit stuffing
   - Data frame can have an arbitrary number of bits
   - flag byte : 01111110
   - bit stuffing : inserting a “0” after 5 consecutive ones in the data



(a) Data sent by the network layer.
(b) Data after being character stuffed by the data link layer.
(c) Data passed to the network layer on the receiving side.

(a) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

(b) 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0 0 1 1 1 1 1 1 0

Stuffed bits

(c) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

Bit stuffing.

(a) The original data.
(b) The data as they appear on the line.
(c) The data as they are stored in the receiver's memory after destuffing.

- Approximate computation of the overhead for bit oriented framing
- Assumptions
  - A frame consists of i.i.d. binary r.v.'s with equal probability
  - Terminating signal for a frame is 01^j

– For data of length K, the overhead becomes

$E(OV) = 2^{-(j-1)}+(E(K)-j+1)2^{-j}+j+1$

$= (E(K)-j+3)2^{-j}+j+1 = f(j)$

– Opt. j → j* s.t. $f(j^*) < f(j^*+1)$

– $j^* = [\log_2 E(K)]$, where [.] is the floor operation

– For this optimal value,

$E(OV) \leq \log_2 E(K) + 2$ (Prob. 2.34)

3. Frame Length

- Length represented by ordinary binary numbers → [$\log_2 K_{max}$]+1 bits needed in the length field
- Using source coding, the length field could be as short as the entropy of the length distribution → $H = -1 * \text{Sum\_K}\ P(K) \log_2 P(K)$
- For uniform distribution, $H = \log_2 K_{max}$
- For geometric distribution, $H = \log_2 E(K) + \log_2 e$ ← max. entropy given E(K)

- Framing with errors
  - Data → flag or flag → data due to error
  - To handle data sensitivity problem of DLC
  - Extra CRC for header
  - Fixed frame size (with possible long delay)

Pipelining

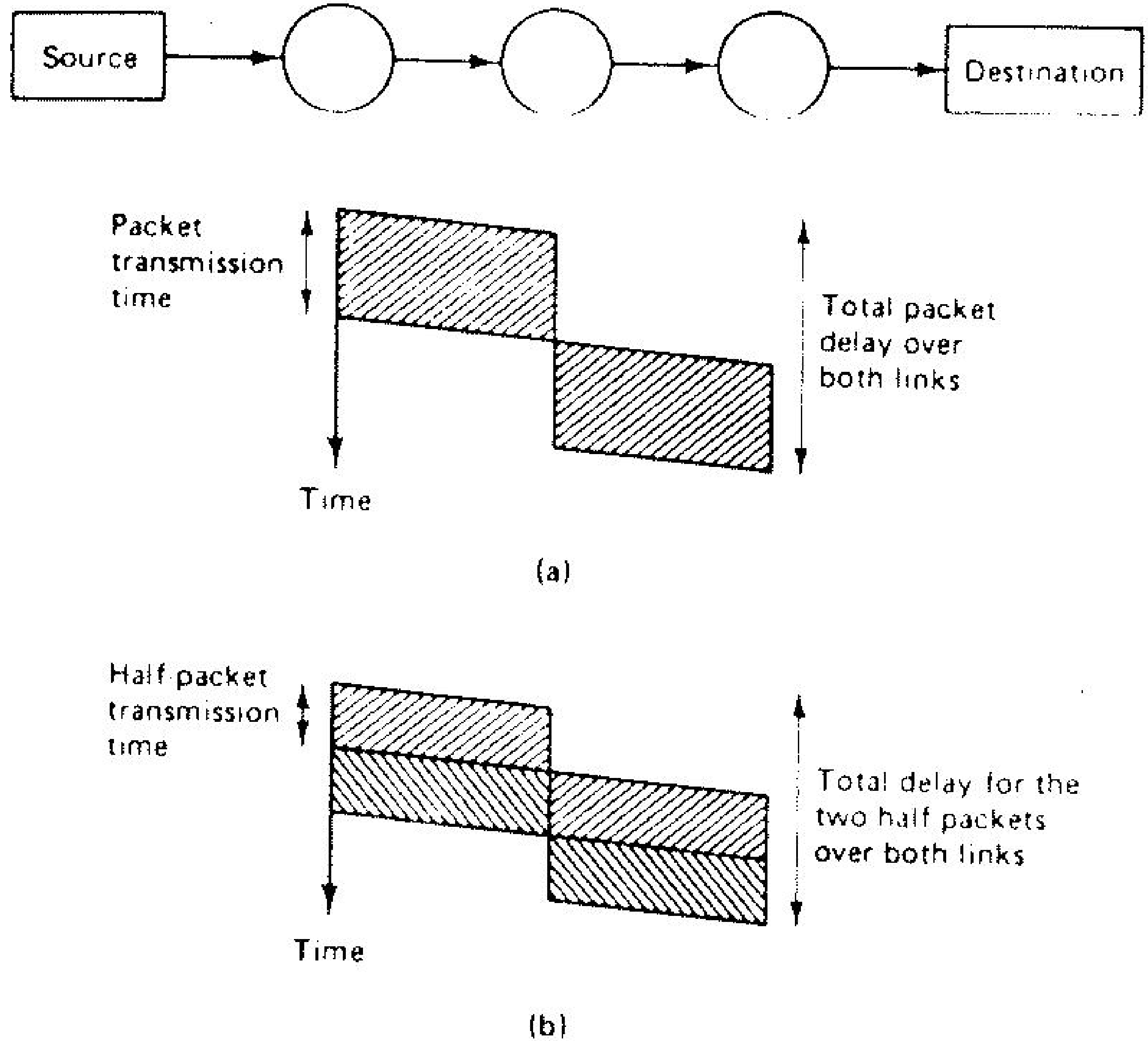
- Maximum Frame Size
  - V : overhead bits per frame
  - K_{max} : allowed max length of a frame
  - M : length of the message to be transmitted
  - The number of packets to be transmitted is $\lceil M / K_{max} \rceil$
  - The total number of bit to be transmitted is then

$$M + \lceil M / K_{max} \rceil V$$

- Pipelining
  - T : total time to transmit the message to the destination
  - C : link capacity [bits/sec]
  - j : total number of links to reach the destination

$$TC = (K_{\max} + V)(j-1) + M + \lceil M / K_{\max} \rceil V$$

  - Assumption : no error, no queueing and propagation delay, equal capacity of links


Source
Destination
Packet transmission time
Total packet delay over both links
Time
(a)
Half packet transmission time
Total delay for the two half packets over both links
Time
(b)

- Let $E\{\lceil M/K_{\max}\rceil\} = E\{M/K_{\max}\} + 1/2$ then

$$E\{TC\} \cong (K_{\max} + V)(j-1) + E\{M\} + E\{M\}V/K_{\max} + V/2$$

$$\frac{\partial E\{TC\}}{\partial K_{\max}} = (j-1) - \frac{E\{M\}V}{K_{\max}^{2}} = 0$$

$$K_{\max}^{*} \cong \sqrt{E\{M\}V/(j-1)}$$

Example Data Link Protocols

- HDLC
- SLIP (Internet)
- PPP (Internet)

HDLC

- High-level Data Link Control
- Derived from SDLC in IBM SNA
- Bit oriented and bit-stuffing for data transparency
- Fields in the frame format
  - Address : identification of terminals
  - Control : sequence numbers, ack and other purposes

- Data : arbitrary long
- Checksum : CRC using CRC-CCITT as the generator polynomial
- Flag : 01111110

- Operation modes
  - Normal Response Mode : Master-slave type, use polling
  - Asynchronous Response Mode : Master-slave type, but the secondary is not tightly restricted

- Asynchronous Balanced Mode : for full duplex point to point links stations

- Three kinds of frames
  - Information frame : go back n with mod of 8
  - Supervisory frame : use for a speedy ack (ex. type : Receive-Ready (RR), Receive-Not-Ready (RNR), REJect, Selective-REJect
  - Unnumbered frame : link initialization, link disconnection

    Poll/Final - unnumbered frame with P=1 requires an ack with F=1 from the secondary

Bits	8	8	8	≥ 0	16	8
	0 1 1 1 1 1 1 0	Address	Control	Data	Checksum	0 1 1 1 1 1 1 0

Fig. 3-24. Frame format for bit-oriented protocols.

Bits	1	3		1	3
(a)	0	SN		P/F	RN
(b)	1	0	Type	P/F	RN
(c)	1	1	Type	P/F	Type

Fig. 3-25. Control field of (a) an information frame, (b) a supervisory frame, (c) an unnumbered frame.

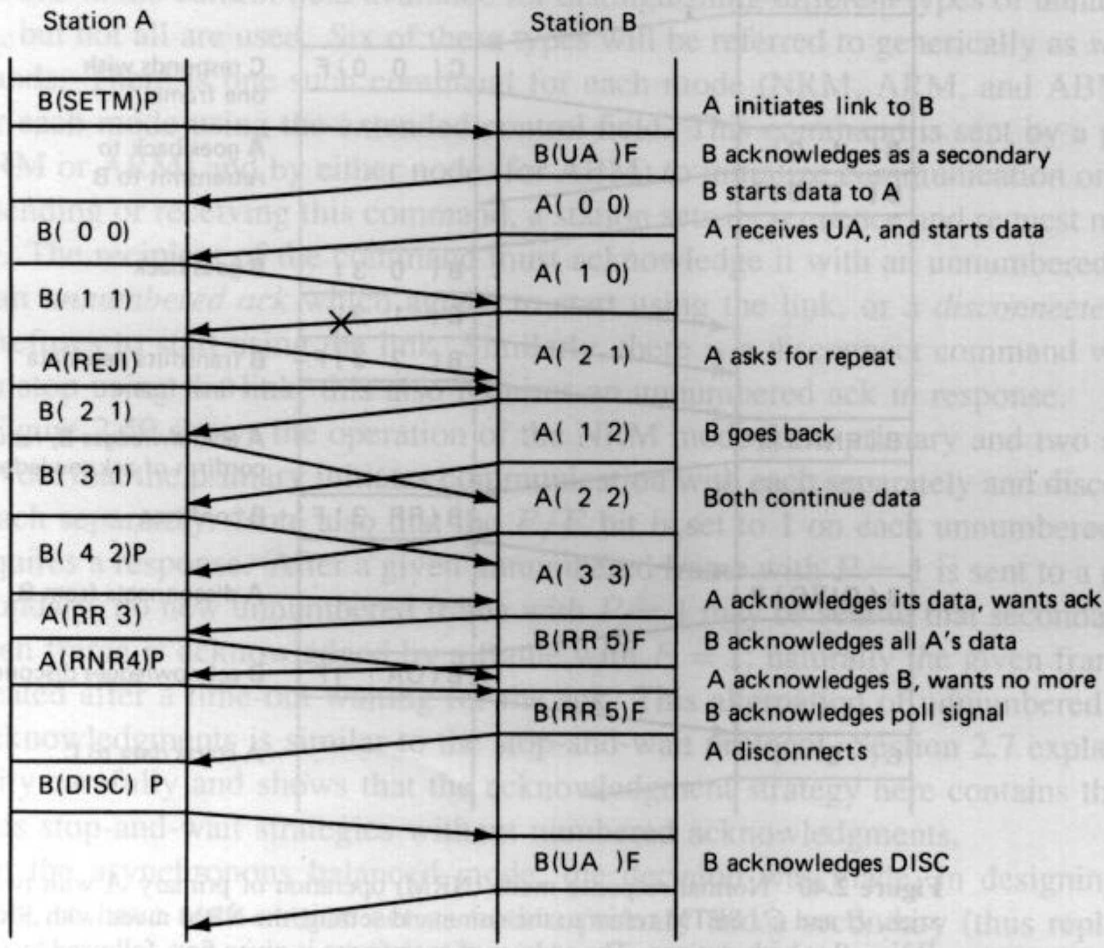


Figure 2.41 Asynchronous balanced mode (ABM) of operation. SETM refers to the set ABM command, which initiates the link with SN and RN at both sides equal to 0. Note the use of addresses in different types of frames.

Initialization and Disconnect for ARQ Protocols

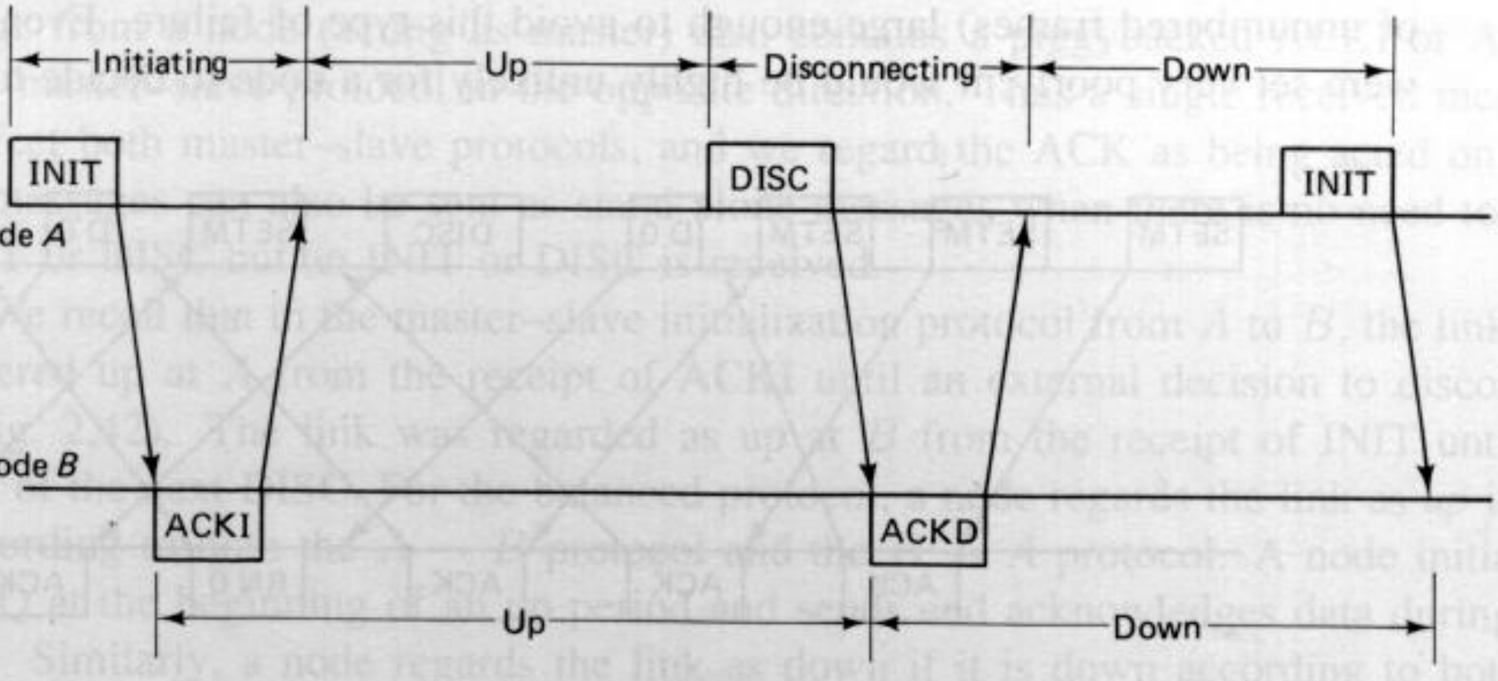


Figure 2.42 Initialization and disconnection protocol. Data are transmitted only in up periods and the ARQ is initialized at the beginning of an up period. Note that node A cycles between periods of initiating, being up, disconnecting, and being down.

- **Node failure case**

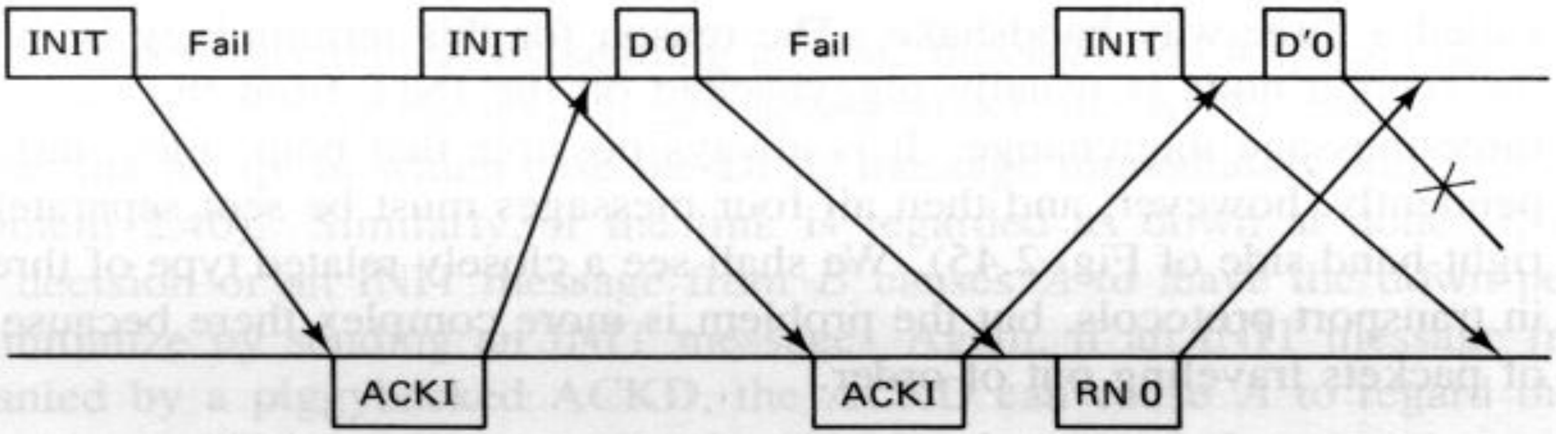


Figure 2.46 Example of a sequence of node failures, with loss of state, that causes a failure in correct initialization. Each time the node recovers from a failure, it assumes that ACKD was the last protocol message received, thus allowing acknowledgments from earlier initialization periods to get confused with later periods.

- ACK for D0 was considered the ACK for D'0
- B will never have chance to receive D'0
- Sufficiently long time-out

SLIP

- Serial Line IP : RFC 1055
- The home PC acts like a Internet host (All Internet services are available)
- SLIP is older than PPP
- Devised in 1984 to connect Sun workstations to the Internet over a dial-up line using a modem
- The workstation sends raw IP packets over the line

- Problems
  - No error detection or correction
  - It supports only IP
  - Each side must know the other IP address in advance : no dynamic address assignment
  - No authentication
  - Not an approved Internet Standard (Many versions)

PPP

- Point-to-Point Protocol
- IETF’s improvement (RFC 1661 & 1662)
- PPP
  - handles error detection
  - supports multiple protocols, e.g., IP, AppleTalk
  - allows IP addresses to be negotiated at connection time
  - permits authentication

- PPP provides
  - a framing method for delineation and error detection
  - LCP (Link Control Protocol) for bringing lines up, testing them, negotiating options, and bringing them down
  - NCP (Network Control Protocol) to negotiate network-layer options (ex. IP address)

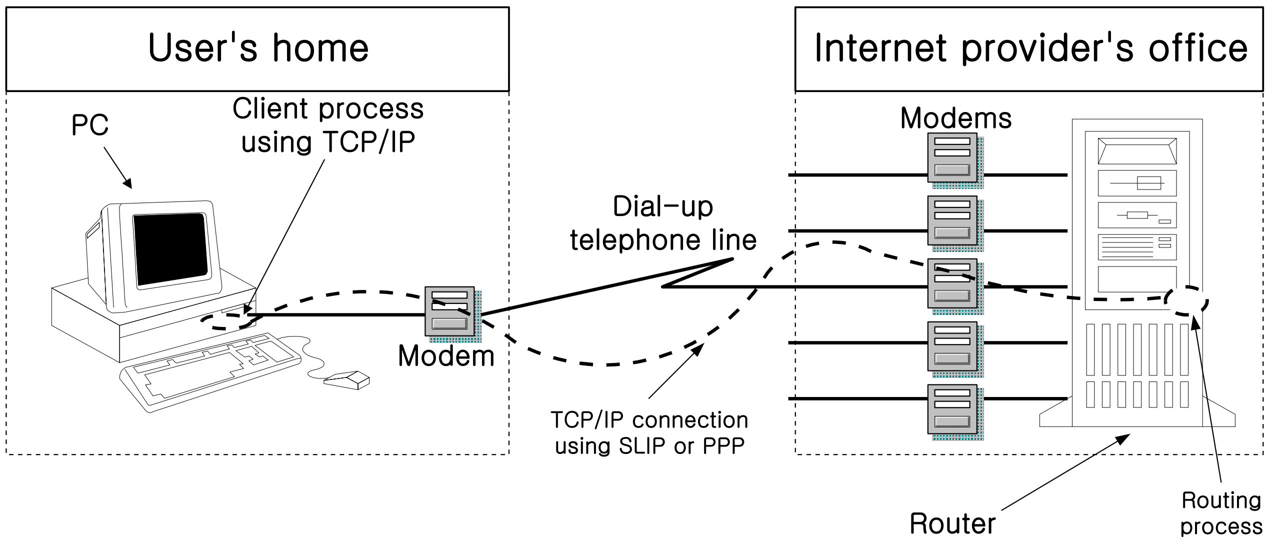


Fig. 3-26. A home personal computer acting as an internet host.

- PPP frame format
  - similar to that of HDLC
  - character oriented
  - Address field is always 11111111 : no need to assign data link addresses
  - use of unnumbered frames (no sequence number)
  - variable length of the payload field

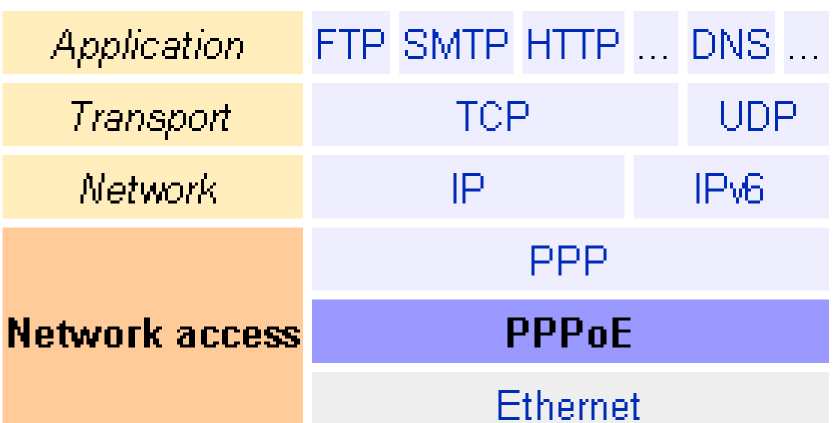
Bytes	1	1	1	1 or 2	Variable	2 or 4	1
	Flag 01111110	Address 11111111	Control 00000011	Protocol	Payload	Checksum	Flag 01111110

Fig. 3-27. The PPP full frame format for unnumbered mode operation.

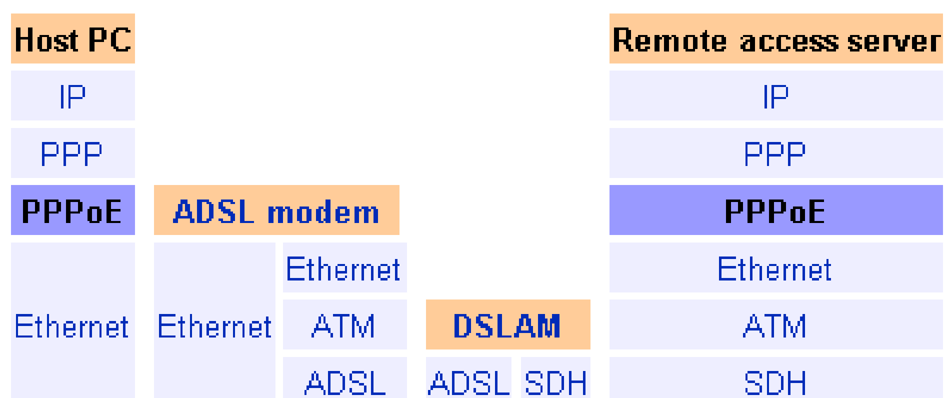
PPPoE (over Ethernet)

- Widely used for xDSL
  - PPP frames encapsulated in Ethernet frame
  - Provide PPP's connection-oriented service to Ethernet users (e.g., authentication, billing, etc.)

PPPoE and TCP/IP protocol stack



ADSL internet access architecture

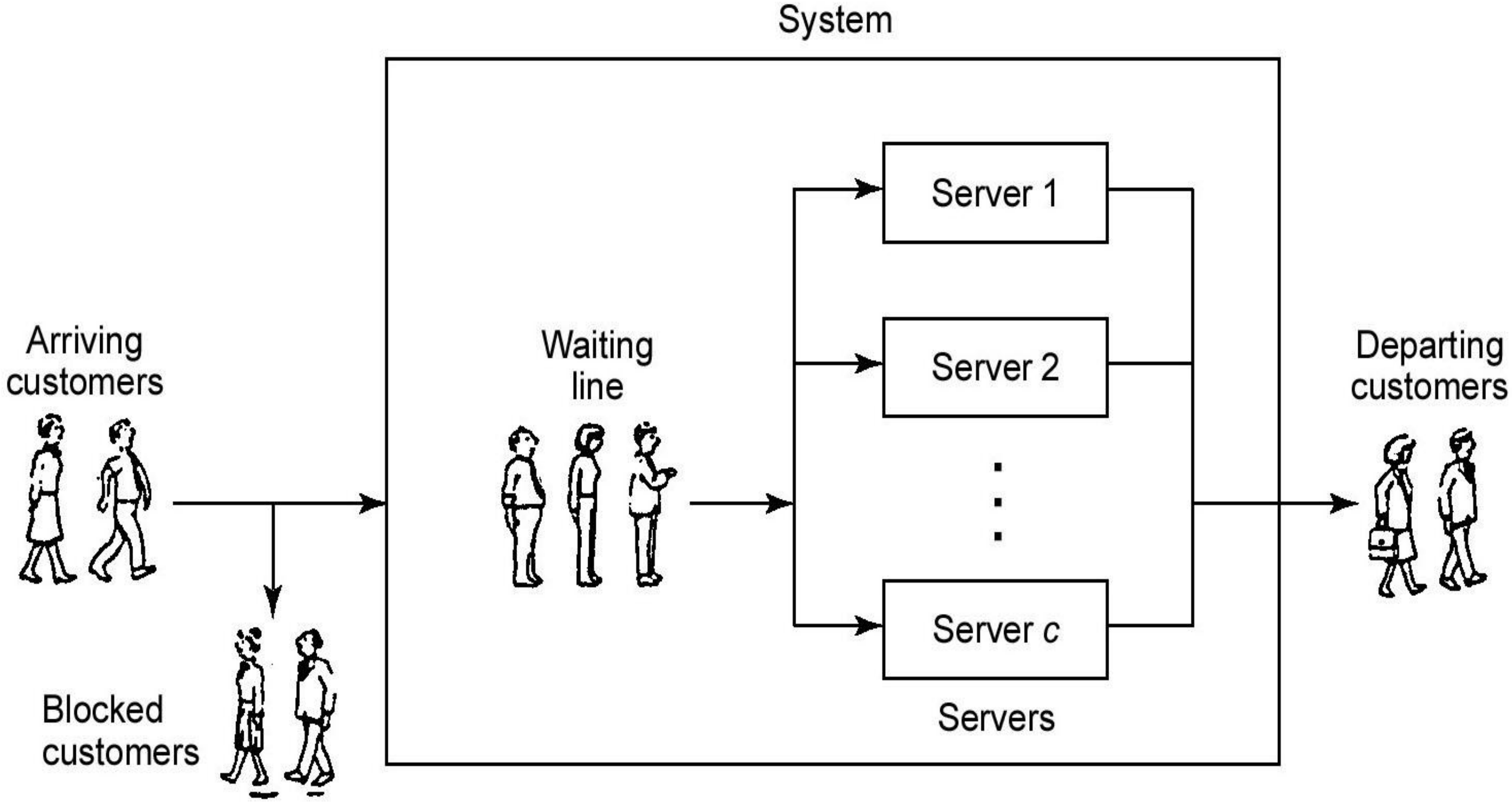


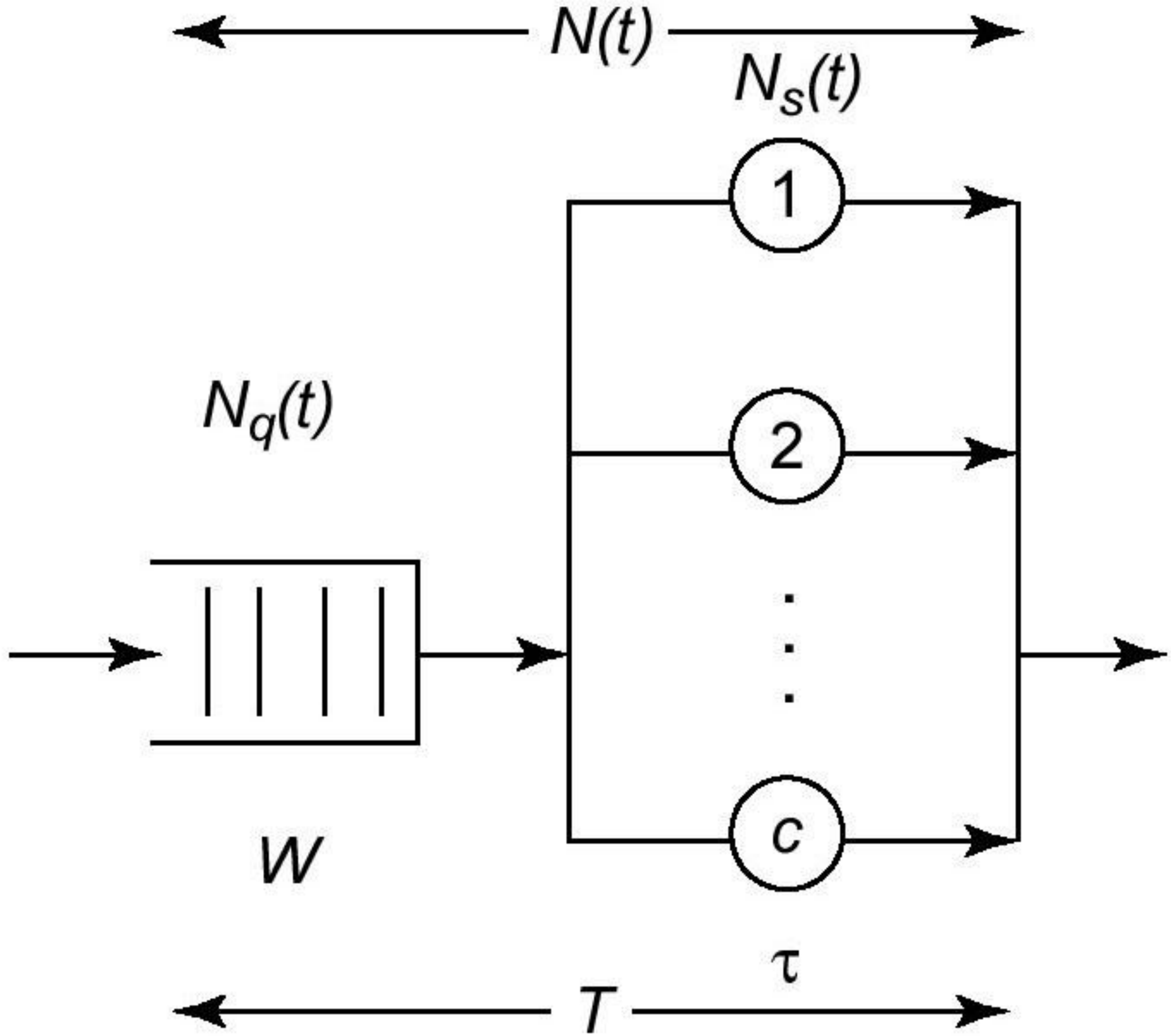
Queueing Theory (Delay Models)

Sunghyun Choi

Adopted from Prof. Saewoong Bahk's material

Introduction





N(t)
Ns(t)
1
2
Nq(t)
c
W
τ
T

- Total delay of the i-th customer in the system

$$T_i = W_i + \tau_i$$

  - $N(t)$: the number of customers in the system
  - $N_q(t)$: the number of customers in the queue
  - $N_s(t)$: the number of customers in the service
  - W : the delay in the queue
  - τ : the service time

- T : the total delay in the system
- λ: the customer arrival rate [#/sec]

Little's Theorem

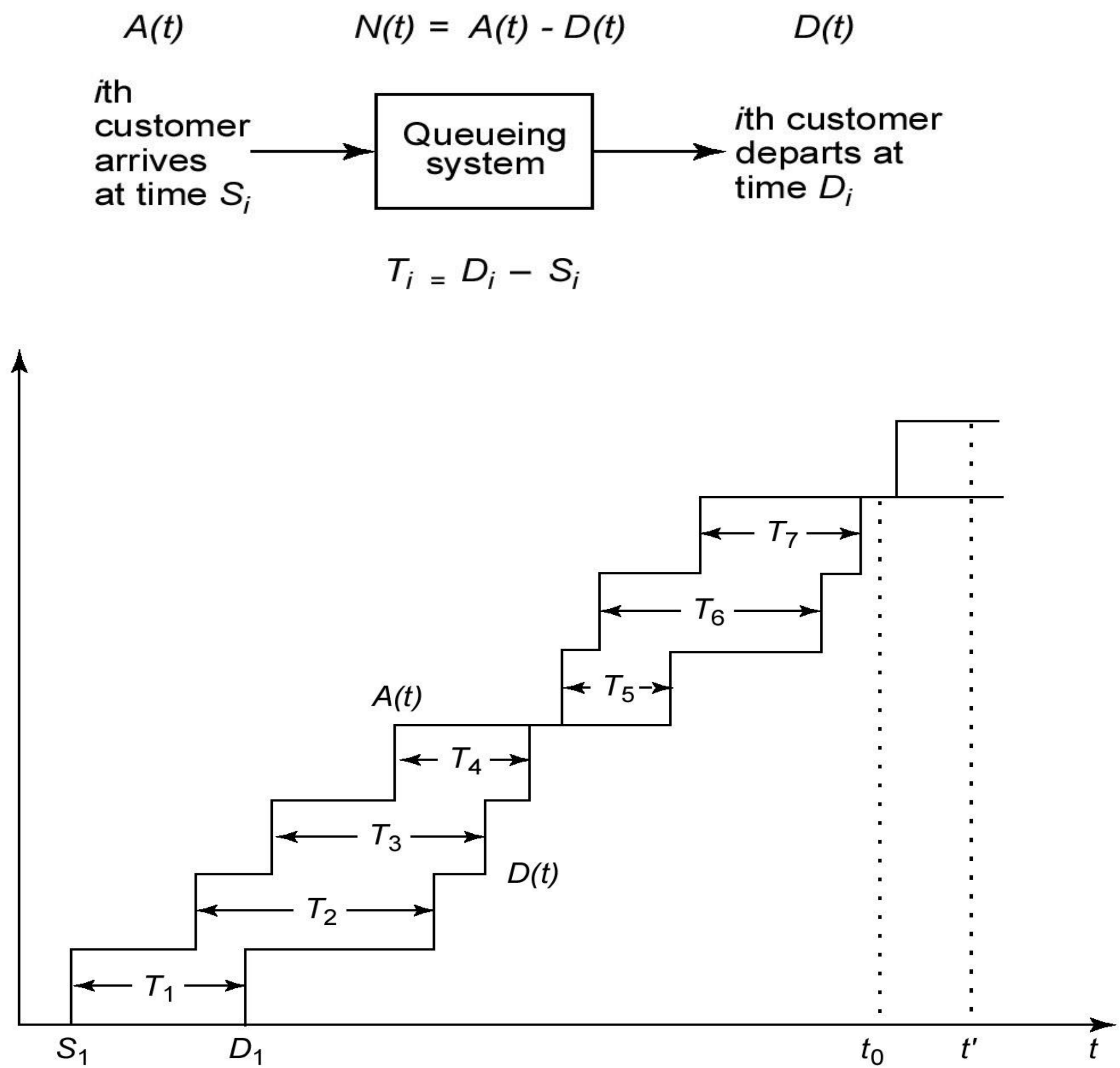
$$E[N] = \lambda E[T]$$

- Number of customer in the system at t

$$N(t)=A(t)-D(t)$$

where

- D(t) : the number of customer departures up to time t
- A(t) : the number of customer arrivals up to time t



A(t)
N(t) = A(t) - D(t)
D(t)
ith customer arrives at time S_i
Queueing system
ith customer departs at time D_i
T_i = D_i − S_i
A(t)
D(t)
T_1
T_2
T_3
T_4
T_5
T_6
T_7
S_1
D_1
t_0
t'
t

– Time average of the number $N(t)$ of customers in the system during the interval $(0,t]$, where $N(t) = 0$

$$< N >_t = \frac{1}{t}\int_0^t N(t')dt'$$

$$= \frac{1}{t}\sum_{i=1}^{A(t)} T_i$$

$$< \lambda >_t = \frac{A(t)}{t}$$

$$< N >_t = < \lambda >_t \frac{1}{A(t)}\sum_{i=1}^{A(t)} T_i$$

- Let $<T>_t$ be the average of the times spent in the system by the first $A(t)$ customers

$$<T>_t = \frac{1}{A(t)} \sum_{i=1}^{A(t)} T_i$$

  Then $<N>_t = <\lambda>_t <T>_t$
- Assume an ergordic process and $t \rightarrow \infty$, then
- $E[N] = \lambda E[T]$
- This relationship holds even in non-FIFO case

– $E[N_q] = \lambda E[W]$

- Server utilization

$E[N_s] = \lambda E[\tau]$

where utilization factor

$\rho=\lambda/(\mu c) < 1$ to be stable for c server case

Review of Markov chain theory

- Discrete time Markov chains
  - discrete time stochastic process $\{X_n \mid n=0,1,2,..\}$ taking values from the set of nonnegative integers
  - Markov chain if

$$P_{ij} = P\{X_{n+1} = j \mid X_n = i, X_{n-1} = i_{n-1}, \cdots, X_0 = i_0\}$$

$$= P\{X_{n+1} = j \mid X_n = i\}$$

where $$P_{ij} \geq 0, \quad \sum_{j=0}^{\infty} P_{ij} = 1, \quad i = 0,1,\cdots$$

– The transition probability matrix

$$P = \begin{bmatrix} P_{00} & P_{01} & P_{02} & \cdots \\ P_{10} & P_{11} & P_{12} & \cdots \\ \cdots & \cdots & & \\ & & & \end{bmatrix}$$

– n-step transition probabilities

$$P_{ij}^{n} = P\{X_{n+m} = j \mid X_m = i\}$$

- Chapman-Kolmogorov equations

$$P_{ij}^{n+m} = \sum_{k=0}^{\infty} P_{ik}^{n} P_{kj}^{m} \qquad (n, m, i, j \geq 0)$$

- Stationary distribution

$$p_j = \sum_{i=0}^{\infty} p_i P_{ij} \quad (j \geq 0) \qquad \sum_{j=0}^{\infty} p_j = 1$$

- For irreducible and aperiodic MCs, there exists

$$p_j = \lim_{n \to \infty} P\{X_n = j \mid X_0 = i\} \qquad (i \geq 0)$$

and we have (w/ probability 1)

$$p_j = \lim_{k \to \infty} \frac{\text{\# of visits to state } j \text{ up to time } k}{k}$$

– Theorem. In an irreducible, aperiodic MC, there are two possibilities for $p_j = \lim_{n\to\infty} P\{X_n = j \mid X_0 = i\}$

- $\forall j, p_j = 0$ no stationary distribution
- $\forall j, p_j \geq 0$ unique stationary distribution of the MC

– Example for case 1: a queueing system with arrival rate exceeding the service rate

– Case 2: global balance equation

$$p_j \sum_{i=0}^{\infty} P_{ji} = \sum_{i=0}^{\infty} p_i P_{ij} \quad (j \geq 0) \text{ since } \sum_{i=0}^{\infty} P_{ji} = 1$$

- At equilibrium, frequencies out of and into state j are the same

– Generalized global balance equation

$$\sum_{j\in S} p_j \sum_{i\notin S} P_{ji} = \sum_{i\notin S} p_i \sum_{j\in S} P_{ij}$$

- For each transition out of S, there must be (w/ prob. 1) a reverse transition into S at some later time
- Frequency of transitions out of *S* equals that into *S*

– Detailed balance equation: holds for many MCs

$$p_j P_{ji} = p_i P_{ij},\ i, j \geq 0$$

- for birth-death systems $p_n P_{n,n+1} = p_{n+1} P_{n+1,n}$

- Continuous time Markov chains
  - $\{X(t) | t \geq 0\}$ taking nonnegative integer values
  - υ_i : the transition rate out of state *i*
  - q_{ij} : the transition rate from state *i* to *j*

$$q_{ij} = \upsilon_i P_{ij}$$

  - the steady state occupancy probability of state *j*

$$p_j = \lim_{t \to \infty} P[X(t) = j \mid X(0) = i]$$

  - Analog of detailed balance equations for DTMC

$$p_j q_{ji} = p_i q_{ij}$$

M/M/1 queueing system

- Arrival statistics:

  stochastic process $\{A(t) | t \geq 0\}$ taking nonnegative integer values is called a Poisson process with rate λ if

  - *A(t)* is a counting process representing the total number of arrivals from 0 to *t*
  - # of arrivals that occur in disjoint time intervals are independent
  - probability distribution function

$$P[A(t+\tau) - A(t) = n] = e^{-\lambda\tau} \frac{(\lambda\tau)^n}{n!} \quad , \qquad n = 0,1,2,\ldots$$

- Characteristics of the Poisson process
  - Interarrival times are independent and exponentially distributed
  - That is, if t_n denotes the n-th arrival time and the interval $\tau_n = t_{n+1} - t_n$, the probability distribution is

$$P[\tau \leq s] = 1 - e^{-\lambda s}, s \geq 0$$

– The interarrival probability density function

$$p(\tau_n) = \lambda e^{-\lambda \tau_n}$$

– mean: $1/\lambda$, variance: $1/\lambda^2$

– for every t, $\delta \geq 0$

$$P[A(t+\delta) - A(t) = 0] = 1 - \lambda\delta + o(\delta)$$

$$P[A(t+\delta) - A(t) = 1] = \lambda\delta + o(\delta)$$

$$P[A(t+\delta) - A(t) > 1] = o(\delta)$$

where $\lim_{\delta \to 0} \frac{o(\delta)}{\delta} = 0$

- If A_1, A_2, ..., A_k are merged into a process A, A is Poisson with a rate equal to $\lambda_1 + \lambda_2 + \ldots + \lambda_k$

- Service statistics
  - The service times are exponentially distributed with parameter μ . The service time of the *n*-th customer s_n :

$$P[s_n \leq s] = 1 - e^{-\mu s}, s \geq 0$$

where μ is the service rate

- Poisson Process (mδ=T)
  - P["1 arrival in *m*-th interval"] ~λδ
  - P["no arrival in *m*-th interval"] ~1-λδ
  - P["*k* arrivals in (0,*T*)"] ~ ${}_mC_k(\lambda\delta)^k(1-\lambda\delta)^{m-k}$

$$P[N_T = k] = \lim_{m\to\infty} {}_mC_k\left(\frac{\lambda T}{m}\right)^k\left(1-\frac{\lambda T}{m}\right)^{m-k}$$

$$= \lim_{m\to\infty}\frac{m}{m}\frac{m-1}{m}\cdots\frac{m-k+1}{m}\frac{(\lambda T)^k}{k!}\left(1-\frac{\lambda T}{m}\right)^{m-k}$$

$$= \frac{(\lambda T)^k}{k!}\lim_{m\to\infty}\left(1-\frac{\lambda T}{m}\right)^m\left(1-\frac{\lambda T}{m}\right)^{-k} \quad = e^{-\lambda T}\frac{(\lambda T)^k}{k!}$$

– mean

$$E[N_T] = \sum_{k=1}^{\infty} k \frac{e^{-\lambda T}(\lambda T)^k}{k!} = e^{-\lambda T}(\lambda T)\sum_{k=1}^{\infty}\frac{(\lambda T)^{k-1}}{(k-1)!}$$

$$= \lambda T$$

– variance $\sigma^2(N_T) = E[N^2{}_T] - (E[N_T])^2 = \lambda T$

- Memoryless property (if exponentially distributed)

$$P[\tau_n > \gamma + t \,|\, \tau_n > t] = \frac{P[\tau_n > \gamma + t]}{P[\tau_n > t]} = \frac{e^{-\lambda(\gamma+t)}}{e^{-\lambda t}} = P[\tau_n > \gamma]$$

- Markov chain (MC) formulation
  - Consider a discrete time MC

$$P_{ij} = P[N_{k+1} = j \mid N_k = i]$$

where N_k is the number of customers at time k and $N(t)$ is the number of customers at time t

  - probabilities

$$P[1 \text{ arrival in } \delta] = \frac{(\lambda\delta)e^{-\lambda\delta}}{1!} = \lambda\delta(1 - \frac{\lambda\delta}{1!} + \cdots) = \lambda\delta + o(\delta)$$

  - where the arrival and departure processes are independent

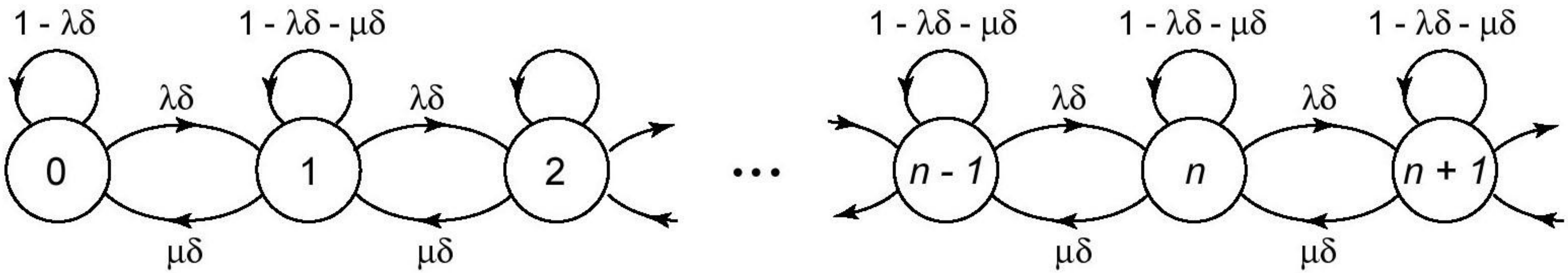
P[1 arrival and no departure in δ]=

$$\frac{(\lambda\delta)e^{-\lambda\delta}}{1!}\frac{(\mu\delta)^0 e^{-\mu\delta}}{0!} = \lambda\delta(1-\frac{\lambda\delta}{1!}+\cdots)(1-\frac{\mu\delta}{1!}+\cdots) = \lambda\delta + o(\delta)$$

$$P[0 \text{ arrival and one departure in } \delta] = \mu\delta + o(\delta)$$

$$P[0 \text{ arrival and 0 departure in } \delta] = 1 - \lambda\delta - \mu\delta + o(\delta)$$

where the arrival and departure processes are independent



- Global balance equation

$$(\lambda + \mu) p_j = \lambda p_{j-1} + \mu p_{j+1} \qquad \lambda p_0 = \mu p_1$$

$$p_n \lambda = p_{n+1} \mu$$

$$p_{n+1} = \rho^{n+1} p_0 \quad , \quad n = 0, 1, \ldots, \quad \rho = \lambda / \mu$$

from $\quad 1=\sum_{n=0}^{\infty} p_n = \sum_{n=0}^{\infty} \rho^n p_0 = \frac{p_0}{1-\rho}$

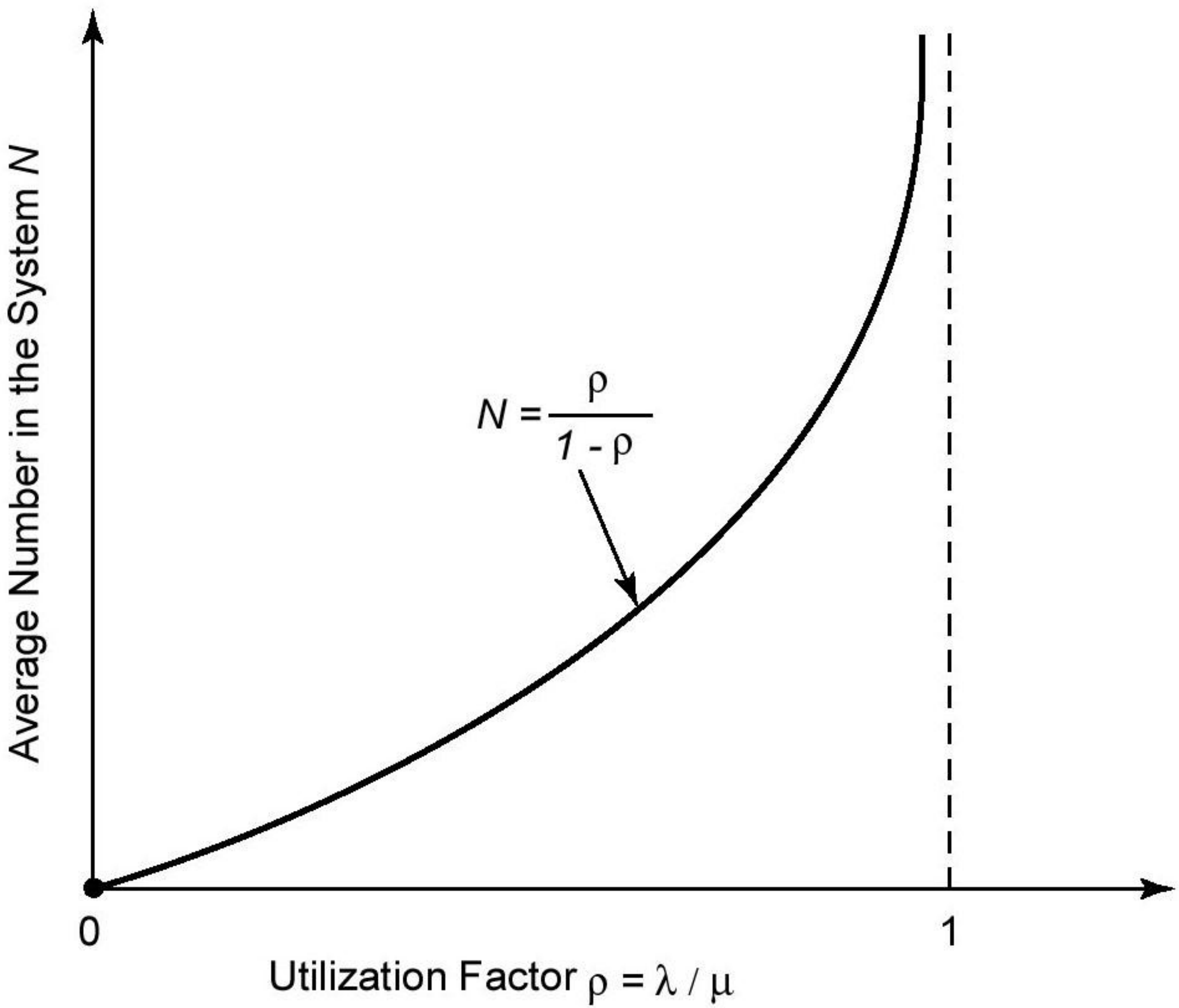
Then $\quad p_0 = 1-\rho$

– Average number of customers in the system

$$N = \lim_{t\to\infty} E[N(t)] = \sum_{n=0}^{\infty} np_n = \sum_{n=0}^{\infty} n\rho^n(1-\rho)$$

$$= \rho(1-\rho)\sum_{n=0}^{\infty} n\rho^{n-1} = \rho(1-\rho)\frac{\partial}{\partial\rho}(\sum_{n=0}^{\infty}\rho^n)$$

$$= \rho(1-\rho)\frac{\partial}{\partial\rho}(\frac{1}{1-\rho}) = \frac{\rho}{1-\rho} = \frac{\lambda}{\mu-\lambda}$$


Average Number in the System N
$N = \frac{\rho}{1 - \rho}$
0
1
Utilization Factor $\rho = \lambda / \mu$

- Average delay per customer (waiting time + service time)

$$E[T] = \frac{E[N]}{\lambda} = \frac{1}{\mu - \lambda} \quad \text{by Little's theorem}$$

- Average waiting time

$$E[W] = E[T] - \frac{1}{\mu} = \frac{\rho}{\mu - \lambda}$$

- Average number of customers in queue

$$E[N_Q] = \lambda E[W] = \frac{\rho^2}{1 - \rho}$$

- Server utilization (ave. # of customers in service)

$$1 - p_0 = \rho = \frac{\lambda}{\mu}$$

– Example 1

1/λ=4 ms, 1/μ=3 ms

$$E[N] = \frac{\rho}{1-\rho} = \frac{3/4}{1-3/4} = 3$$

$$E[T] = \frac{E[N]}{\lambda} = \frac{3}{1/4} = 12[ms]$$

- Example 2
  - Both arrival and service rates increase by k times
  - Average number of customers remains the same
  - But, average delay will decrease by k times
- Example 3: statistical multiplexing vs. time- and freq.-division multiplexing
  - For m Poisson packet streams with rate λ/m
  - If transmission capacity is divided into m equal portions, each portion behaves like an M/M/1 queue
  - Average delay will increase by m times compared with statistical multiplexing

- Occupancy Distribution upon Arrival

$$a_n = \lim_{t\to\infty} P\{N(t) = n \mid \text{an arrival occured just after time } t\}$$

$$p_n = \lim_{t\to\infty} P\{N(t) = n\}$$

$$a_n = p_n \ ???$$

Yes for M/M/1 system

  – Condition for the equality
    - Poisson arrivals where interarrival times and service times are independent!

– Formal proof: consider $a_n(t)$ and $p_n(t)$

$$a_n(t) = P\{N(t) = n \,|\, \text{an arrival occured just after time } t\}$$

$$= \lim_{\delta\to\infty} P\{N(t) = n \,|\, A(t, t+\delta)\} \text{ } (A(t,t+\delta) \text{ is an arrival event})$$

$$= \lim_{\delta\to\infty} \frac{P\{N(t) = n, A(t,t+\delta)\}}{P\{A(t,t+\delta)\}} \text{ (using Bayes' rule)}$$

$$= \lim_{\delta\to\infty} \frac{P\{A(t,t+\delta) \,|\, N(t) = n\} P\{N(t) = n\}}{P\{A(t,t+\delta)\}}$$

$$= P\{N(t) = n\} = p_n(t)$$

(since arrival is independent from the current number)

- Counter example 1:
  - Arrival is not Poisson, but independent and uniformly distributed between 2 and 4 sec.
  - Service time is fixed at 1 sec
  - An arrival customer always see an empty system
  - But, the average number seen by an outside observer is 1/3
  - Since $\lambda = 1/3$ while $T = 1$

- Counter example 2:
  - Arrival is Poisson, but the service times and the future arrival times are correlated
  - That is, service time is always ½ of the subsequent interarrival time
  - Accordingly, an arrival customer always see an empty system
  - But, the average number seen by an outside observer is 1/2

- Occupancy Distribution upon Departure

$$d_n = \lim_{t\to\infty} P\{N(t) = n \mid \text{a departure occured just before time } t\}$$

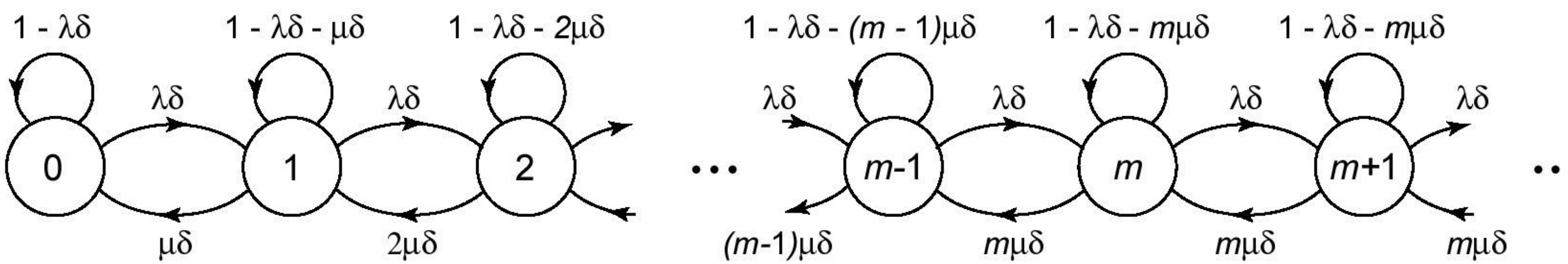
$$d_n = a_n \text{ ???}$$

Yes for "stable" M/M/1 system

  - Condition for the equality
    - System reaches a steady-state with all *n* having positive steady-state probabilities, and *N(t)* changes in unit increments
    - Holds for most stable single-queue systems

M/M/m, M/M/m/m, M/M/∞

- M/M/m (infinite buffer)
  - detailed balance equations in steady state



$$\lambda p_{n-1} = n\mu p_n, \; n \le m$$

$$\lambda p_{n-1} = m\mu p_n, \; n > m$$

$$p_n = \begin{cases} p_0 \dfrac{(m\rho)^n}{n!}, & n \le m \\ p_0 \dfrac{m^m \rho^n}{m!}, & n > m \end{cases}$$

where $\rho = \dfrac{\lambda}{m\mu} < 1$

From $\displaystyle\sum_{n=0}^{\infty} p_n = 1$

$$p_0 = [1 + \sum_{n=1}^{m-1} \frac{(m\rho)^n}{n!} + \sum_{n=m}^{\infty} \frac{(m\rho)^n}{m!} \frac{1}{m^{n-m}}]^{-1}$$

– The probability that all servers are busy

$$P_Q = P[\text{all servers are busy}] = \sum_{n=m}^{\infty} p_n = \frac{p_0 (m\rho)^m}{m!(1-\rho)}$$

- Erlang C formula

– expected number of customers waiting in queue

$$N_Q = \sum_{n=0}^{\infty} n p_{m+n} = \sum_{n=0}^{\infty} n p_0 \frac{m^m \rho^{m+n}}{m!} = P_Q \frac{\rho}{1-\rho}$$

– average waiting time of a customer in queue

$$W = \frac{N_Q}{\lambda} = \frac{\rho P_Q}{\lambda(1-\rho)}$$

– average delay per customer

$$T = \frac{1}{\mu} + W$$

– average number of customer in the system

$$N = \lambda T = \frac{\lambda}{\mu} + \frac{\lambda P_Q}{m\mu - \lambda}$$ by Little's theorem

- M/M/∞: The infinite server case
  - The detailed balance equations

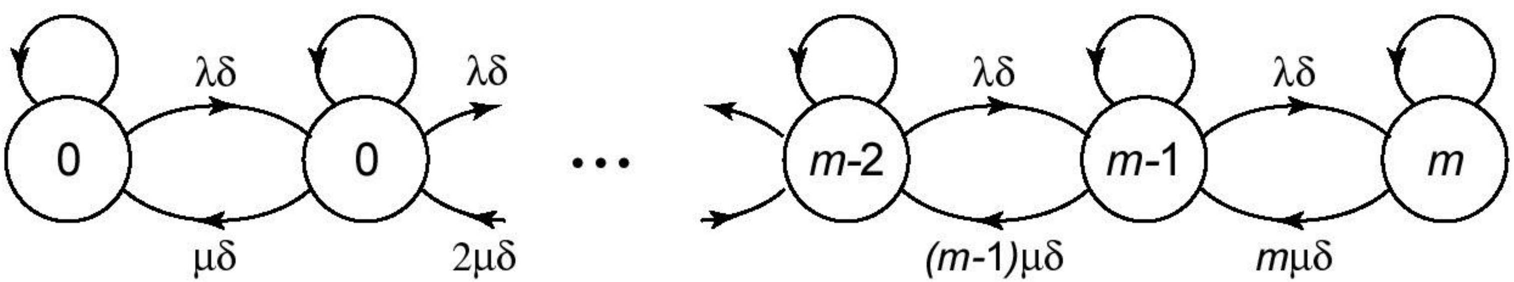
$$\lambda p_{n-1} = n\mu p_n \;, \qquad n = 1,2,\ldots$$

$$p_n = p_0 (\frac{\lambda}{\mu})^n \frac{1}{n!} \quad , \qquad n = 1,2,\ldots$$

$$\sum_{n=0}^{\infty} p_n = 1 \qquad\qquad p_0 = [1 + \sum_{n=1}^{\infty} (\frac{\lambda}{\mu})^n \frac{1}{n!}]^{-1} = e^{-\frac{\lambda}{\mu}}$$

Then $$p_n = (\frac{\lambda}{\mu})^n \frac{e^{-\frac{\lambda}{\mu}}}{n!} \;, n = 0,1,\cdots$$

- M/M/m/m : The m server loss system
  - when m servers are busy, next arrival will be lost
  - circuit switched network model



$$\lambda p_{n-1} = n\mu p_n, \ n = 1,2,\cdots,m$$

$$p_n = p_0 (\frac{\lambda}{\mu})^n \frac{1}{n!} \ ,n = 1,2,\cdots,m$$

$$\sum_{n=0}^{m} p_n = 1 \qquad\qquad p_0 = [\sum_{n=0}^{m} (\frac{\lambda}{\mu})^n \frac{1}{n!}]^{-1}$$

– The blocking probability (Erlang-B formula)

$$P_B = p_m = \frac{(\lambda / \mu)^m / m!}{\sum_{n=0}^{m} (\lambda / \mu)^n / n!}$$

M/G/1

- The service times are generally distributed
- X_i : the service time of the *i*-th arrival
  - $E[X] = 1/\mu$ = average service time
  - $E[X^2]$ = second moment of service time
- Assume:
  - Random variable X_i's are identically and independently distributed
  - Independent interarrival times

- Pollaczeck-Khinchin (P-K) formula

$$W = \frac{\lambda E[X^2]}{2(1-\rho)}$$

$$N_Q = \frac{\lambda^2 E[X^2]}{2(1-\rho)}$$

where

W : the customer waiting time in queue

W_i : the *i*-th customer's waiting time in queue

R_i : the residual service time seen by the *i*-th customer

N_i : the number of customers in queue seen by *i*-th customer

- Derivation of P-K formula

$$W_i = R_i + \sum_{j=i-N_i}^{i-1} X_j$$

$$E[W_i] = E[R_i] + E[\sum_{j=i-N_i}^{i-1} E[X_j \mid N_i]]$$

$$= E[R_i] + E[N_i]E[X]$$

(since N_i and X_i are independent)

Taking the limit,

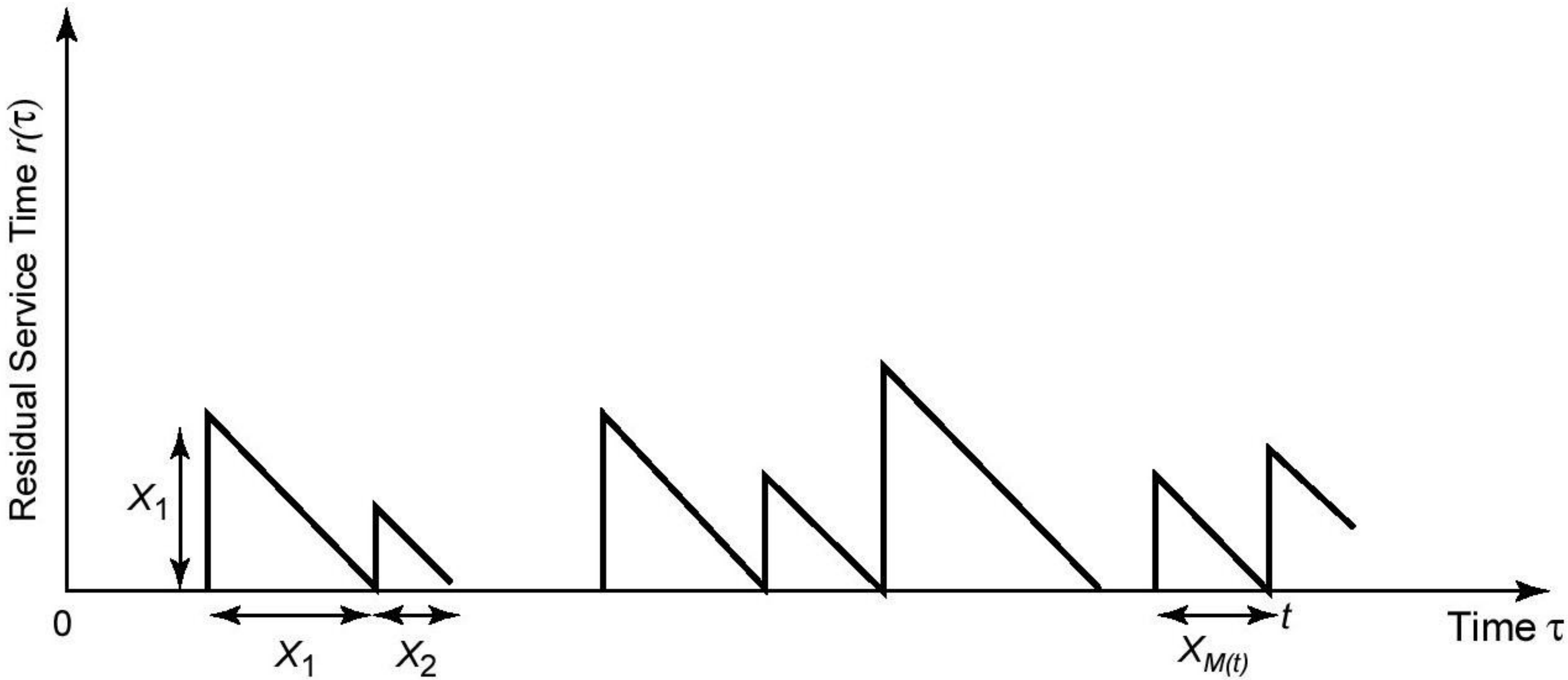
$$W = R + \frac{1}{\mu} N_Q = R + \frac{\lambda}{\mu} W = R + \rho W$$

Then $\quad W = \dfrac{R}{1-\rho}$

Define:

M(t): the number of customers departing during $[0,t]$

r(τ): the average of the residual service time



Residual Service Time $r(\tau)$
X_1
0
X_1
X_2
t
$X_{M(t)}$
Time τ

The time average of the residual service time

$$\frac{1}{t}\int_0^t r(\tau)d\tau = \frac{1}{t}\sum_{i=1}^{M(t)}\frac{1}{2}X_i^2 = \frac{1}{2}\frac{M(t)}{t}\frac{\sum_{i=1}^{M(t)}X_i^2}{M(t)}$$

Take the limit as $t \rightarrow \infty$ and assume the ergodic process

$$R = \lim_{t\to\infty}\frac{1}{t}\int_0^t r(\tau)d\tau = \frac{1}{2}\lim_{t\to\infty}\frac{M(t)}{t}\lim_{t\to\infty}\frac{\sum_{t=1}^{M(t)}X_i^2}{M(t)}$$

$$= \frac{1}{2}\lambda E[X^2]$$

Then $$W = \frac{\lambda E[X^2]}{2(1-\rho)}$$

$$T = E[X] + W = \frac{1}{\mu} + \frac{\lambda E[X^2]}{2(1-\rho)}$$

Example:

$$W = \frac{\rho}{\mu(1-\rho)}, \quad \text{for M/M/1 where } E[X^2] = \frac{2}{\mu^2}$$

$$= \frac{\rho}{2\mu(1-\rho)}, \quad \text{for M/D/1 where } E[X^2] = \frac{1}{\mu^2}$$

- Example: Delay analysis of go back *n* ARQ
  - Assume that a Poisson input process
  - Max. wait for an ack of n-1 frames before a packet is retransmitted
  - p : probability that the frame is rejected at the receiver due to channel error ignoring the feedback channel error.
  - X : effective service time = start of the first transmission of a given packet after the last transmission of the previous packet ~ end of the last transmission of the given packet

– Service time distribution

$$P[X = 1 + kn] = (1-p)p^k, \qquad k = 0,1,\ldots$$

$$E[X] = \sum_{k=0}^{\infty}(1+kn)(1-p)p^k = 1 + \frac{np}{1-p}$$

$$E[X^2] = \sum_{k=0}^{\infty} x^2 P[X = 1+kn] = \sum_{k=0}^{\infty}(1+kn)^2(1-p)p^k$$

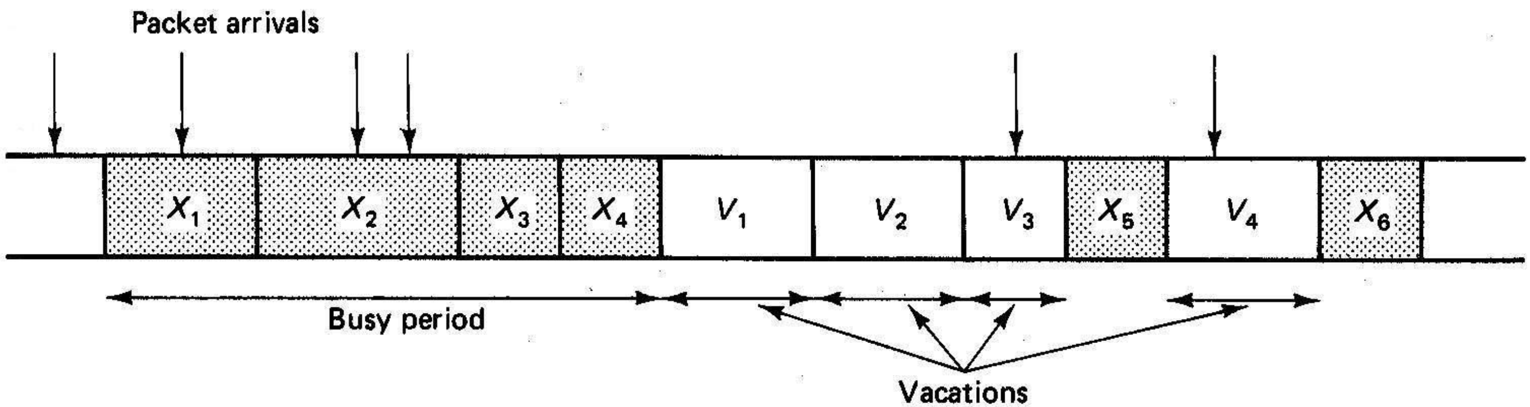
$$= 1 + \frac{2np}{1-p} + \frac{n^2(p+p^2)}{(1-p)^2}$$

Then

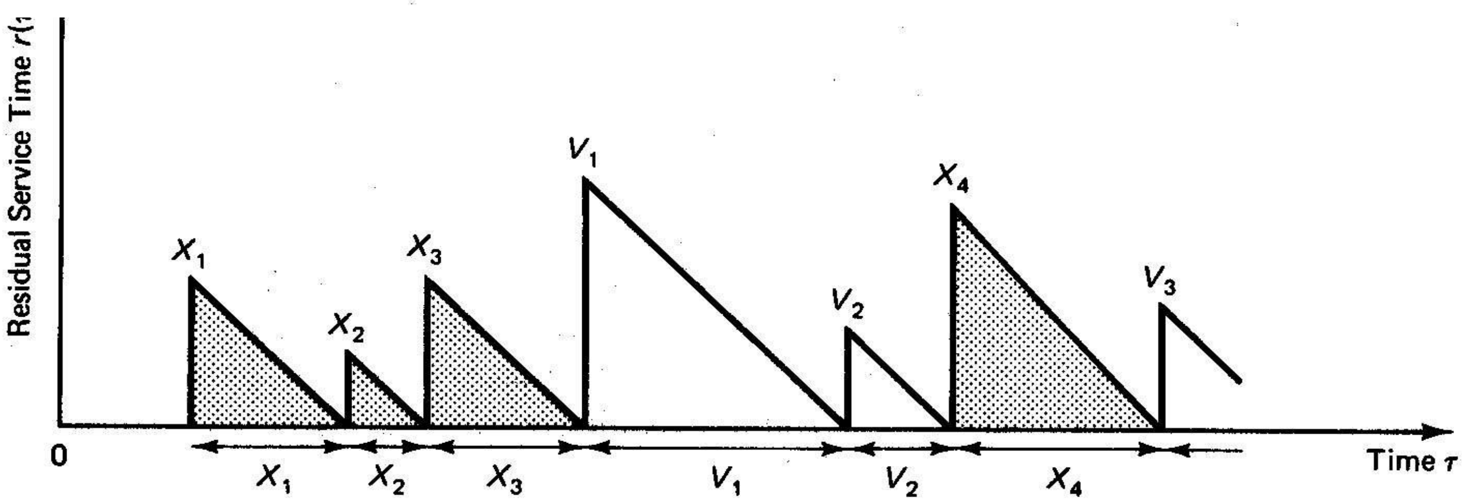
$$W = \frac{\lambda E[X^2]}{2(1-\rho)} \qquad where \quad \rho = \lambda E[X]$$

$$T = E[X] + W$$

M/G/1 queues with vacations



- Example of vacation = control packets



$$R=\frac{1}{t}\int_0^t \gamma(\tau)d\tau=\frac{1}{t}\sum_{i=1}^{M(t)}\frac{1}{2}X_i^{\,2}+\frac{1}{t}\sum_{i=1}^{L(t)}\frac{1}{2}V_i^2$$

$$=\frac{M(t)}{t}\frac{\sum_{i=1}^{M(t)}\frac{1}{2}X_i^{\,2}}{M(t)}+\frac{L(t)}{t}\frac{\sum_{i=1}^{L(t)}\frac{1}{2}V_i^2}{L(t)}$$

$$\lim_{t\to\infty}\frac{M(t)}{t}\lim_{t\to\infty}\frac{\sum_{i=1}^{M(t)}\frac{1}{2}X_i^{\,2}}{M(t)}=\frac{1}{2}\lambda E\left[X^2\right]$$

$$\lim_{t\to\infty}\frac{L(t)}{t}\lim_{t\to\infty}\frac{\sum_{i=1}^{L(t)}\frac{1}{2}V_i^2}{L(t)}=\frac{(1-\rho)}{E[V]}\frac{1}{2}E\left[V^2\right]$$

where ρ is the server utilization. Then

$$W=\frac{R}{(1-\rho)}$$

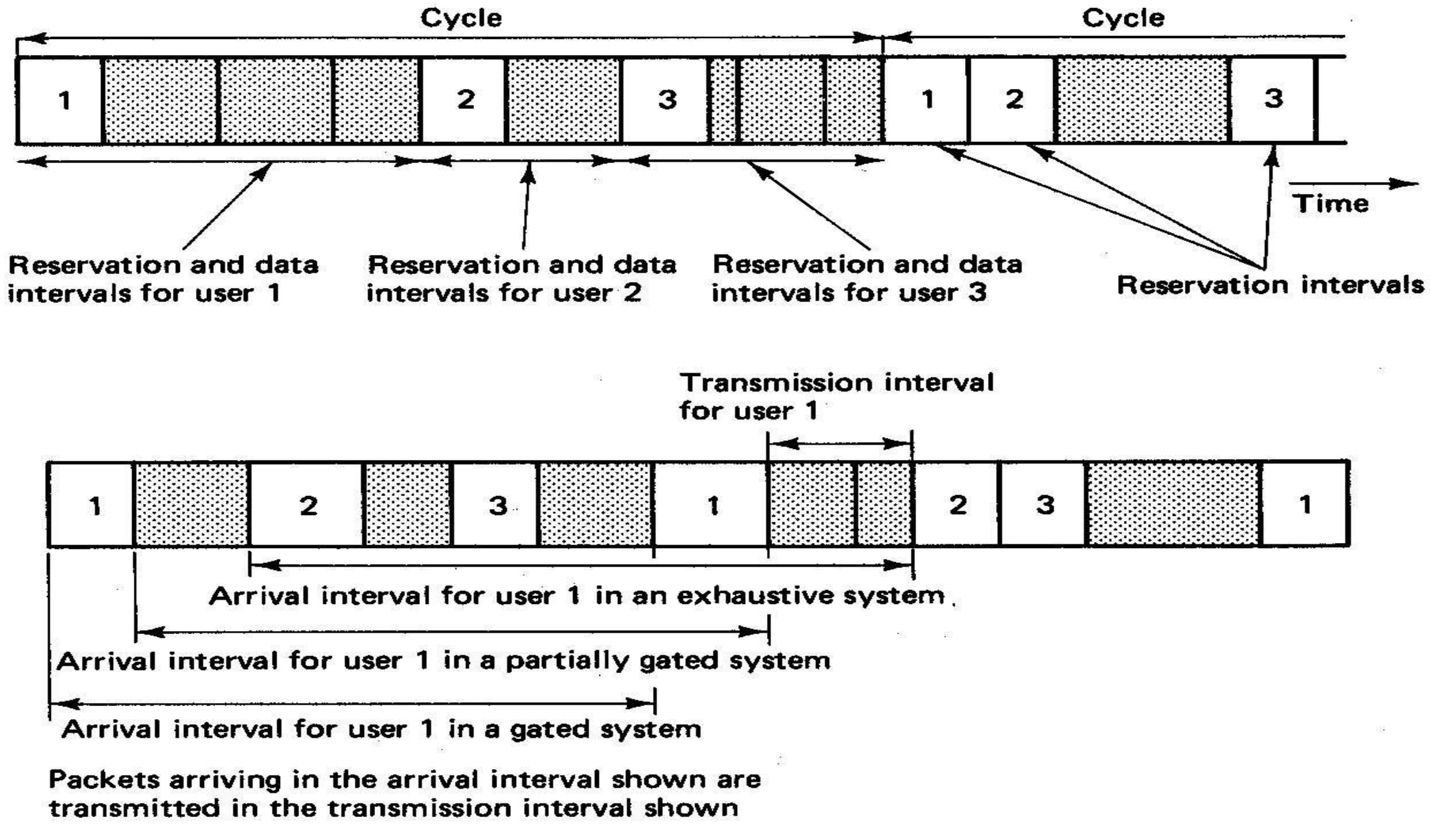
$$=\frac{1}{(1-\rho)}\left(\frac{1}{2}\lambda E\left[X^2\right]+\frac{(1-\rho)}{E[V]}\frac{1}{2}E\left[V^2\right]\right)$$

$$=\frac{\lambda E\left[X^2\right]}{2(1-\rho)}+\frac{E\left[V^2\right]}{2E[V]}$$

Reservations and Polling

- No use of statistical multiplexing
- A cycle consists of the reservation and data intervals
- Three versions of systems
  - Fully gated: a packet arriving during the user's reservation interval must wait for an entire cycle
  - Partially gated: a packet arriving during the user's data interval must wait for an entire cycle
  - Exhaustive: a packet arriving during the user's reservation or data interval is transmitted in the same data interval

Reservation (or polling) System with three users



- Assumption
  - Poisson arrival: λ/m (m users)
  - Service time: $E[X] = \frac{1}{\mu}$ $E[X^2]$
  - Each arrival requires different and independent length of reservation intervals
- Single user system (m=1)
  - Gated system
  - V_l : the duration of the l^{th} reservation interval

    $E[V]$, $E[V^2]$
  - $V_{l(i)}$: the duration of the reservation interval used for the i^{th} data packet reservation

* i^{th} arrival must wait until all the others arrived earlier make reservations.

– Expected queueing delay for the i^{th} packet

$$E[W_i] = E[R_i] + \frac{E[N_i]}{\mu} + E[V_{l(i)}]$$

– $$E[R] = \frac{\lambda E[X^2]}{2} + \frac{(1-\rho)E[V^2]}{2E[V]}$$ (M/G/1 with vacations)

$$\lim_{i\to\infty} \frac{E[N_i]}{\mu} = \frac{\lambda}{\mu} W = \rho W$$

$$\lim_{i\to\infty} E[V_{l(i)}] = E[V]$$

– The expected queueing time (gated single user system)

$$W = \frac{\lambda E[X^2]}{2} + \frac{(1-\rho)E[V^2]}{2E[V]} + \rho W + E[V]$$

$$W = \frac{\lambda E[X^2]}{2(1-\rho)} + \frac{E[V^2]}{2E[V]} + \frac{E[V]}{1-\rho}$$

– For constant reservation interval of A

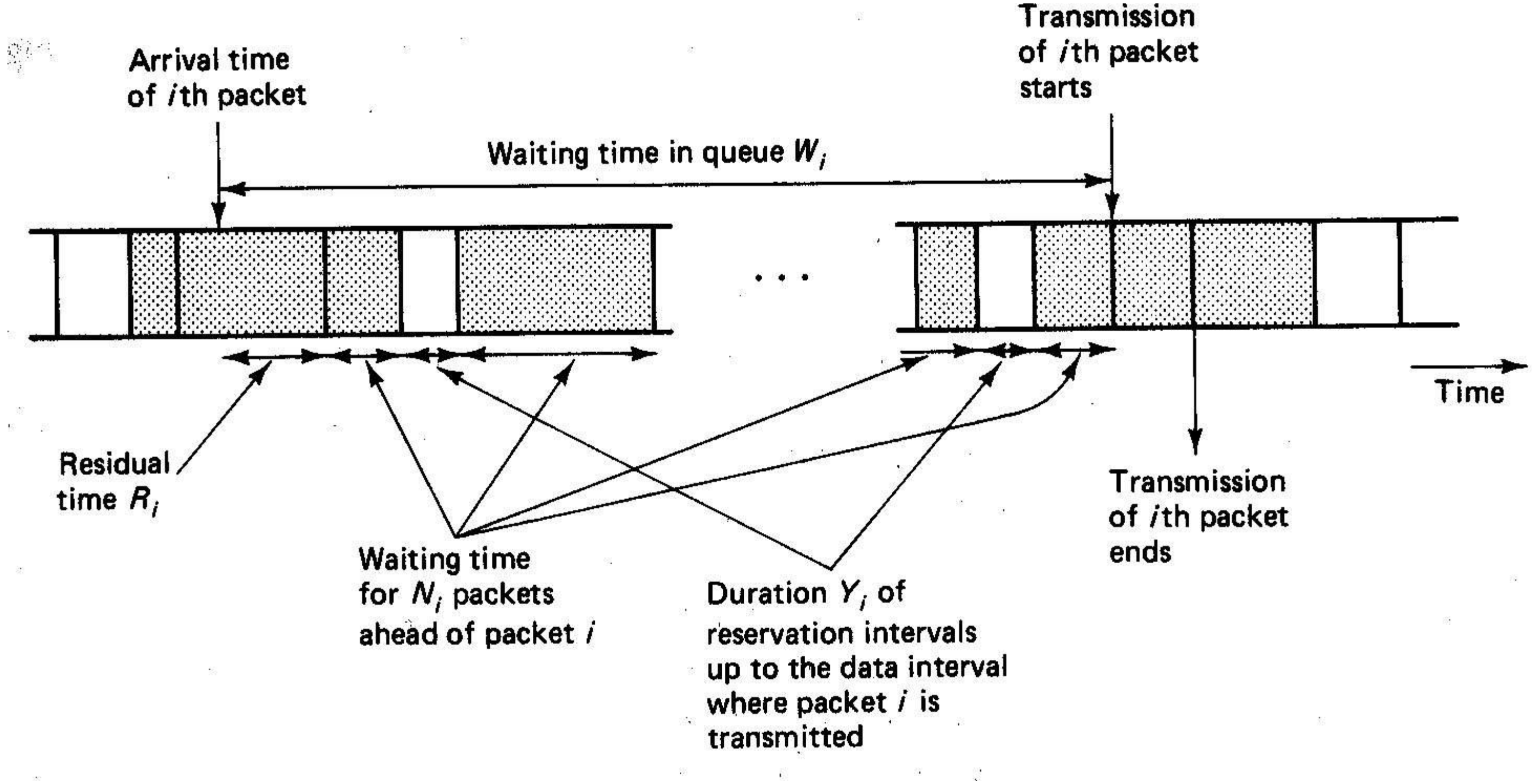
$$W = \frac{\lambda E[X^2]}{2(1-\rho)} + \frac{A}{2}\frac{3-\rho}{1-\rho}$$

Multiple user system (m users)

- Each user with independent Poisson arrival rate of λ/m
- $E[X]$, $E[X^2]$ for each user's service time
- $E[V_i]$, $E[V_i^2]$: reservation interval for user "i"
- Assume lth reservation interval is used to make reservations for user (l mod m) (i.e., 0 ~ m-1)

→ l^{th} interval for the corresponding reservations

– Exhaustive system



– Y_i: the duration of the whole reservation intervals for the *i*th system time

– $$E[W_i] = E[R_i] + \frac{1}{\mu} E[N_i] + E[Y_i]$$

– From M/G/1 with vacations (eq. 3-54) and taking the limit $i \to \infty$

$$R = \frac{\lambda E[X^2]}{2} + \frac{(1-\rho)\sum_{l=0}^{m-1} E[V_l^2]}{2\sum_{l=0}^{m-1} E[V_l]}$$

(independent of input arrival and res. interval processes)

– $\lim_{i\to\infty} \frac{E[N_i]}{\mu} = \frac{1}{\mu}\lambda W = \rho W$

– Define

$a_{lj} = E[Y_i \mid$ packet "i" arrives in user l's reservation or data interval and belongs to user $(l+j) \bmod m]$

$a_{lj} = 0, \text{ for } j = 0$

(exhaustive, no extra reservation for user l mod m)

$= E\left[V_{(l+1)\bmod m}\right] + \cdots + E\left[V_{(l+j)\bmod m}\right] \quad j = 1, 2, \cdots, m-1$

– $E[Y_i|$packet i arrives in user l's reservation or data interval

$$= \frac{1}{m}\sum_{j=0}^{m-1} a_{lj} = \sum_{j=1}^{m-1}\frac{m-j}{m}E\left[V_{(l+j)\bmod m}\right]$$

since paket "*i*" belongs to any user with equal prob. of *1/m*

– $\rho\frac{1}{m}$: prob. that a packet arrives during user l's data interval

– $(1-\rho)\dfrac{E[V_l]}{\sum_{k=0}^{m-1} E[V_k]}$: prob. that a packet arrives during user l's reservation interval

$$Y = \lim_{i\to\infty} E[Y_i]$$

$$= \sum_{l=0}^{m-1}\left(\sum_{j=1}^{m-1}\frac{m-j}{m}E\left[V_{(l+j)\bmod m}\right]\right)\left(\frac{\rho}{m}+(1-\rho)\frac{E[V_l]}{\sum_{k=0}^{m-1}E[V_k]}\right)$$

$$= \frac{(m-\rho)E[V]}{2} - \frac{(1-\rho)\sum_{l=0}^{m-1}E[V_l]^2}{2mE[V]} \text{ where } E[V] = \frac{1}{m}\sum_{l=0}^{m-1}E[V_l]$$

$$W = \frac{R+Y}{1-\rho}$$

$$= \frac{\lambda E\left[X^2\right]}{2(1-\rho)} + \frac{(m-\rho)E[V]}{2(1-\rho)} + \frac{\sum_{l=0}^{m-1}\left(E\left[V_l^2\right] - E[V_l]^2\right)}{2mE[V]}$$

$$= \frac{\lambda E\left[X^2\right]}{2(1-\rho)} + \frac{(m-\rho)E[V]}{2(1-\rho)} + \frac{\sigma_V^2}{2E[V]}$$

where $\sigma_V^2 = \frac{\sum_{l=0}^{m-1}\left(E\left[V_l^2\right] - E[V_l]^2\right)}{m}$

for the exhaustive multiuser system

$$W = \frac{\lambda E\left[X^2\right]}{2(1-\rho)} + \frac{(m+\rho)E[V]}{2(1-\rho)} + \frac{\sigma_V^2}{2E[V]}$$ (partially gated) since $Y' = Y + \rho E[V]$

$$W = \frac{\lambda E\left[X^2\right]}{2(1-\rho)} + \frac{(m+2-\rho)E[V]}{2(1-\rho)} + \frac{\sigma_V^2}{2E[V]}$$ (gated) since $Y'' = Y' + (1-\rho)E[V]$

–For constant reservation interval, i.e., A/m,

$$W = \frac{\lambda E\left[X^2\right]}{2(1-\rho)} + \frac{A}{2}\left(\frac{1-\rho/m}{1-\rho}\right)$$ (exhaustive)

$$W = \frac{\lambda E\left[X^2\right]}{2(1-\rho)} + \frac{A}{2}\left(\frac{1+\rho/m}{1-\rho}\right)$$ (partially gated)

$$W = \frac{\lambda E\left[X^2\right]}{2(1-\rho)} + \frac{A}{2}\left(\frac{1+(2-\rho)/m}{1-\rho}\right)$$ (gated)

Priority Queueing

- M/G/1 with *n* priority classes of customers
  - For class *k*, $\lambda_k, \overline{X_k} = \frac{1}{\mu_k}, \overline{X_k^2}$
  - Independent Poisson arrival processes
- Non preemptive priority
  - Complete the on-going service without interruption
  - Separate queue for each priority class
  - W_k : avg. number of customers in queue for priority *k*

    N_Q^k : avg. queueing time for priority *k*

    ρ_k : $= \frac{\lambda_k}{\mu_k}$

    R : mean residual service time

– $\sum_{i=1}^{n} \rho_i < 1$

– for class 1 (highest priority) and 2

$$W_1 = R + \frac{1}{\mu_1} N_Q^1 = R + \rho_1 W_1 \quad \text{Then}$$

$$W_1 = \frac{R}{1-\rho_1}$$

$$W_2 = R + \frac{1}{\mu_1} N_Q^1 + \frac{1}{\mu_2} N_Q^2 + \frac{1}{\mu_1} \lambda_1 W_2$$

additional queueing delay due to customers of class 1 that arrive while a class 2 customer is waiting in queue

$$= R + \rho_1 W_1 + \rho_2 W_2 + \rho_1 W_2$$

$$W_2 = \frac{R + \rho_1 W_1}{1 - \rho_1 - \rho_2} = \frac{R}{(1-\rho_1)(1-\rho_1-\rho_2)}$$

– For class k

$$W_k = \frac{R}{(1-\rho_1-\cdots-\rho_{k-1})(1-\rho_1-\rho_2-\cdots-\rho_k)}$$

– System delay for class k

$$T_k = \frac{1}{\mu_k} + W_k$$

– Mean residual service time

$$R_k = \frac{1}{2}\sum_{i=1}^{n}\lambda_i \overline{X_i^2}$$

– finally

$$W_k = \frac{\sum_{i=1}^{n}\lambda_i \overline{X_i^2}}{2(1-\rho_1-\cdots-\rho_{k-1})(1-\rho_1-\rho_2-\cdots-\rho_k)}$$

- Preemptive priority
  - High priority customer interrupts low priority customer's on-going service
  - T_k: ave. time in the system of priority k customer
  - Note that the presence of customers of priorities k +1 through n does not affect priority k customer's operation
  - Three components of T_k

$$T_k = \frac{1}{\mu_k} + \frac{R_k}{(1-\rho_1-\cdots-\rho_k)} + \left(\sum_{i=1}^{k-1}\rho_i\right)T_k$$

where $R_k = \dfrac{\sum_{i=1}^{k}\lambda_i\overline{X_i^2}}{2}$

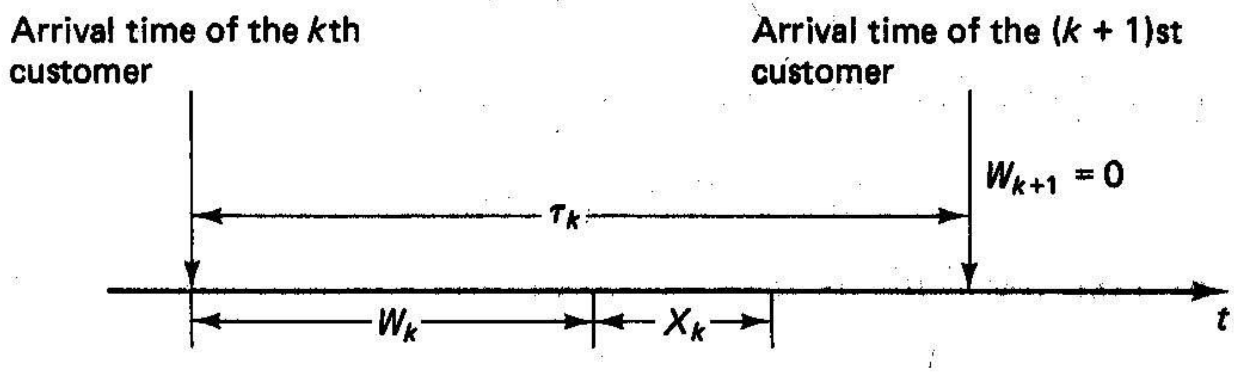
–First: ave. service time

–Second: ave. time required to service customers of priority 1 to k already in the system

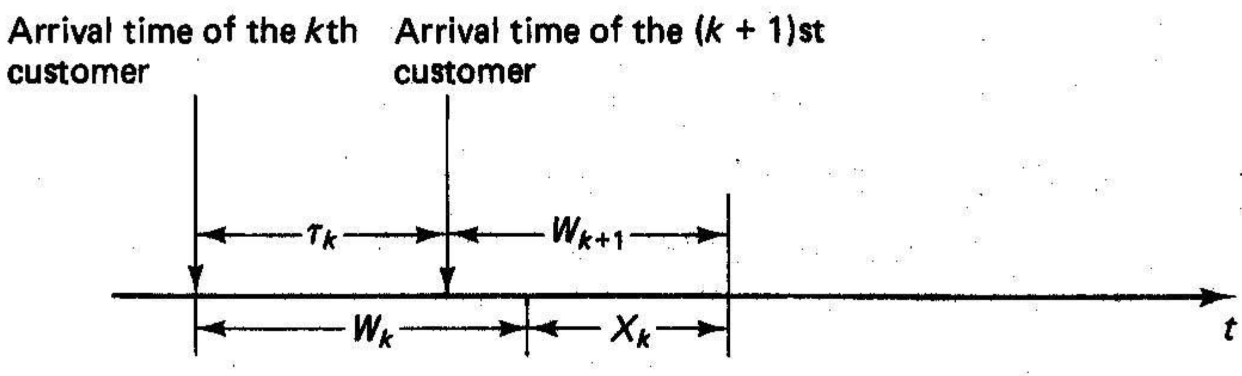
–Third: ave. waiting time for customers of priority 1 to $k-1$ who arrive while priority k customer is in the system

$$\Rightarrow T_k = \frac{(1/\mu_k)(1-\rho_1-\cdots-\rho_k)+R_k}{(1-\rho_1-\cdots-\rho_{k-1})(1-\rho_1-\cdots-\rho_k)}$$

Upper bound for the G/G/1 system



(a)



(b)

$$W_{k+1} = \max\left[0, W_k + X_k - \tau_k\right]$$

$$W \le \frac{\lambda\left(\sigma_a^2 + \sigma_b^2\right)}{2\left(1-\rho\right)}$$

where σ_a^2 : variance of the interarrival times

σ_b^2 : variance of the service times

τ_k : interarrival time between the k^{th} and $k+1^{\text{st}}$ customer

I_k : idle time between the k^{th} and $k+1^{\text{st}}$ customer

$$Y^+ = \max\{0, Y\} \qquad Y^- = -\min\{0, Y\}$$

$$Y = Y^+ - Y^- \qquad Y^+ \cdot Y^- = 0$$

$$\overline{Y} = \overline{Y^+} - \overline{Y^-} \qquad \sigma_Y^2 = \sigma_{Y^+}^2 + \sigma_{Y^-}^2 + 2\overline{Y^+} \cdot \overline{Y^-}$$

$$W_{k+1} = \left(W_k + V_k\right)^+ \qquad V_k = X_k - \tau_k$$

$$I_{k+1} = \left(W_k + V_k\right)^-$$

$$\sigma^2_{(W_k + V_k)} = \sigma^2_{(W_k + V_k)^+} + \sigma^2_{(W_k + V_k)^-} + 2\overline{\left(W_k + V_k\right)^+} \cdot \overline{\left(W_k + V_k\right)^-}$$

$$\sigma^2_{(W_k + V_k)} = \sigma^2_{W_k} + \sigma^2_{V_k} = \sigma^2_{W_k} + \sigma^2_a + \sigma^2_b \text{ since } W_k \,\&\, V_k \text{ are indep.}$$

$$\Rightarrow \sigma^2_{W_k} + \sigma^2_a + \sigma^2_b = \sigma^2_{W_{k+1}} + \sigma^2_{I_k} + 2\overline{W_{k+1}} \cdot \overline{I_k}$$

$$W = \frac{\sigma_a^2 + \sigma_b^2}{2I} - \frac{\sigma_I^2}{2I} \text{ as } t \to \infty$$

$$W = \frac{\lambda\left(\sigma_a^2 + \sigma_b^2\right)}{2(1-\rho)} - \frac{\lambda\sigma_I^2}{2(1-\rho)} \text{ since } I = (1-\rho)/\lambda$$

$$W \le \frac{\lambda\left(\sigma_a^2 + \sigma_b^2\right)}{2\left(1-\rho\right)}$$

–Bound becomes tigher as $\rho \to 1$

–*e.g.*, consider M/G/1 queue

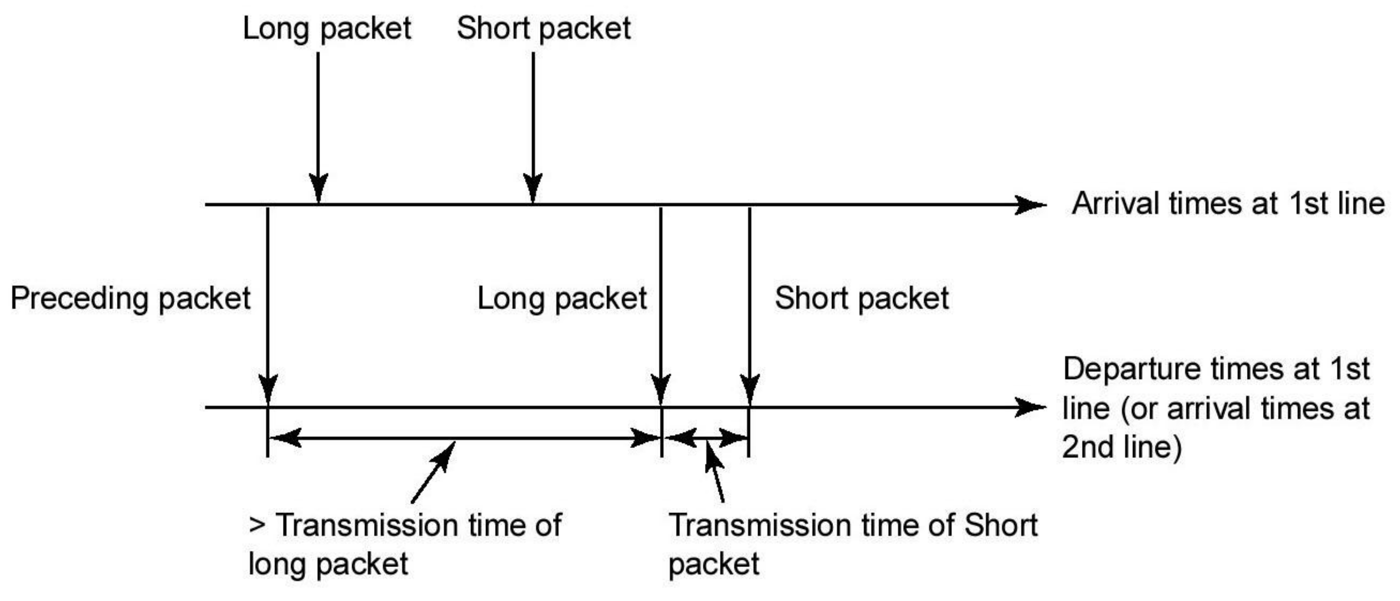
$$W = \frac{\lambda\overline{X^2}}{2\left(1-\rho\right)} = \frac{\lambda\left(\sigma_b^2 + 1/\mu^2\right)}{2\left(1-\rho\right)} \Rightarrow \sigma_I^2 = \frac{1}{\lambda^2} - \frac{1}{\mu^2}$$

Network of Transmission Lines

- Tandem queues
  - Two equal-capacity transmission lines in tandem
    - Difficult to analyze
    - Interarrival times at the 2nd queue are strongly correlated with the packet lengths
  - If all packets have equal length, i.e., M/D/1 at the 1st queue,
    - Interarrival times at the 2nd queue $\geq 1/\mu$
    - There is no queueing delay at queue 2

– If M/M/1 queue at the 1st queue, the 2nd queue cannot be M/M/1 queue

  - Interarrival time at the 2nd queue ≥ transmission time of the 2nd packet at the 1st queue



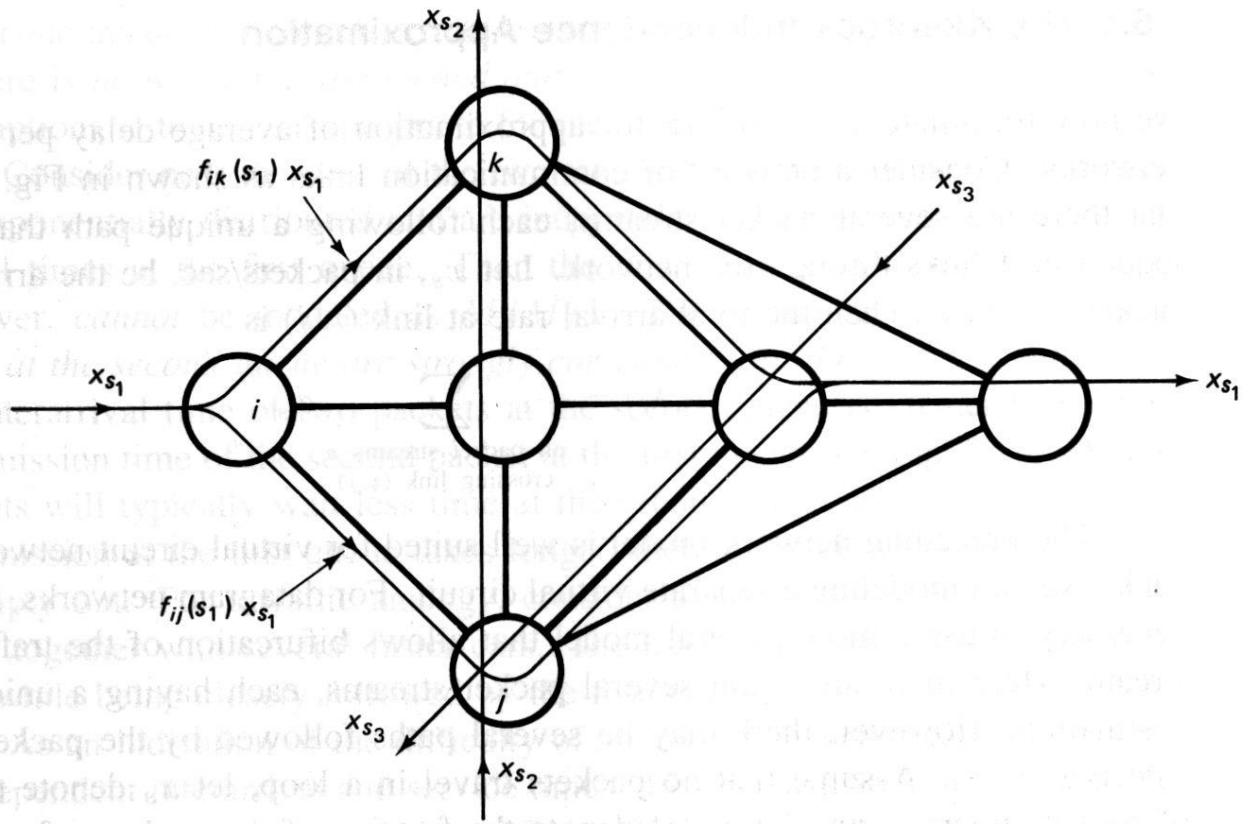


Figure 3.28 Model suitable for datagram networks. There are several packet streams, each associated with a unique origin–destination pair. However, packets of the same stream may follow one of several paths. The total arrival rate λ_{ij} at a link (i, j) is equal to the sum of the fractions $f_{ij}(s)x_s$ of the arrival rates of all packet streams s traversing

- The Kleinrock independence assumption
  - Restore the independence of interarrival times and packet length at every queue (M/M/1)
  - A reasonably good assumption for systems involving
    - Poisson stream arrivals at the entry points
    - Exponentially distributed packet lengths
    - Densely connected network
    - Moderate-to-heavy traffic loads

– At link (i,j), ave. number of packets

$$N_{ij} = \frac{\lambda_{ij}}{\mu_{ij} - \lambda_{ij}}$$

– Over all queues

$$N = \sum_{(i,j)} \frac{\lambda_{ij}}{\mu_{ij} - \lambda_{ij}}$$

$$\text{where } \lambda_{ij} = \sum_{\substack{\text{all packet streams } s \\ \text{crossing link } (i,j)}} f_{ij}(s) x_s$$

$1/\mu_{ij}$: ave. packet transmission time on link (i, j)

$f_{ij}(s)$: fraction of packets of s going through (i, j)

x_s: arrival rate of packet stream s

– From Little's theorem, the average packet delay in the network

$$T = \frac{1}{\gamma} \sum_{(i,j)} \frac{\lambda_{ij}}{\mu_{ij} - \lambda_{ij}}$$

$$\gamma = \sum_{s} x_s \text{: total arrival rate in the system}$$

– Considering ave. processing and propagation delay d_{ij} at (i,j),

$$T = \frac{1}{\gamma} \sum_{(i,j)} \left(\frac{\lambda_{ij}}{\mu_{ij} - \lambda_{ij}} + \lambda_{ij} d_{ij} \right)$$

– Ave. delay per packet over path “p” is

$$T_p = \sum_{all(i,j)\text{ on path p}} \left(\frac{\lambda_{ij}}{\mu_{ij}(\mu_{ij} - \lambda_{ij})} + \frac{1}{\mu_{ij}} + d_{ij} \right)$$

- First: ave. waiting time in queue
- Second: ave. transmission time
- Third: proc./prop. delay

– Two links between nodes A and B

- Poisson arrival and exponential transmission times
- Packet is equally divided into two links
- Example: Randomization (from HW#4, Prob. 3.11)

$$T_R = \frac{1}{\mu - \lambda/2} = \frac{2}{2\mu - \lambda}$$

- Example: Metering (works like M/M/2 queue) breaking M/M/1 approximation

$$T_M = \frac{1}{\mu} + \frac{P_Q}{2\mu - \lambda} = \frac{2}{2\mu - \lambda}\frac{1}{1+\rho}, \quad \rho = \lambda / 2\mu$$

Time Reversibility - Burke's theorem

- Birth-death process: Markov chain with integer states in that transitions occur only between neighboring states
- Consider a discrete-time Markov chain X_n, X_{n+1} ,…

$$p_j = P[X_n = j] \ \text{ for } all\ n$$

$$P[X_m = j \mid X_{m+1} = i, X_{m+2} = i_2, \cdots, X_{m+k} = i_k]$$

$$= \frac{P[X_m = j, X_{m+1} = i, X_{m+2} = i_2, \cdots, X_{m+k} = i_k]}{P[X_{m+1} = i, X_{m+2} = i_2, \cdots, X_{m+k} = i_k]}$$

$$= \frac{P[X_m = j]P[X_{m+1} = i \mid X_m = j]P[X_{m+2} = i_2, \cdots, X_{m+k} = i_k \mid X_{m+1} = i, X_m = j]}{P[X_{m+1} = i]P[X_{m+2} = i_2, \cdots, X_{m+k} = i_k \mid X_{m+1} = i]}$$

$$= \frac{p_j P_{ji}}{p_i}$$

(independent of states at times $m+2, m+3, \ldots$)

- The backward transition probability

$$P_{ij}^* = P[X_m = j \mid X_{m+1} = i] = \frac{p_j P_{ji}}{p_i} \quad i, j \geq 0$$

  - If $P_{ij}^* = P_{ij}$ for all i,j the chain is time reversible!!!

- Properties of the reversed chain (discrete time)
  - (P1) Irreducible, aperiodic, the same stationary distribution as the forward chain
    - Proof:

$$\sum_{i=0}^{\infty} p_i P_{ij}^{*} = \sum_{i=0}^{\infty} p_i \left(\frac{p_j P_{ji}}{p_i} \right) = \sum_{i=0}^{\infty} p_j P_{ji} = p_j \sum_{i=0}^{\infty} P_{ji} = p_j$$

$\therefore \{ p_i \}$ is the steady-state distribution of reverse chain!

– (P2) If there are $\{p_i\}$ s.t. $\sum_{i=0}^{\infty} p_i = 1$, and $P_{ij}^* = \frac{p_j P_{ji}}{p_i}$

form a transition prob. matrix, i.e., $\sum_{j=0}^{\infty} P_{ij}^* = 1$,

then $\{p_i\}$ is the stationary distribution and

P_{ij}^* are the transition prob. of reversed chain.

- Proof:

$$\sum_{i=0}^{\infty} p_j P_{ji} = p_i \sum_{i=0}^{\infty} P_{ij}^* = p_i$$

$\Rightarrow$ global balance eq. $\Rightarrow \{p_i\}$ is the stationary distr.

- (P3) A chain is time reversible iff

  $$p_i P_{ij} = p_j P_{ji} \quad \text{(i.e., detailed balance eq.)}$$

- (P1) & (P2) hold even if chain is not time reversible!
- Birth-death process (e.g., M/M/1, M/M/m, etc.) are time reversible

- For irreducible CTMC
  - (P1) The same stationary distribution as the forward chain with $q_{ij}^* = \dfrac{p_j q_{ji}}{p_i}$
  - (P3) The forward chain is time reversible iff

$$p_i q_{ij} = p_j q_{ji} \quad \text{(detailed balance eq.)}$$

– (P2) If there are $\{p_i\}$ s.t. $\sum_{i=0}^{\infty} p_i = 1$, and

$q_{ij}^* = \dfrac{p_j q_{ji}}{p_i}$ satisfies $\sum_{j=0}^{\infty} q_{ij} = \sum_{j=0}^{\infty} q_{ij}^*$

then $\{p_i\}$ is the stationary distribution and

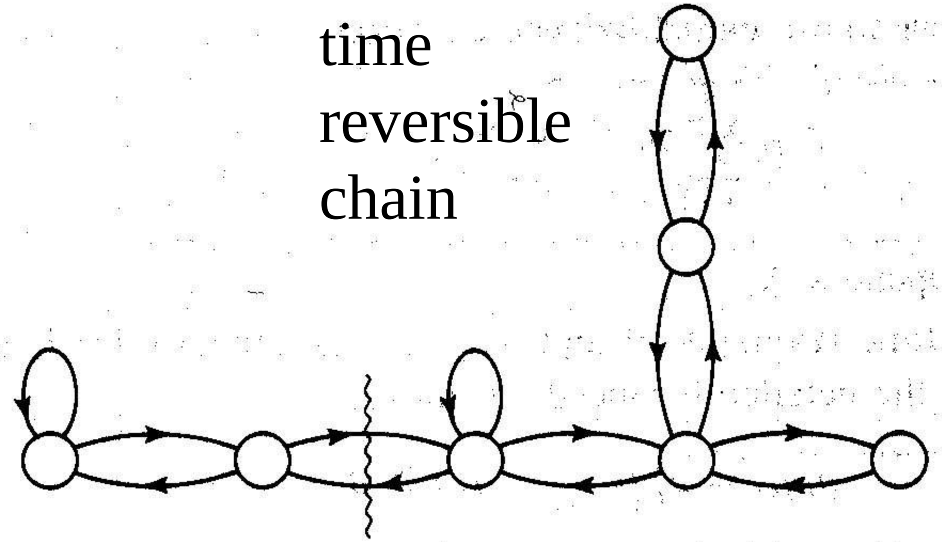
q_{ij}^* are the transition rates of reversed chain.

- Proof:

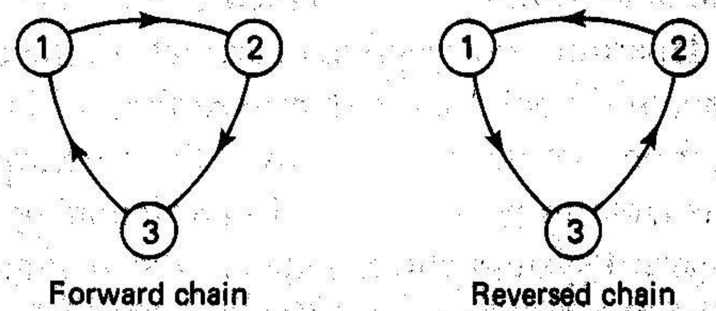
$$\sum_{i=0}^{\infty} p_j q_{ji} = p_i \sum_{i=0}^{\infty} q_{ij}^* = p_i \sum_{i=0}^{\infty} q_{ij}$$

$\Rightarrow$ global balance eq. $\Rightarrow \{p_i\}$ is the stationary distr.

time
reversible
chain

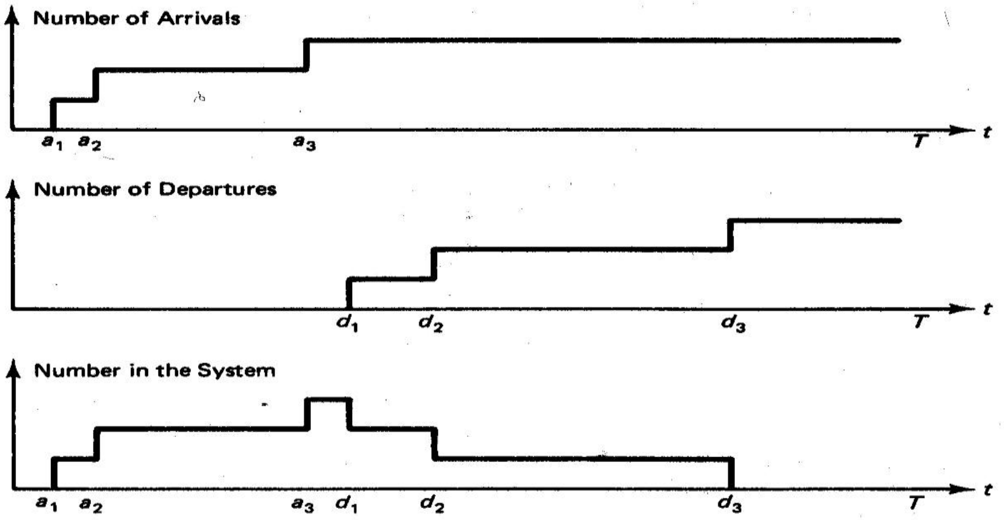


(a)

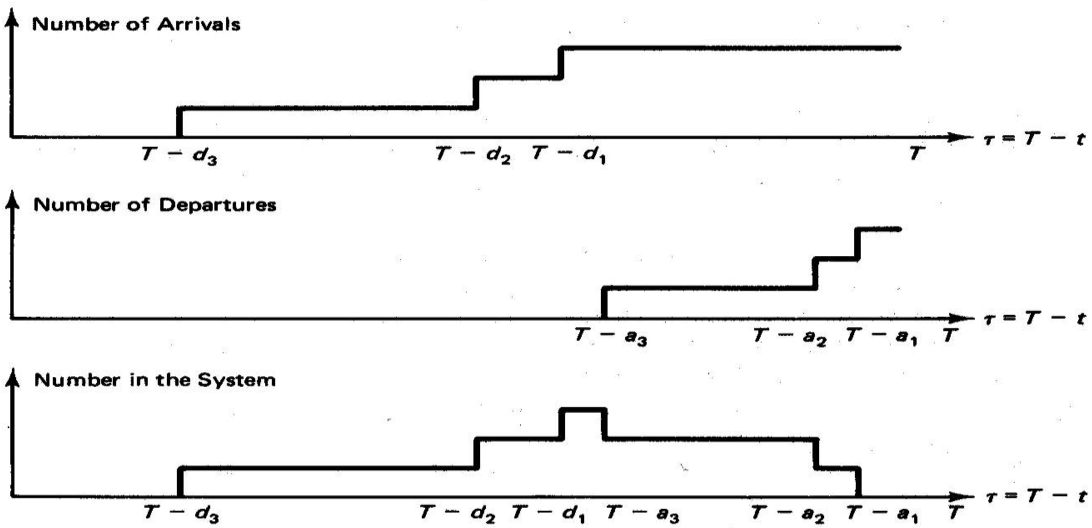


(b)

Non-time
reversible
chain



(a)



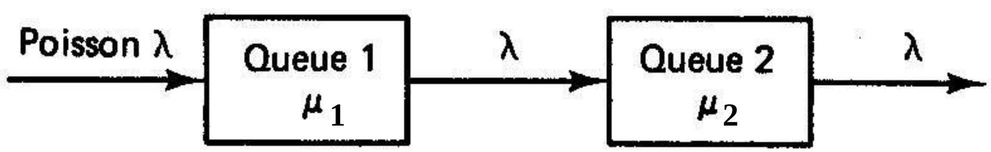
(b)

In steady-state, forward and reverse systems are statistically indistinguishable!

- **Burke's theorem** : Consider an M/M/1, M/M/m, M/M/∞ system with arrival rate λ. Suppose that the system starts in steady-state. Then the following is true:
  - The departure process is Poisson with rate λ.
  - At each time t, the number of customers in the system is independent of the sequence of departure times prior to t.

EX) Two M/M/1 queues in tandem

- Poisson arrival and exp. service time
- Assume that the service times of a customer at 1st and 2nd queues are mutually independent and independent of arrival process.



$$P[n \text{ at queue } 1,\, m \text{ at queue } 2]$$

$$= P[n \text{ at queue } 1]\, P[m \text{ at queue } 2] = \rho_1^n (1-\rho_1)\rho_2^m (1-\rho_2)$$

Networks of Queues – Jackson's theorem

- Each queue is assumed to be M/M/1
- K queues
- P_{ij}: transition prob. from queues i to j
- r_j: external input rate at queue j
- λ_j : total arrival rate at queue j

$$\lambda_j = r_j + \sum_{i=1}^{K} \lambda_i \ P_{ij} \qquad j = 1, \ \cdots, \ K$$

- vector denoting the number of customers

$$n = (n_1,\ n_2,\ \cdots,\ n_K)$$

- state transition rate (Continuous Time MC)

Arrival $\quad q_{nn(j^+)} = r_j$

Departure $\quad q_{nn(j^-)} = \mu_j (1 - \sum_i P_{ji})$

Transition $\quad q_{nn(i^+, j^-)} = \mu_j\ P_{ji}$

- Theorem: Assume $\rho_j < 1$ and let the stationary distribution of the chain

$P(n_1, \cdots, n_k)$ Then

$$P(n) = P_1(n_1)\, P_2(n_2) \cdots P(n_k)$$

where $P_j(n_j) = \rho_j^{\,n_j}(1-\rho_j) \qquad n_j \ge 0$

- Proof: using (P2) of reverse chain of CTMC, we show that with *P(n)* defined above,

$$q_{nn'}^{*} = \frac{P(n')q_{n'n}}{P(n)} \text{ satisfies } \sum_m q_{nm} = \sum_m q_{nm}^{*}$$

$$q_{nm} = \begin{cases} r_j, & m = n(j^+) \\ \mu_j \left(1 - \sum_i P_{ji}\right), & m = n(j^-) \\ \mu_j P_{ji}, & m = n(i^+, j^-) \\ 0, & \text{otherwise} \end{cases}$$

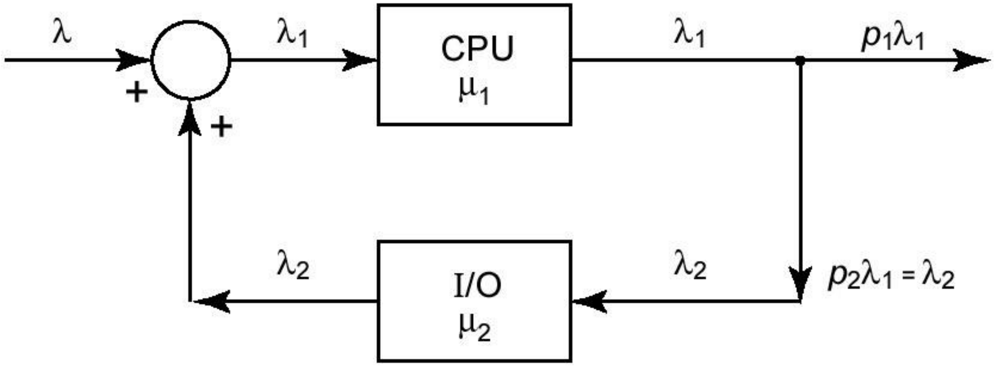
$$q_{nm}^{*} = \begin{cases} \lambda_j \left(1 - \sum_i P_{ji}\right), & m = n(j^{+}) \\ \dfrac{\mu_j r_j}{\lambda_j}, & m = n(j^{-}) \\ \dfrac{\mu_j \lambda_i P_{ij}}{\lambda_j}, & m = n(i^{+}, j^{-}) \\ 0, & \text{otherwise} \end{cases}$$

$$\Rightarrow \sum_m q_{nm} = \sum_{j=1}^{K} r_j + \sum_{\{j \mid n_j > 0\}} \mu_j = \sum_m q_{nm}^{*}$$

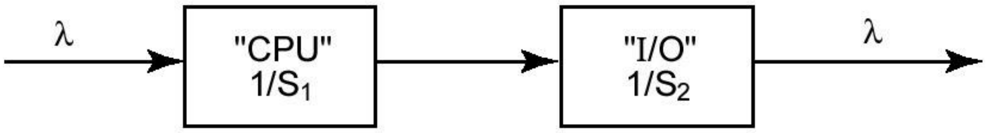
Q.E.D.

Network of Queues

- Open queueing networks



(a)



(b)

– Service times at CPU and I/O are independent

$$\lambda_1 = \lambda + \lambda_2, \quad \lambda_2 = p_2 \lambda_1$$

– where $p_1 + p_2 = 1, \;\; \rho_1 = \lambda_1 / \mu_1, \;\; \rho_2 = \lambda_2 / \mu_2$

Then $\lambda_1 = \lambda / p_1, \; \lambda_2 = \lambda p_2 / p_1$

By Jackson's theorem

$$P[n_1, n_2] = \rho_1^{n_1}(1-\rho_1)\rho_2^{n_2}(1-\rho_2)$$

$$N_1 = \frac{\rho_1}{1-\rho_1}, N_2 = \frac{\rho_2}{1-\rho_2} \qquad N = N_1 + N_2$$

$$T = \frac{N}{\lambda} = \frac{\rho_1}{\lambda(1-\rho_1)} + \frac{\rho_2}{\lambda(1-\rho_2)}$$

– For equivalent tandem model

$$T = \frac{\rho_1}{\lambda(1-\rho_1)} + \frac{\rho_2}{\lambda(1-\rho_2)}$$

$$= \frac{\lambda/(\mu_1 p_1)}{\lambda(1-\lambda/(\mu_1 p_1))} + \frac{\lambda p_2/(\mu_2 p_1)}{\lambda(1-\lambda p_2/(\mu_2 p_1))}$$

$$= \frac{S_1}{1-\lambda S_1} + \frac{S_2}{1-\lambda S_2} \quad \left(\text{cf. } T = \frac{1}{\mu-\lambda} \text{ in M/M/1 queue}\right)$$

– where $S_1 = \dfrac{1}{\mu_1 p_1}, S_2 = \dfrac{p_2}{\mu_2 p_1}$ avg. serv. time

Note: Equivalent model can't be used for the probability density function representing the number of customers at each queue.

Extensions of Jackson's theorem

- State dependent service rates (e.g., multi servers)
  - Define
    $$\rho_j(m) = \frac{\lambda_j}{\mu_j(m)} \quad j = 1, \cdots, K$$
    $$m = 1, 2, \cdots$$
  - Define
    $$\hat{p}_j(n_j) = \begin{cases} 1 & \text{if } n_j = 0 \\ \rho_j(1)\rho_j(2)\cdots \rho_j(n_j) & \text{if } n_j > 0 \end{cases}$$
  - Then
    $$p(n) = \frac{\hat{p}_1(n_1)\cdots \hat{p}_k(n_K)}{G}$$
    where $\quad G = \sum_{n_1=0}^{\infty} \cdots \sum_{n_K=0}^{\infty} \hat{p}_1(n_1)\cdots \hat{p}_K(n_K)$

Multiple classes of customers

– Classes of customers

– $\lambda_j(c) = r_j(c) + \sum_{j=1}^{K} \lambda_i(c)\, P_{ij}(c)$

– Class composition of queue j (total of n_j cust.)

$$Z_j = (c_1, c_2, \cdots, c_{n_j})$$

– Network state at a given time

$$Z = (z_1,\ z_2,\ \cdots,\ z_K)$$

$$\hat{\rho}_j(c,m) = \lambda_j(c) / \mu_j(m)$$

– Define
$$\hat{p}_j(z_j) = \begin{cases} 1 & \text{if } n_j = 0 \\ \hat{\rho}_j(c_1,1)\hat{\rho}_j(c_2,2)\cdots\ \hat{\rho}_j(c_{n_j}, n_j) & \text{if } n_j > 0 \end{cases}$$

where $G = \sum_{(z_1,\cdots,z_k)} \prod_{j=1}^{K} \hat{\rho}_j(z_j)$

- Then $$\hat{p}(z) = \frac{\hat{p}_1(z)\cdots\hat{p}_K(z_K)}{G}$$
  - $$p(n) = p(n_1,\cdots,n_k) = \sum_{z\in z(n)} \hat{p}(z)$$

    Where $Z(n)$ is the set of states with n_j customers at queue j
  - if $\mu_j(m) = \mu_j,$ the same form as in the basic Jackson's Theorem

    $$p(n) = \prod_{j=1}^{K} \rho_j^{\,n_j}(1-\rho_j)$$

    where $$\rho_j = \frac{\sum_{c=1}^{C} \lambda_j(c)}{\mu_j}$$

Closed Queueing Networks

– No customers are allowed to arrive or depart
– M: the number of customers in the system

$$\sum_{j=1}^{K} p_{ij} = 1 \qquad i = 1, \cdots, K$$

– $\mu_j(m)$: the state dependent service rate at queue j when there are m customers

$\lambda_j(m)$: the total arrival rates (need to be found)

$$\lambda_j = \sum_{i=1}^{K} \lambda_i \ p_{ij} \qquad j = 1, \cdots, K$$

$$\lambda_j(m) = \alpha(m)\overline{\lambda_j}$$

where $\alpha(m)$: the constant related with M

$\overline{\lambda}_j$: a particular solution

Let $\rho_j(m) = \dfrac{\overline{\lambda_j}}{\mu_j(m)}$

$$\hat{P}_j(n_j) = \begin{cases} 1 & \text{if } \; n_j = 0 \\ \rho_j(1)\rho_j(2)\cdots\rho_j(n_j) & \text{if } \; n_j > 0 \end{cases}$$

Closed queueing networks

– states for the network of K queues: **n**=(n_1,n_2,…,n_K)

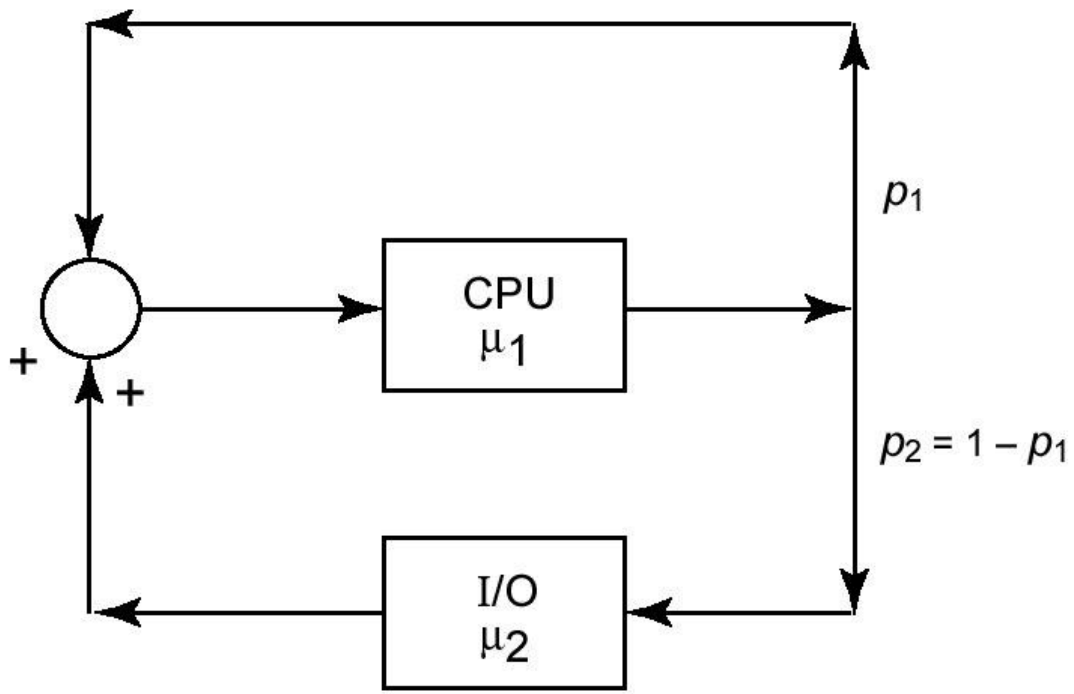
– total of *M* customers in the system:

$$P[n] = \frac{\hat{P}_1(n_1)\cdots\hat{P}_K(n_K)}{G(M)}$$

$$G(M) = \sum_{\{(n_1,\cdots,n_K) | n_1+\cdots+n_K=M\}} \hat{P}_1(n_1)\cdots\hat{P}_K(n_K)$$

where G(M) is a normalization constant that ensures that P[**n**] is a probability distribution

- Example (closed queueing networks)



– M jobs in the system

Let $\bar{\lambda}_1 = \mu_1, \bar{\lambda}_2 = p_2 \mu_1$ then $\rho_1 = 1, \quad \rho_2 = \dfrac{p_2 \mu_1}{\mu_2}$

$$P(M-n, n) = \frac{1^{M-n} \rho_2^n}{G(M)} \quad , n = 0, 1, \cdots, M$$

where $G(M) = \sum_{n=0}^{M} \rho_2^n$

– The CPU utilization factor is

$$U(M) = 1 - P(0, M) = 1 - \frac{\rho_2^M}{G(M)} = \frac{G(M-1)}{G(M)}$$

Mean value analysis

- Obtain $N_j(M)$ and $T_j(M)$ iteratively for a closed queueing network with M customers

  $N_j(M)$: Avg. num. of customers in queue j

  $T_j(M)$: Avg. customer time in queue j

- By Little's theorem for queue j

$$\lambda_j(m) = \frac{N_j(m)}{T_j(m)}$$

start with $T_j(0) = N_j(0) = 0 \quad j = 1, \cdots, K$

$$T_j(s) = \frac{1}{\mu_j}\left(1 + N_j(s-1)\right) \qquad s = 1, \cdots, M$$

$$N_j(s) = s \frac{\overline{\lambda_j}\, T_j(s)}{\sum_{i=1}^{K} \overline{\lambda_i}\, T_i(s)}$$

where $\overline{\lambda_j}$ is a positive solution of

$\lambda_j = \sum_{i=1}^{K} \lambda_i \, p_{ij}$ regardless of M.

K equations with no external inputs will have many solutions.

- To derive previous *T* and *N* for some scalar $\alpha(s) > 0$
- As $\lambda_j(s) = \alpha(s)\bar{\lambda}_j$ and $s = \sum_{i=1}^{K} N_i(s)$ and $N_i(s) = \lambda_i(s)T_i(s)$

$$N_j(s) = s\frac{\lambda_i(s)T_i(s)}{\sum_{i=1}^{K}\lambda_i(s)T_i(s)}$$

- *Arrival Theorem*: The occupancy distribution found by a customer upon arrival at queue *j* is the same as the steady-state distribution with the arriving customer removed

- Proof:

$$\alpha_{ij} = P\{x(t) = n \mid \text{a customer moved from queue } i \text{ to } j \text{ just after } t\}$$

$$= \frac{P\{x(t) = n, M_{ij}(t) \mid M_i(t)\}}{P\{M_{ij}(t) \mid M_i(t)\}}$$

$$= \frac{P\{x(t) = n \mid M_i(t)\} P\{M_{ij}(t) \mid x(t) = n, M_i(t)\}}{P\{M_{ij}(t) \mid M_i(t)\}}$$

$$= \frac{P(n) P_{ij}}{\sum_{\{n' = (n'_1, \ldots, n'_K) \mid n'_i > 0\}} P(n') P_{ij}}$$

$$= \frac{\hat{P}_1(n_1) \cdots \hat{P}_K(n_K)}{\sum_{\{(n'_1, \ldots, n'_K) \mid n'_1 + \ldots + n'_K = s, n'_i > 0\}} \hat{P}_1(n'_1) \cdots \hat{P}_K(n'_K)}$$

$$= \frac{\hat{P}_1(n_1) \cdots \hat{P}_{i-1}(n_{i-1}) \hat{P}_i(n_i - 1) \hat{P}_{i+1}(n_{i+1}) \cdots \hat{P}_K(n_K)}{\sum_{\{(n'_1, \ldots, n'_K) \mid n'_1 + \ldots + n'_K = s-1\}} \hat{P}_1(n'_1) \cdots \hat{P}_K(n'_K)}$$

Multiaccess Communication

Sunghyun Choi

Adopted from Prof. Saewoong Bahk’s material

Channel Allocation Problem

- In any broadcast network, the key issue is how to determine who gets to use the channel
- MAC: medium access control
  - Between the data link layer and the physical layer
  - Allocates the multi-access medium among the competing nodes(hosts)

- Channel allocation
  - Free for all or perfectly scheduled approach
- Multi-access channels
  - Satellite channels
  - Multidrop telephone lines
  - Multitapped bus
  - Packet radio networks

Idealized Slotted Multiaccess Model

- *m* transmitting nodes and one receiver
- Slotted system
  - Perfect synchronization and same packet length
- Poisson arrivals with rate λ/m individually
- Collision or perfect reception
  - no channel error or capture effect
- 0, 1, e(rror) and immediate feedback

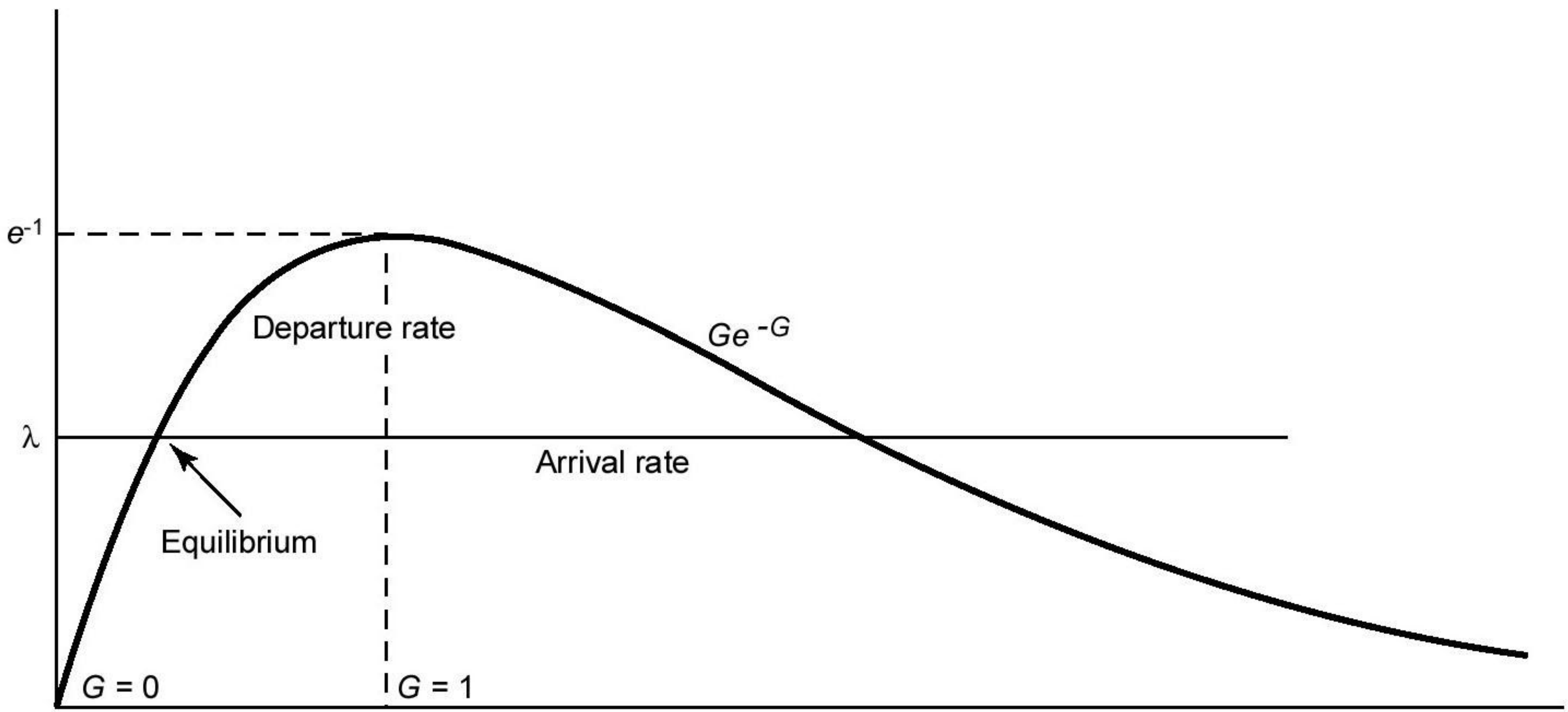
- Retransmission of collisions
  - *Backlogged* node is with a packet to be retransmitted
- Either of the following two assumptions
  - No buffering: packet arriving at active node is simply discarded
  - Infinite set of nodes ($m = \infty$): packet always arrives at a new node

Slotted ALOHA

- Every node tries to transmit packet of the same length only at the beginning of a slot.
- Unbacklogged node simply transmit a new packet in the first slot after the arrival.
  - Achieves shorter delay in lightly loaded case
  - Cf. TDM w/ ave. delay = $m/2$ slots
- Backlogged nodes wait for the random amount of time to transmit again.

- Assuming infinite number of nodes and retransmissions are well randomized
- New transmissions + retransmissions are approximated by Poisson
  - G: combined arrival rate in a frame time unit (retransmission + new transmission λ)
- From the Poisson pmf, the probability of a successful transmission in a slot = Ge^{-G}

- At equilibrium, arrival rate λ should be the same as the departure rate Ge^{-G}
- Maximum at G=1, throughput = $e^{-1} \approx 0.368$
- Otherwise, two values of G result in the same departure rate!
- $G < 1$, too many idle slots
- $G > 1$, too many collisions



e^{-1}
Departure rate
Ge^{-G}
λ
Arrival rate
Equilibrium
$G = 0$
$G = 1$

- Precise model (Discrete Time Markov chain)
  - q_r : the prob. that a backlogged node retransmits
  - n : number of backlogged nodes
  - m : total number of nodes (assuming no buffering)
  - q_a: the prob. that an unbacklogged node transmits a packet in a given slot $(= 1 - e^{-\lambda/m})$.
  - λ: total new arrival rate from m nodes

– $Q_a(i,n)$: the prob. that i unbacklogged nodes transmit packets in a given slot

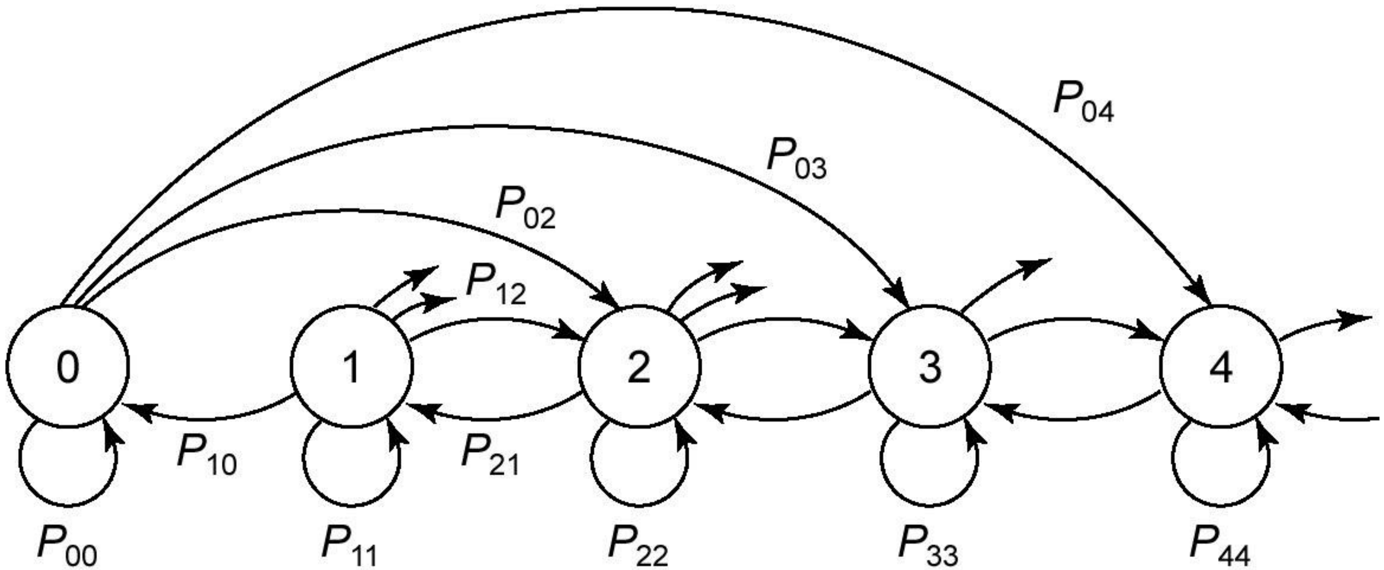
$$Q_a(i,n) = {}_{m-n}C_i (1-q_a)^{m-n-i} q_a^i$$

– $Q_r(i,n)$: the prob. that i backlogged nodes transmit packets in a given slot

$$Q_r(i,n) = {}_nC_i (1-q_r)^{n-i} q_r^i$$

– state: number of backlogged packets

– transition probability from state n to $n+i$



P04
P03
P02
P12
0
1
2
3
4
P10
P21
P00
P11
P22
P33
P44

$$
\begin{aligned}
P_{n,n+i} &= Q_a(i,n) && 2 \le i \le (m-n) \\
&= Q_a(1,n)[1-Q_r(0,n)] && i = 1 \\
&= Q_a(1,n)Q_r(0,n) + Q_a(0,n)[1-Q_r(1,n)] && i = 0 \\
&= Q_a(0,n)Q_r(1,n) && i = -1
\end{aligned}
$$

- decrease by at most one per transition, increase by an arbitrary amount.

- Analysis procedures
  - find p_n, p_0

– find the expected number of backlogged nodes.

– from Little's theorem, find the average delay.

- Characteristics
  - if $q_r n >> 1$, many collisions occur and the system is heavily backlogged for a long time.
  - drift : expected backlog change over one slot at state n, $D_n = (m-n)q_a - P_{succ}$

(i.e., new arrival - departure)

where $P_{succ} = Q_a(1,n)Q_r(0,n) + Q_a(0,n)Q_r(1,n)$

– total attempt rate

$$G(n) = (m-n)q_a + nq_r$$

– For small q_a and q_r, approximation of

$$P_{succ} = \left(\frac{(m-n)q_a}{1-q_a} + \frac{nq_r}{1-q_r} \right)(1-q_a)^{m-n}(1-q_r)^n$$

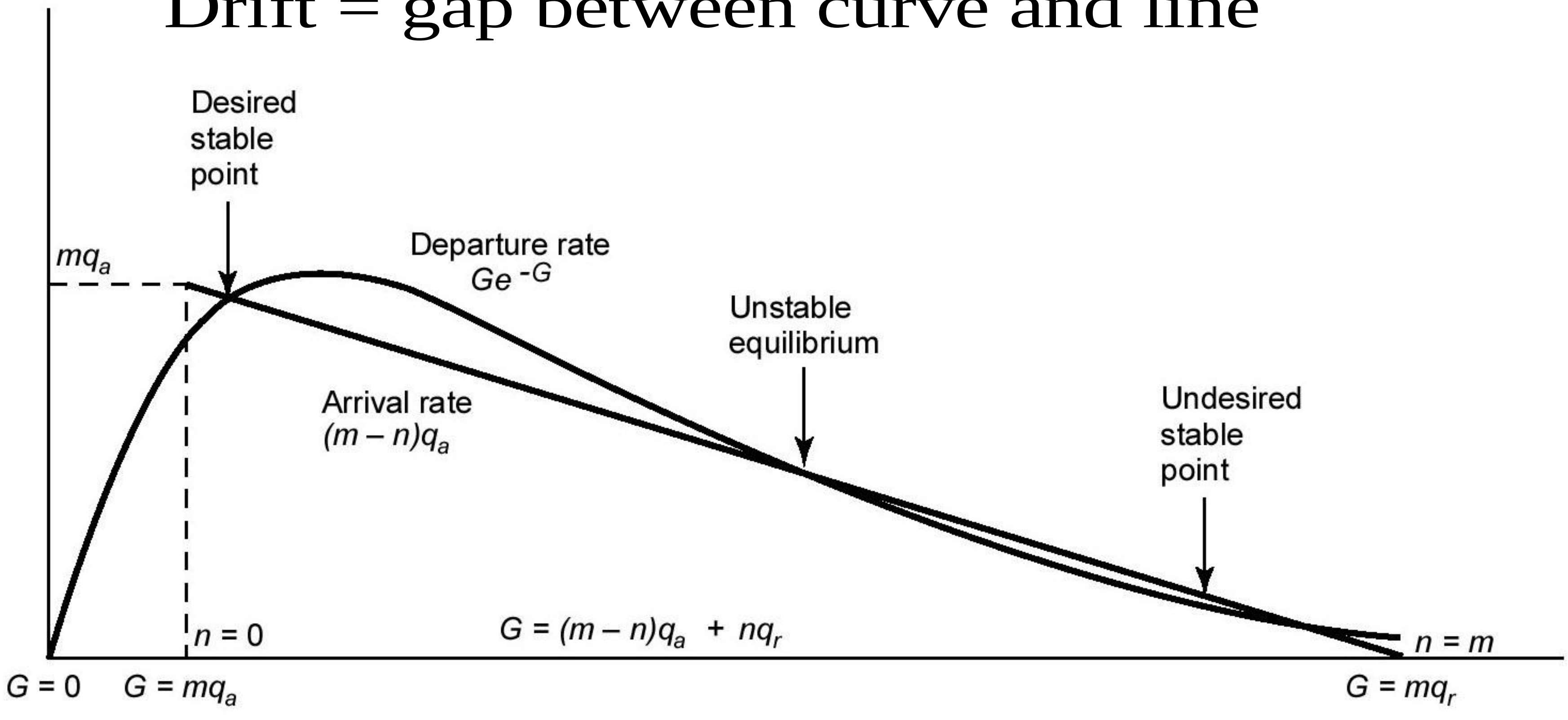
$$\approx G(n)e^{-q_a(m-n)}e^{-q_r n} = G(n)e^{-G(n)} \text{ (i.e., Poisson distr.)}$$

$$\because (1-x)^y \approx e^{-xy} \text{ for small } x$$

– maximum throughput $1/e$ at G=1

When $q_r > q_a$ (the only case of interest)

Drift = gap between curve and line



- Three equilibrium points
  - desired stable point
  - unstable equilibrium
  - undesired stable point (bad performance)
- For very small q_r, there is only a single equilibrium point, i.e., desired stable point.

– Now, with the infinite-node assumption

$$G(n) = \lambda + nq_r \qquad \text{(horizontal line)}$$

– Undesirable stable point disappears

– Once the state passes the unstable equilibrium, it tends to increase without bound!

– Infinite MC has no steady-state distri.

Stabilized Slotted ALOHA

- Maintain the attempt rate G*(n)* at 1.
  - when an idle slot occurs, increase q_r
  - when a collision occurs, decrease q_r
  - Success, idle, collision probabilities = 1/*e*, 1/*e*, 1- 2/*e*
- Pseudo-Bayesian algorithm
  - new arrivals are regarded as backlogged immediately on arrival
  - $G(n) = nq_r$ (*n* backlogged packets including new arrivals)

– prob. of a successful transmission

$$ {}_nC_1 q_r (1-q_r)^{n-1} = n q_r (1-q_r)^{n-1} $$

– maintain an estimate $\hat{n}$ of the backlog at the beginning of each slot.

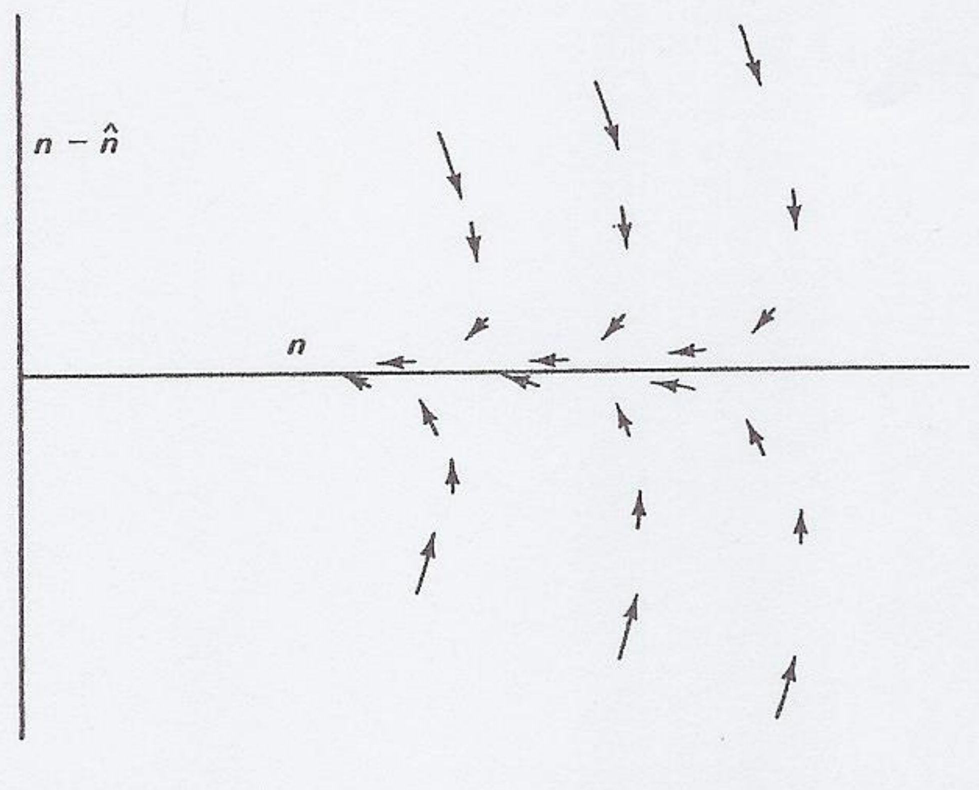
– $$ q_r(\hat{n}) = \min\left[1, \frac{1}{\hat{n}}\right] $$

– $$ \hat{n}_{k+1} = \max[\lambda, \hat{n}_k + \lambda - 1] \qquad \text{for idle or success} $$

$$ \hat{n}_k + \lambda + (e-2)^{-1} \qquad \text{for collision} $$

– updated at the beginning of slot (k+1)
– for idle, decrease $\hat{n}$ by 1 to improve performance

- Stability of pseudo-Bayesian algorithm
  - It is stable for all $\lambda < 1/e$
- Drift of n and $n - \hat{n}$



- Approximate delay analysis
  - W_i: delay from the arrival of i-th packet until the beginning of the i-th successful tx.
  - From Prob. 3.32 (M/G/1 system w/ arbitrary order of service)
    - P-K formula for delay W is valid independent of the service order provided that relative order of arrival of the customer chosen is independent of the service times

- When there are n_i backlogged nodes upon arrival of *i*-th packet

$$W_i = R_i + \sum_{j=1}^{n_i} t_j + y_i$$

$$W = 1/2 + \lambda e W + E\{y\}$$

$$E[t_j] = e \text{ since succ. slot prob.} \approx 1/e$$

when there are at least

2 backlogged nodes

$$E[n_i] = \lambda W \text{ due to Little's theorem}$$

- When p_n is the prob. of n backlogged nodes,

$$E\{y\} = (e-1)\left(1 - \frac{p_1}{\lambda}\right) + 0 \cdot \frac{p_1}{\lambda}$$

$$\lambda = p_1 + (1 - p_1 - p_0)/e$$

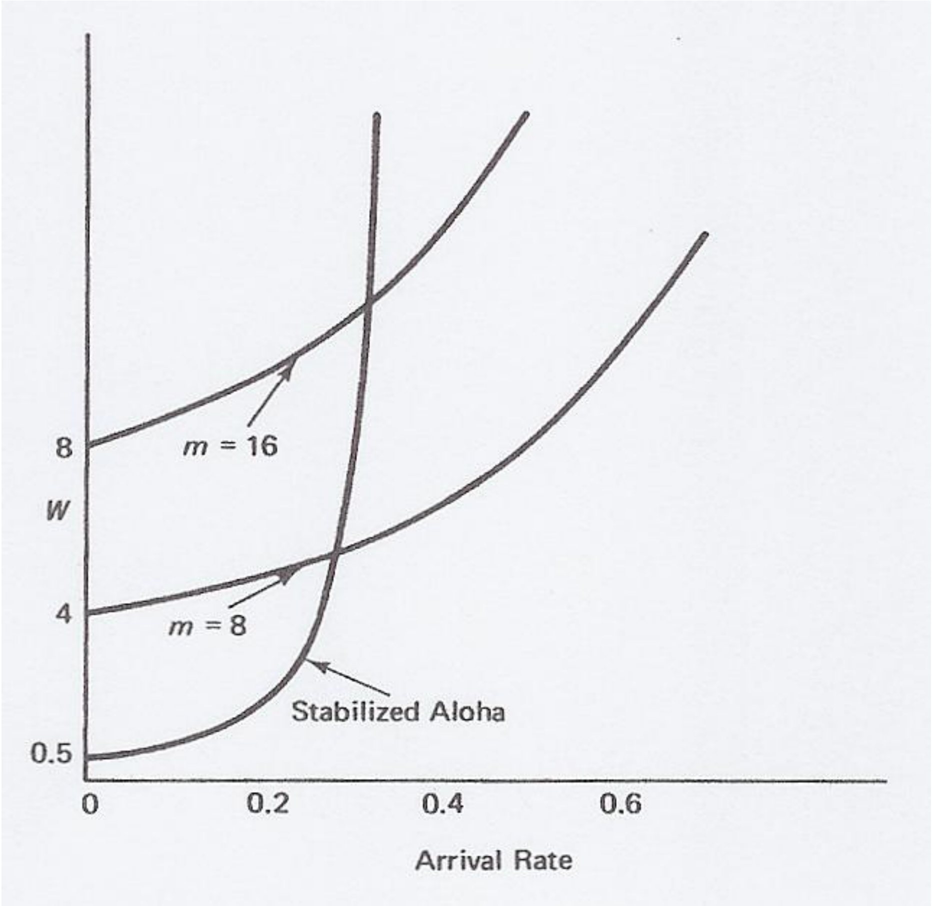
$$p_0 = (p_0 + p_1)e^{-\lambda};\ \text{no packet arrives}$$

$$\text{when 0 or 1 backlogged node}$$

$$p_1 = \frac{(1-\lambda e)(e^{\lambda} - 1)}{1-(e-1)(e^{\lambda} - 1)}$$

$$W = \frac{e - 1/2}{1-\lambda e} - \frac{(e-1)(e^{\lambda} - 1)}{\lambda[1-(e-1)(e^{\lambda} - 1)]}$$

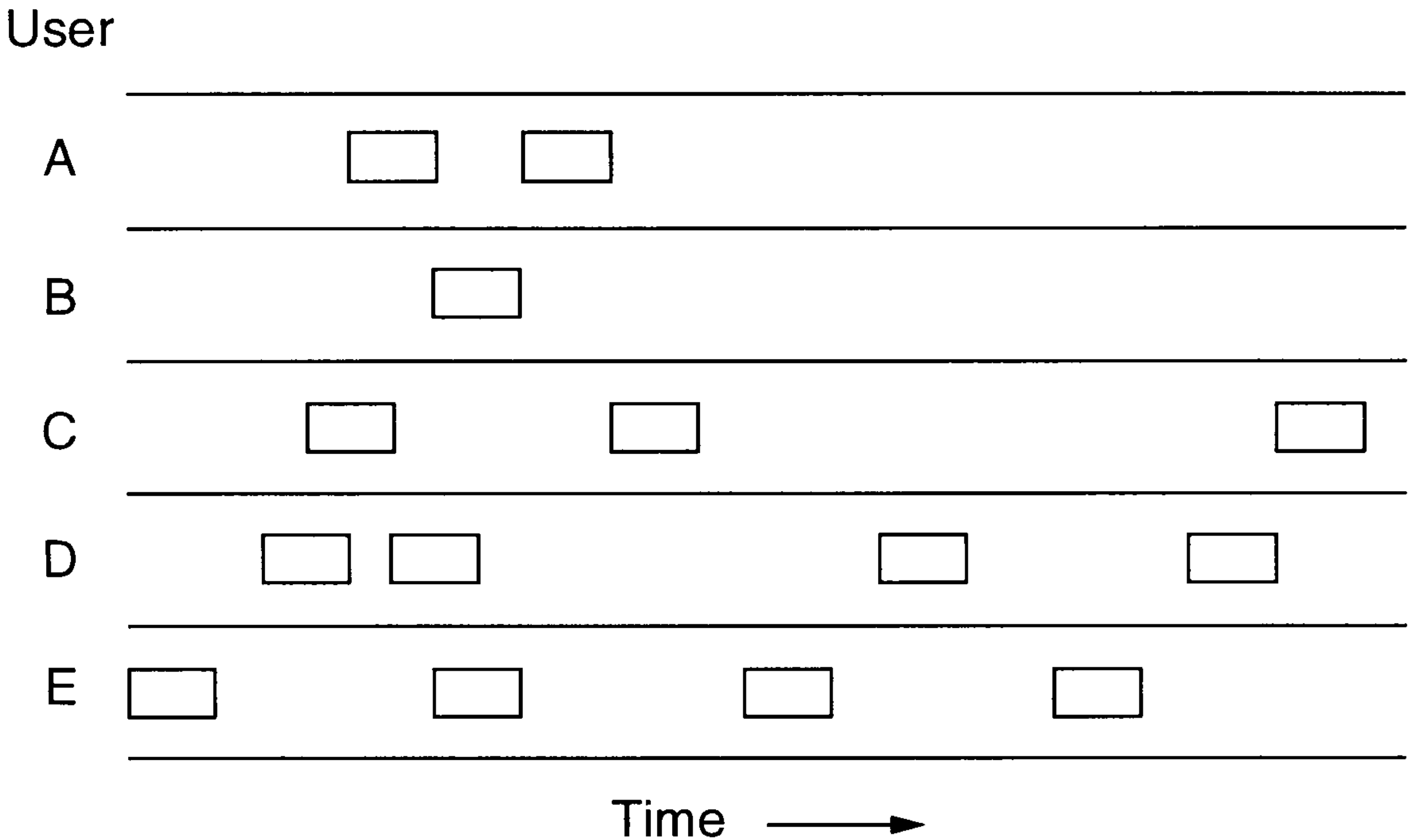
- Comparison to TDM w/ *m* slots



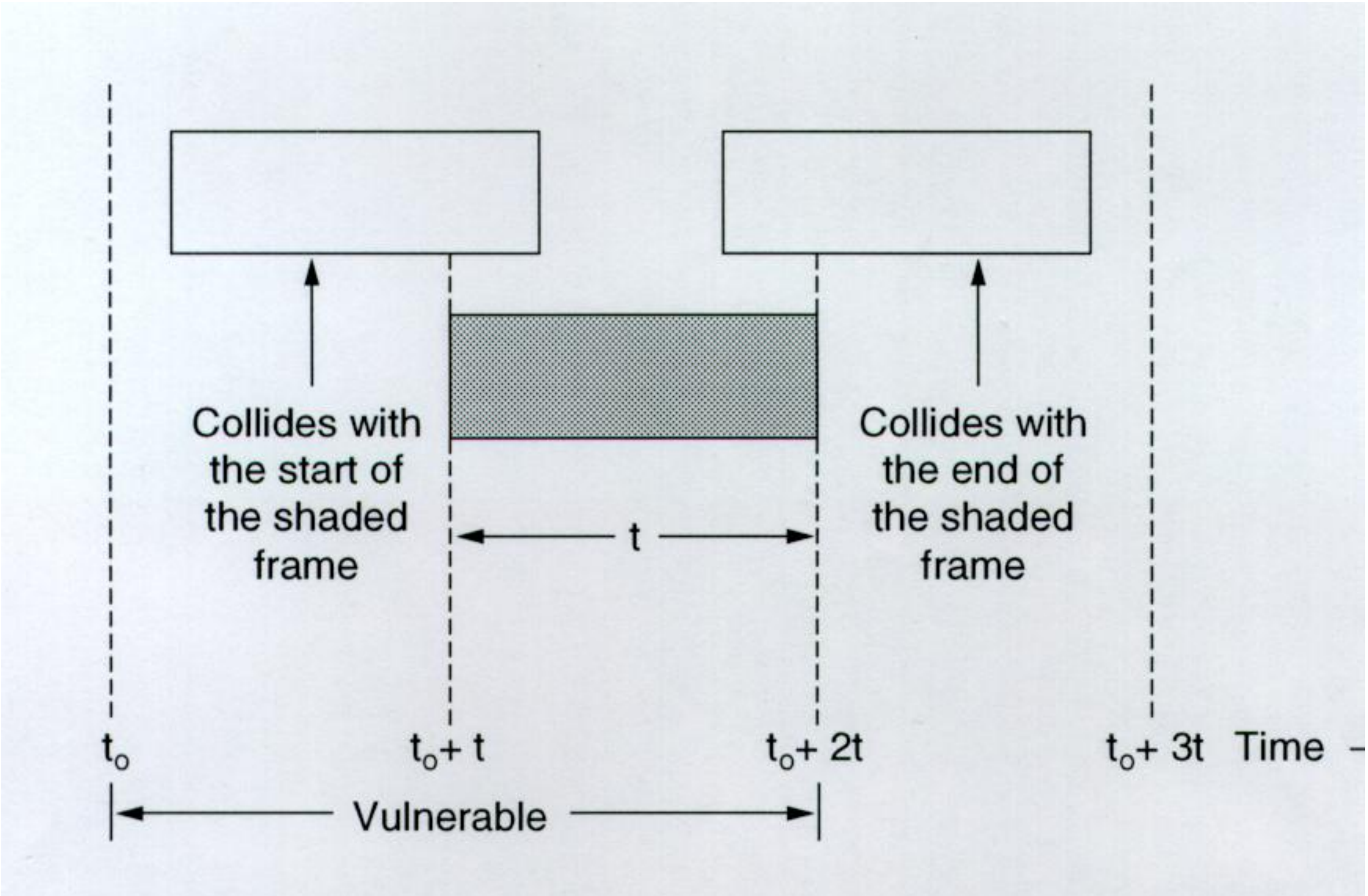
- Binary exponential backoff
  - Practically there is only ACK, not feedback in terms of 0, 1, e
    - Insufficient for pseudo-Bayesian algorithm
  - Another practical stabilization strategy employed by Ethernet
    - After *i* unsuccessful tx, $q_r = 2^{-I}$
  - For infinite nodes, this strategy is unstable for every λ

Pure ALOHA

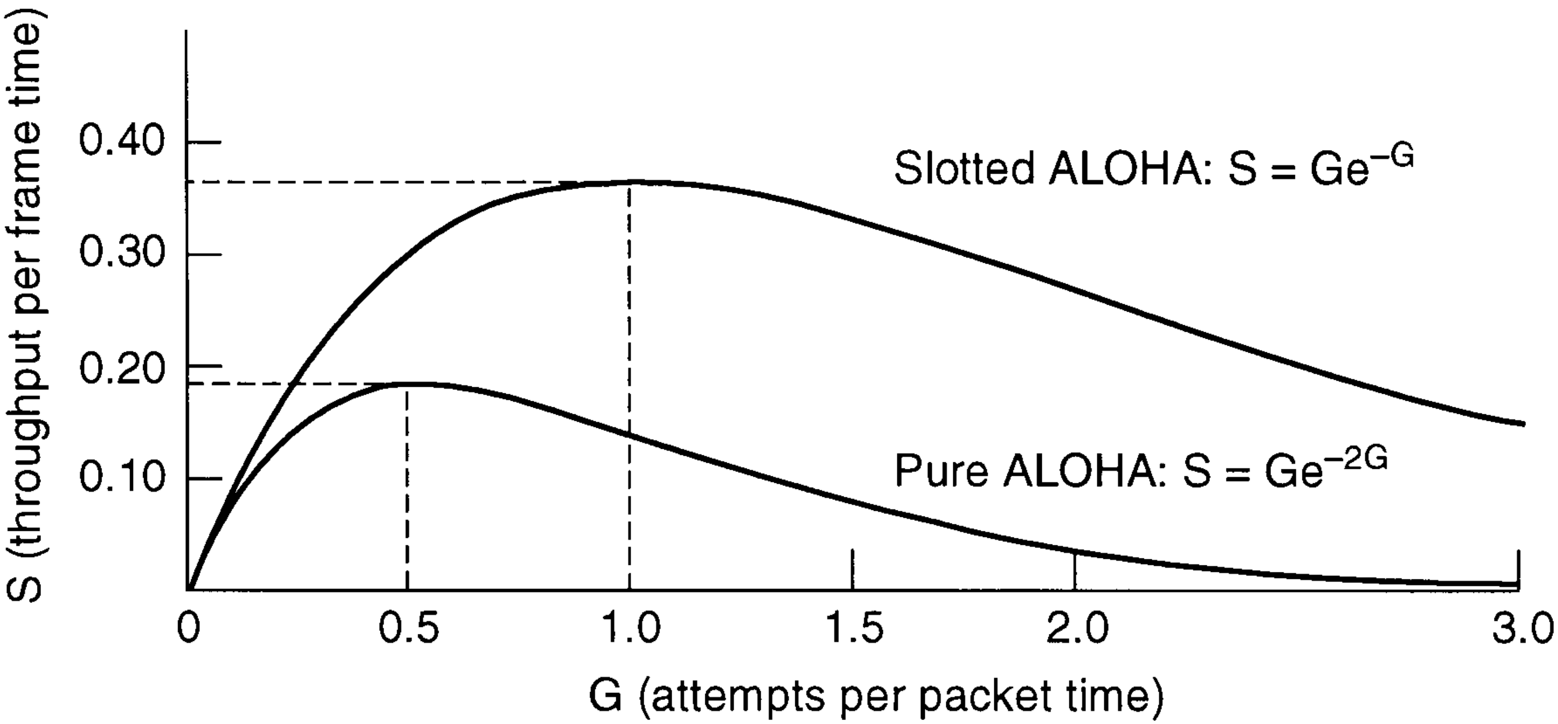
- Assumptions
  - Poisson arrivals: rate of λ
  - Collision: two or more nodes send a packet in an overlapped manner
  - Feedback: 0,1,e(rror) from the receiver
  - Retransmission: A node with a packet that must be retransmitted is called backlogged.
  - Buffering: A node is buffering new arrivals at that node.


User
A
B
C
D
E
Time

- New packets are transmitted immediately.
- Unsuccessful transmissions are repeated after a random delay.
- Throughput
  - X : a node's retransmission attempt rate
  - $xe^{-x\tau}$: Attempted retransmission period
  - $G(n) = \lambda + nx$: Time varying Poisson rate when n of backlog nodes


Collides with the start of the shaded frame
t
Collides with the end of the shaded frame
t_o
$t_o + t$
$t_o + 2t$
$t_o + 3t$
Time
Vulnerable

- $P_{succc} = e^{-2G(n)}$: 0 arrivals for two frame time
- Throughput = $G(n)e^{-2G(n)}$
- Advantage of pure ALOHA is no constraint on the packet length



S (throughput per frame time)
0.40
0.30
0.20
0.10
Slotted ALOHA: S = Ge^{–G}
Pure ALOHA: S = Ge^{–2G}
0
0.5
1.0
1.5
2.0
3.0
G (attempts per packet time)

Splitting Algorithms

- With small attempt rates, most collisions should be between two nodes
- To resolve collision of two users with each trans. prob. of ½

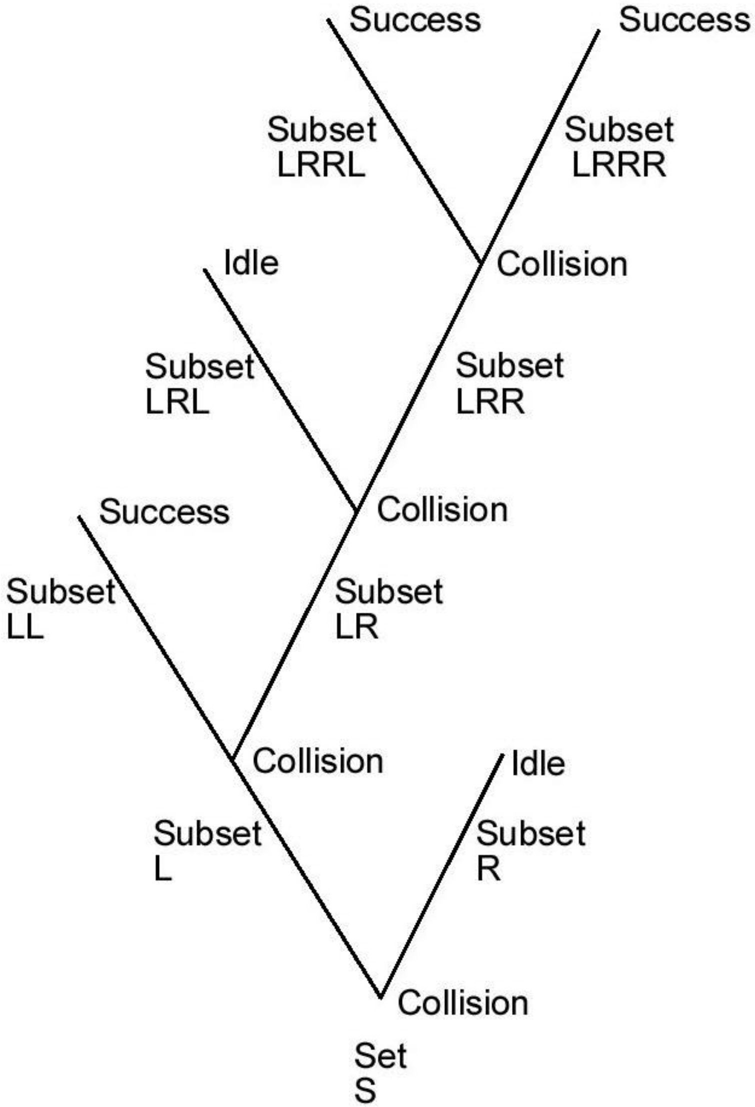
$$E[N] = \sum_{k=2}^{\infty} k \cdot 2^{-(k-1)} = 3$$

  – Throughput becomes 2/3
- Split the set of colliding nodes into subsets in order to reduce collision rate

Tree algorithm (Splitting)

- Assume a collision at the k^{th} slot.
  - All nodes not involved in the collision go into a waiting mode.
  - All nodes involved in the collision are split into two subsets.
  - The first subset transmits in slot $(k+1)$
  - If a collision occurs again, repeat the above.
  - If $(k+1)$ is idle or successful (0 or 1), the second subset transmits in slot $(k+2)$.

- Splitting can be done in various ways
  - Coin flipping, node ID, arrival time
- Improvements to the tree algorithm, e.g., using the example in the next slide
  - If a collision is followed by an idle slot, split the next subset (i.e., LRR) before transmission
  - After two consecutive collisions, the second subset (i.e., R) as regarded as just another part of the waiting new arrivals, never involved in a collision



Slot	Xmit Set	Waiting Sets	Feedback
1	S	–	*e*
2	L	R	*e*
3	LL	LR, R	1
4	LR	R	*e*
5	LRL	LRR, R	0
6	LRR	R	*e*
7	LRRL	LRRR, R	1
8	LRRR	R	1
9	R	–	0

First come first serve splitting algorithm

- Algorithm
  - Define

    F: feedback, σ : L or R (status),

    α: allocation interval

    1: success, 0: idle, e: collision,

    T: allocation starting time
  - If F=e,

    $$T(k)=T(k-1); \quad \alpha(k)=\frac{\alpha(k-1)}{2}; \quad \sigma(k)=L;$$
  - If F=1 and $\sigma(k-1)=L$,

    $$T(k)=T(k-1)+\alpha(k-1); \quad \alpha(k)=\alpha(k-1); \quad \sigma(k)=R;$$

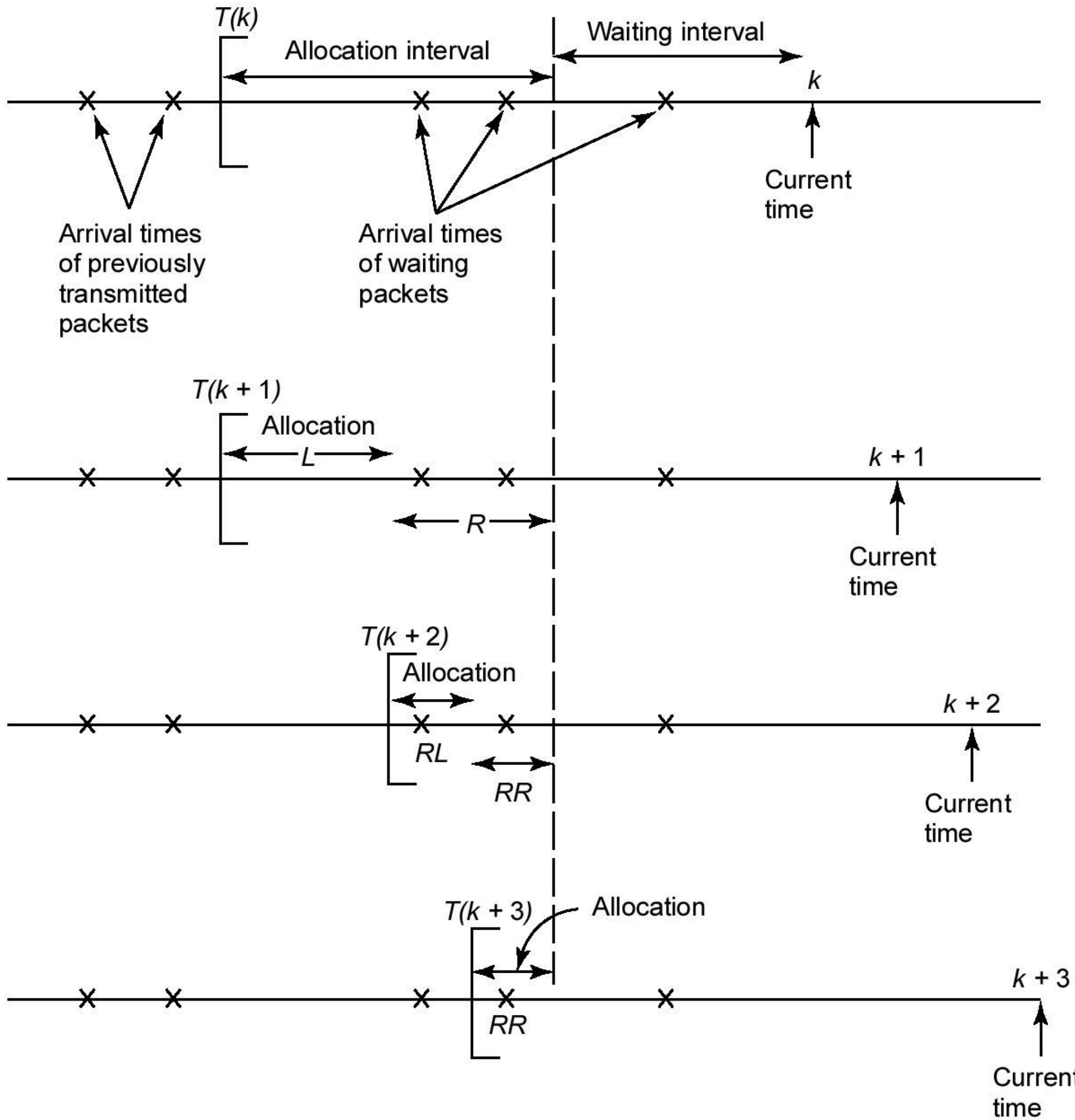
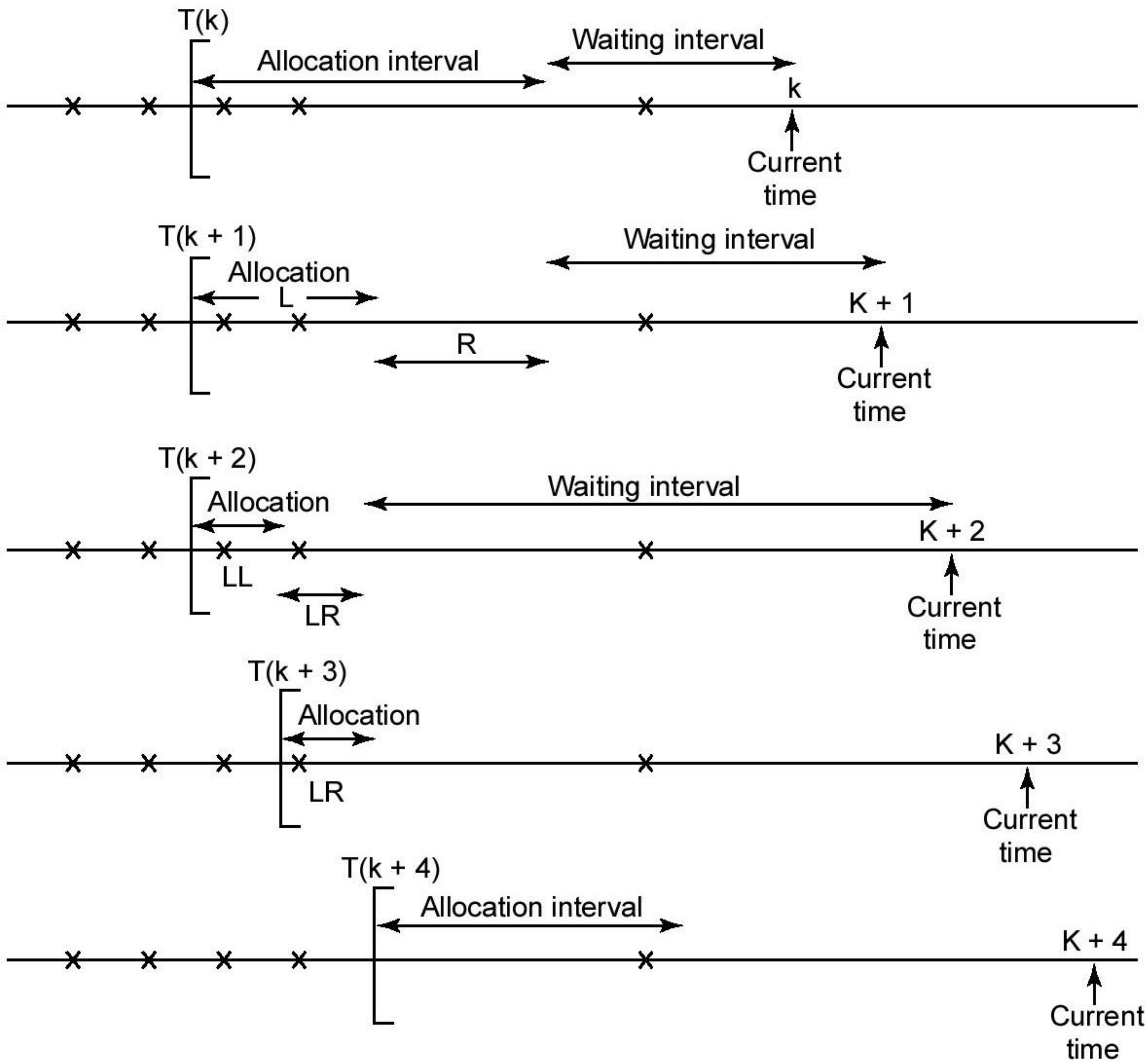


Figure 4.11 FCFS splitting algorithm.



T(k)
Allocation interval
Waiting interval
k
Current time
T(k + 1)
Allocation
L
R
Waiting interval
K + 1
Current time
T(k + 2)
Allocation
LL
LR
Waiting interval
K + 2
Current time
T(k + 3)
Allocation
LR
K + 3
Current time
T(k + 4)
Allocation interval
K + 4
Current time

– If F=0 and $\sigma(k-1)=L$,

$$T(k)=T(k-1)+\alpha(k-1); \quad \alpha(k)=\frac{\alpha(k-1)}{2}; \quad \sigma(k)=L;$$

– If (F=0 or 1) and $\sigma(k-1)=R$,

$$T(k)=T(k-1)+\alpha(k-1); \quad \alpha(k)=\min[\alpha_0, k-T(k)]; \quad \sigma(k)=R;$$

- Markov chain model
- Contention Resolution Protocol (CRP)
  – (L, i) :status = L and i-th split of the original allocation interval (AI)

- A success from a right interval ends the CRP always with a transition back to (R,0).
- If a collision occurs from a right interval, transition from (R,i) to (L,i+1).
- If idle or collision from a left interval, transition from (L,i) to (L,i+1).
- $G_i = \lambda 2^{-i} \alpha_0$: expected number of packets for the interval of i-th split

$$P_{R,0} = \frac{e^{-G_0} G_0^0}{0!} + \frac{e^{-G_0} G_0^1}{1!} = (1 + G_0) e^{-G_0}$$

  - prob. of an idle or success (0 or 1) in the initial AI.

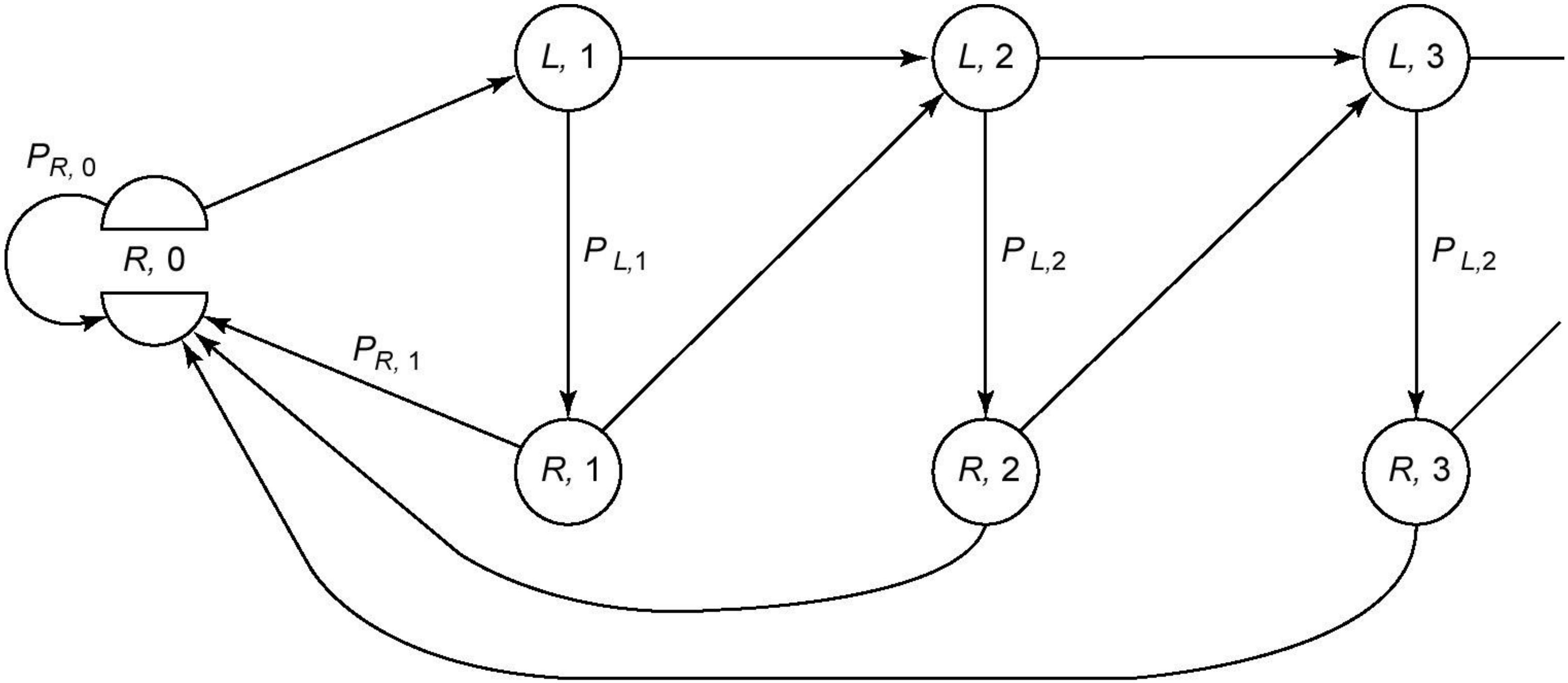


Figure 4.13 Markov chain for FCFS splitting algorithm. The top states are entered after splitting an interval and correspond to the transmission of the left side of that interval. The lower states are entered after success on the left side and correspond to transmission of the right side. Transitions from top to bottom and from bottom back to *R*, 0 correspond to successes.

– x_L: the number of packets in the left half interval of new AI

– x_R : the number of packets in the right half interval of new AI

$$P_{L,1} = \frac{P[x_L = 1]P[x_R \geq 1]}{P[x_L + x_R \geq 2]} = \frac{\text{success in } \alpha_0 / 2 \text{ interval}}{\text{collision in } \alpha_0 \text{ interval}}$$

$$= \frac{G_1 e^{-G_1}(1 - e^{-G_1})}{1 - (1 + G_0)e^{-G_0}}$$

$$P_{R,1} = \frac{P[x_R = 1]}{P[x_R \geq 1]} = \frac{G_1 e^{-G_1}}{1 - e^{-G_1}}$$

- Generally

$$P_{L,i} = \frac{G_i e^{-G_i}(1-e^{-G_i})}{1-(1+G_{i-1})e^{-G_{i-1}}}$$

$$P_{R,i} = \frac{G_i e^{-G_i}}{1-e^{-G_i}}$$

- Stability analysis
  - No state can be entered more than once before returning to (R,0)
  - notations

K: the number of states visited before returning to $(R,0)$, or equivalently, # of slots in a CRP

$p(L,i)$: the prob. of entering the state (L,i)

$$p(L,1) = 1 - P_{R,0}$$

$$p(R,i) = P_{L,i}\, p(L,i) \qquad\qquad i \geq 1$$

$$p(L,i+1) = (1-P_{L,i})\, p(L,i) + (1-P_{R,i})\, p(R,i)$$

$$E[K] = 1 + \sum_{i=1}^{\infty} [\, p(L,i) + p(R,i)]$$

– define f be the fraction of α_0 returned to the waiting interval (prob. of collisions on the left interval)

– $\alpha_0(1-f)$: change in T(k)

$$P[e\,|\,(L,i)] = \frac{P[x_L \geq 2]}{P[x_L + x_R \geq 2]} \quad = \frac{1-(1+G_i)e^{-G_i}}{1-(1+G_{i-1})e^{-G_{i-1}}}$$

$$E[f] = \sum_{i=1}^{\infty} p(L,i)\,p[e\,|\,(L,i)]2^{-i} \qquad \text{where} \qquad G_i = \lambda\alpha_0 2^{-i}$$

– E[K] and E[f] are functions of $\lambda\alpha_0$

$$\lim_{i\to\infty} P_{L,i} = 1/2$$

$$\lim_{i\to\infty} p(L,i) = \lim_{i\to\infty} p(R,i) = 0$$

– Define the drift D to be the expected change in the time backlog, k-$T(k)$, over a CRP.

$$D = E[K] - \alpha_0 (1 - E[f])$$

– To be stable, D<0, then

$$\lambda < \frac{\lambda \alpha_0 (1 - E[f])}{E[K]}$$

- Example) for $\alpha_0 = 2.6, \quad \lambda < 0.48$

- Last Come First Serve (LCFS) splitting algorithm
  - Try the right half first to resolve the contention.

CSMA Protocol

- ignore the detection delay, synchronization delay for the reception.
- detect idle periods quickly
- β: the propagation and detection delay (in packet trans. units) required for all sources to detect idle after a transmission ends.

$\beta = \dfrac{\tau C}{L}$ where C: channel rate, τ : end to end prop. Delay, L: expected packet length

CSMA: Carrier Sense Multiple Access

- If the channel is idle for β, a new slot begins again.
- Assume infinite set of nodes and Poisson arrivals of λ overall intensity.
- CSMA types
  - non-persistent
  - persistent (transmit new packet with prob.=1 when idle)
  - p-persistent (transmit new packet with prob.=p when idle)

- Non-persistent CSMA (slotted)
  - Idle slots have a duration of β.
  - A packet arriving in an idle slot is transmitted in the subsequent slot
  - A packet arriving in a busy slot is regarded backlogged
    * The backlogged nodes send the frame with the prob. of q_r when idle.

CSMA-Analysis

– Since nodes can transmit after detecting an idle slot, each busy slot (success or collision) must be followed by an idle slot.

– Transition time interval

β : idle slot ($\beta << 1$)

$1+\beta$: an idle slot after 1 busy data slot

– The prob. of no transmissions at state *n* at the end of an idle slot.

(where *n* : number of backlogged nodes)

$$e^{-\lambda\beta}(1-q_r)^n$$

– The expected time between state transitions in state *n*. $\beta + 1 \cdot [1 - e^{-\lambda\beta}(1-q_r)^n]$

– E[arrivals]= $\lambda \cdot [\beta + 1 - e^{-\lambda\beta}(1-q_r)^n]$

– $P_{succ} = (\lambda\beta + \dfrac{q_r n}{1-q_r})e^{-\lambda\beta}(1-q_r)^n$

cf) $P_{succ} = \left(\dfrac{(m-n)q_a}{1-q_a} + \dfrac{nq_r}{1-q_r}\right)(1-q_a)^{m-n}(1-q_r)^n$

– Drift

$$D_n = E[arrivals] - P_{succ} \approx \lambda(\beta + 1 - e^{-g(n)}) - g(n)e^{-g(n)}$$

where $g(n) = \lambda\beta + q_r n$

$$D_n < 0 \qquad \text{if}$$

$$\lambda < \frac{g(n)e^{-g(n)}}{\beta + 1 - e^{-g(n)}}$$

$$= \frac{\text{ave. \# of departures per state transition}}{\text{ave. duration of a state transition period}}$$

$$= \text{departure rate (or ave. departures per unit time)}$$

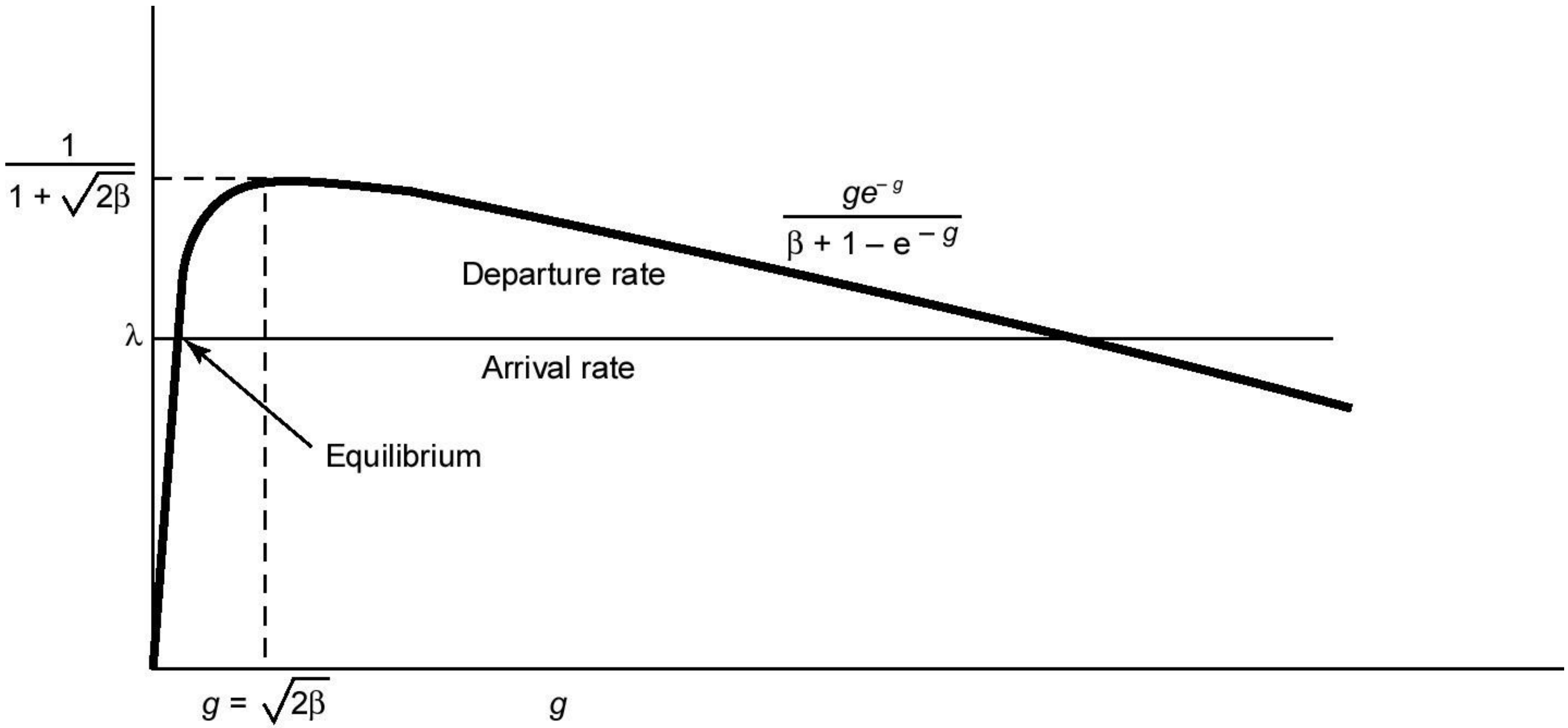


Figure 4.20 Department rate, in packets per unit time, for CSMA slotted Aloha as a function of the attempted transmission rate g in packets per idle slot. If β, the duration of an idle slot as a fraction of a data slot, is small, the maximum departure rate is $1/(1 + \sqrt{2\beta}\,)$.

Pseudo-Baysian Stabilization

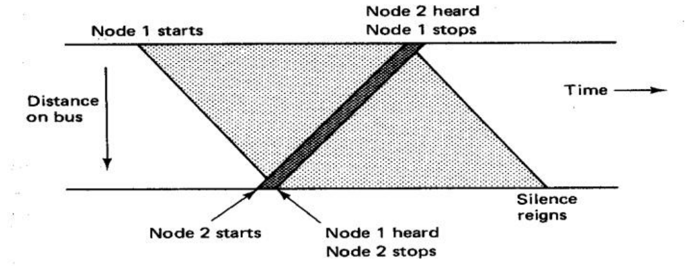
- all packets are backlogged immediately on arrival

$$q_\gamma(\hat{n}) = \min\left[\frac{\sqrt{2\beta}}{\hat{n}}, \sqrt{2\beta}\right]$$

$$\begin{aligned} \hat{n}_{k+1} &= \hat{n}_k\left[1 - q_\gamma(\hat{n}_k)\right] + \lambda\beta && \text{for idle} \\ &= \hat{n}_k\left[1 - q_\gamma(\hat{n}_k)\right] + \lambda(1+\beta) && \text{for success} \\ &= \hat{n}_k + 2 + \lambda(1+\beta) && \text{for collision} \end{aligned}$$

Local Area Networks: CSMA/CD and Ethernet

- Slotted CSMA/CD



  - 1 slot for data packet transmission
  - Mini slot of β for one-way propagation delay
    - Use mini slots for the contention mode
    - If a node succeeds at a mini slot, it reserves the channel to transmit the packet completely
    - If a collision detected, transmission stops after β
  - Nodes are synchronized into mini slots of duration

– Obtain the throughput bound

Number of nodes transmitting after an idle slot (Poisson)

$$g(n) = \lambda\beta + q_r n$$

Expected length of state transition interval

$$E\left[\text{interval}\right] = \beta + 1 \cdot g(n)e^{-g(n)} + \beta\left[1 - (1 + g(n))e^{-g(n)}\right]$$

Success Probability

$$P_{succ} = (\lambda\beta + \frac{q_r n}{1 - q_r})e^{-\lambda\beta}(1 - q_r)^n \approx g(n)e^{-g(n)}$$

Drift

$$\lambda E\left[\text{interval}\right]-P_{succ}$$

To be stable

$$\lambda < \frac{P_{succ}}{\beta + 1\cdot g(n)e^{-g(n)} + \beta\left[1-(1+g(n))e^{-g(n)}\right]}$$

max at $g(n) = 0.77$

$$\lambda < \frac{1}{1+3.31\beta}$$

Multiaccess Reservations

- Reservation via mini-slot of duration *v*, then transmit data of average duration 1 via assigned data slots
- Max. throughput (in data packets/time unit)

$$S = \frac{1}{1 + v / S_r} \qquad \text{for reservation w/o data}$$

$$= \frac{1}{1 + v(1/S_r - 1)} \qquad \text{for reservation w/ data cf. CSMA/CD}$$

  – S_r = 1/e for slotted Aloha, 0.478 for splitting, 1 for TDM

Satellite Reservation Systems

- One way propagation delay of 260msec(β)
- Reservation
  - use TDM, no limit on packet and frame sizes
  - No reservation collision needed
- Single user reservation system (m=1)
  - $E[W_i] = E[R_i] + \frac{E[N_i]}{\mu} + 2A$ where the fixed reservation interval of A.

$$W = \frac{\lambda \hat{X}^2}{2(1-\lambda)} + \frac{A}{2} + \frac{2A}{1-\lambda}$$

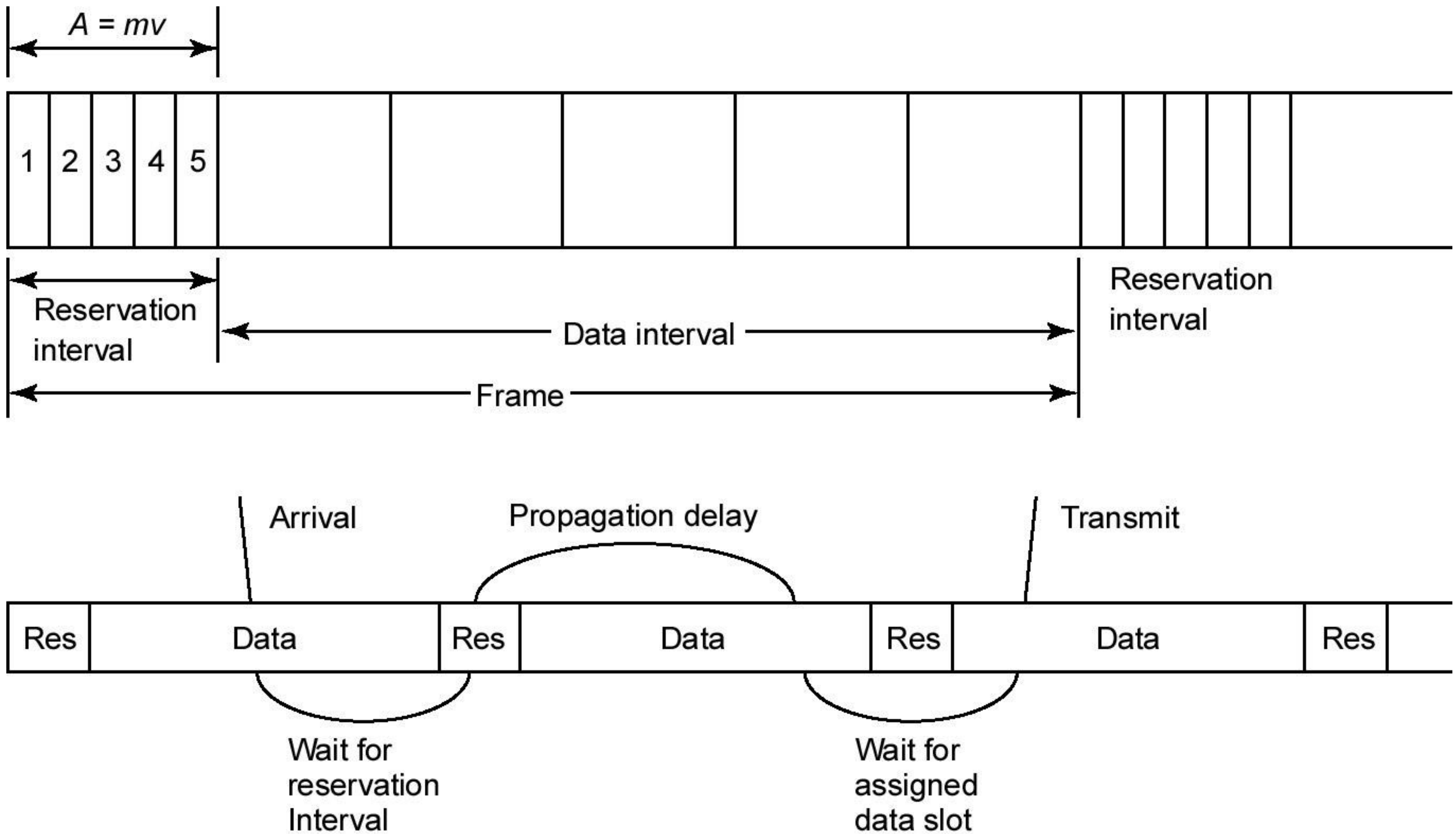


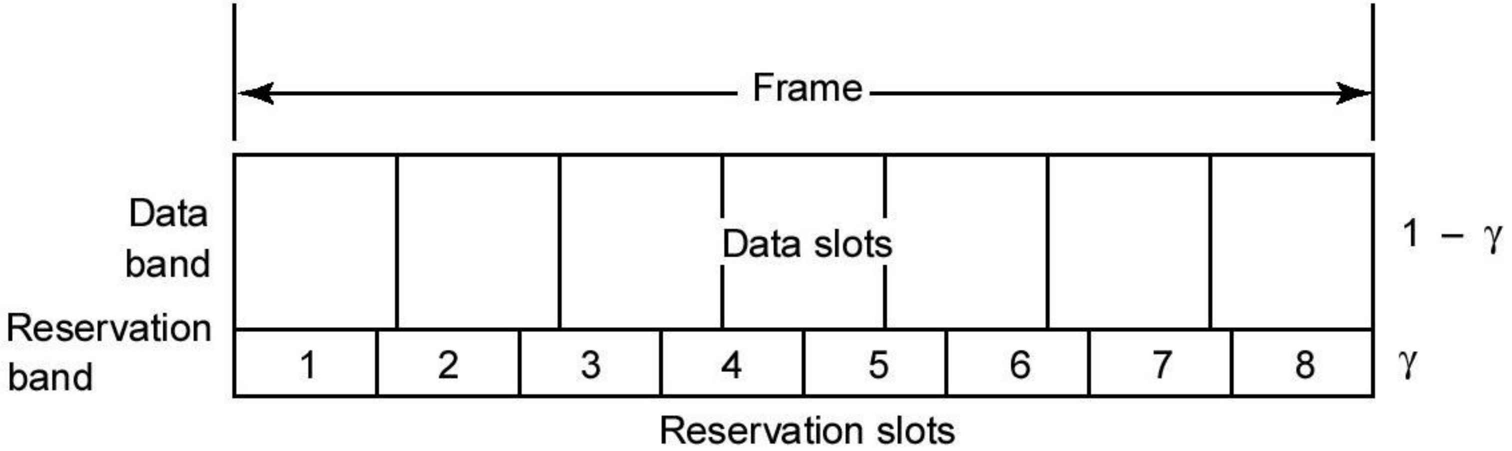
Figure 4.17 Satellite reservation system, using TDM to make reservations.

– Problem : the round-trip delay of 2β has been neglected.

– To be stable

$\lambda < 1$ (as $\lambda \to 1$, the approximation is good)

- System with separate frequency band for reservations (m=8)
  - Fixed frame size



$$\gamma = \frac{mv}{2\beta}$$

- $\frac{2\beta}{2}$: waiting time for the reservation slot
- $\frac{2\beta}{m}$: transmission time for the reservation packet
- 2β : waiting time for the reservation ACK
- $\frac{X}{1-\gamma}$: packet transmission time where X is the packet transmission time using the full bandwidth
- Utilization $\rho = \frac{\lambda}{1-\gamma}$
- From the M/G/1 analysis ($W = \frac{\lambda \overline{X^2}}{2(1-\rho)}$)

$$\frac{\lambda \left(\frac{\overline{X^2}}{(1-\gamma)^2}\right)}{2\left(1-\frac{\lambda}{1-\gamma}\right)} = \frac{\lambda \overline{X^2}}{2(1-\gamma-\lambda)(1-\gamma)}$$

$$W = 3\beta + \frac{2\beta}{m} + \frac{\lambda \overline{X^2}}{2(1-\gamma-\lambda)(1-\gamma)}$$

– Perfect scheduling at the expense of the delay for making reservations