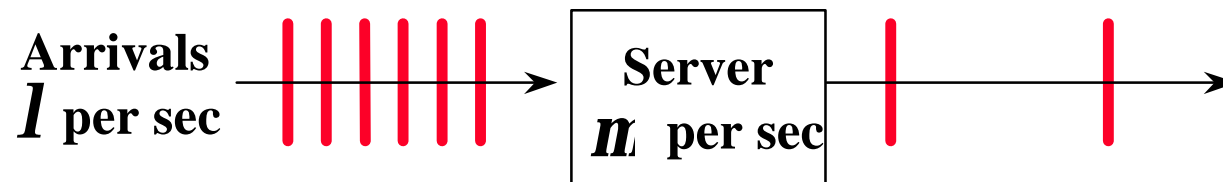


COMPUTER NETWORKS

Queueing Theory Basics - Transport Protocols



The author of these slides is Dr. Mark Pullen of George Mason University. Permission is hereby granted to reproduce the slides for any not-for-profit academic purpose, so long as this notice is contained in any document containing the slides, and each slide remains intact as provided, including the copyright label. Permission for any other use, either in machine-readable or printed form, must be obtained from the author in writing.

Queueing Theory

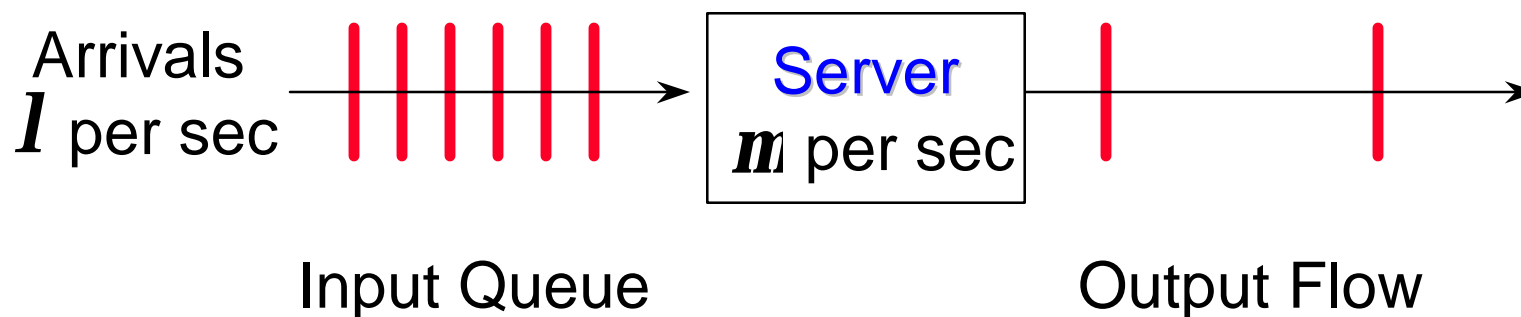
Queueing theory provides a mathematical basis for understanding and predicting the behavior of packet-switched networks.

References:

1. Gross & Harris, *Fundamentals of Queueing Theory*, 3rd Ed., Wiley, 1998
2. Kleinrock, *Queueing Systems*, Wiley, 1975
3. Bertsekas & Gallager, *Data Networks*, 2nd Ed., Prentice-Hall, 1992

Queueing Theory

Basic Model: Single Server M/M/1



Common assumptions (**M/M/1** single-server queue):

Exponentially distributed inter-arrival time (Poisson arrivals)

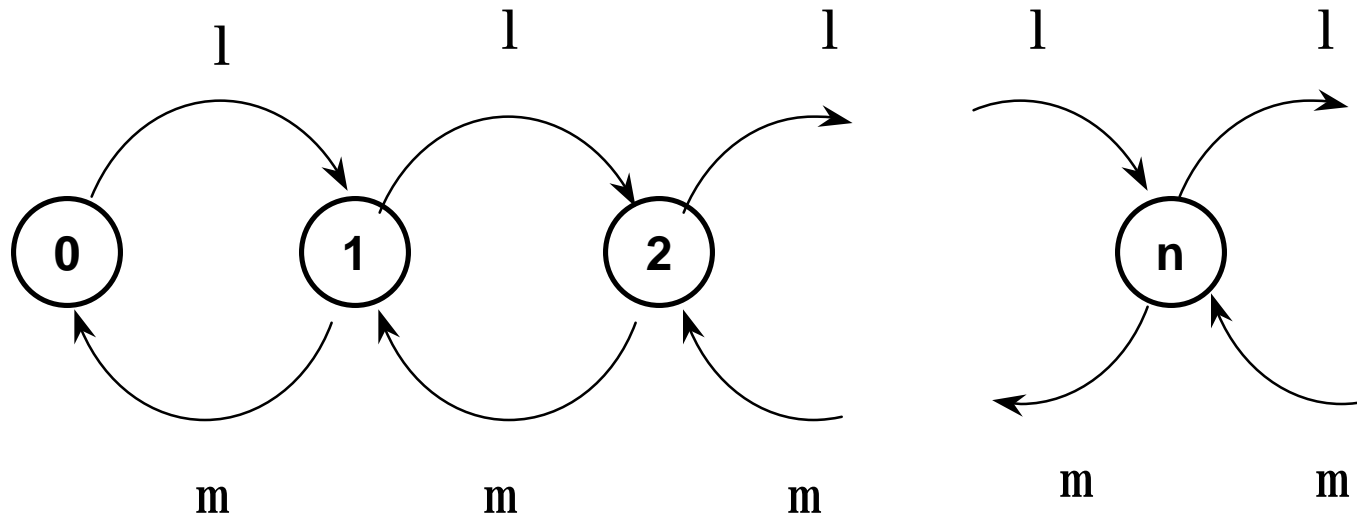
Exponentially distributed service time, independent of arrivals

Mean arrival rate must be less than mean service rate!

These assumptions are often close to reality, generally make the analysis more tractable.

Queueing Theory

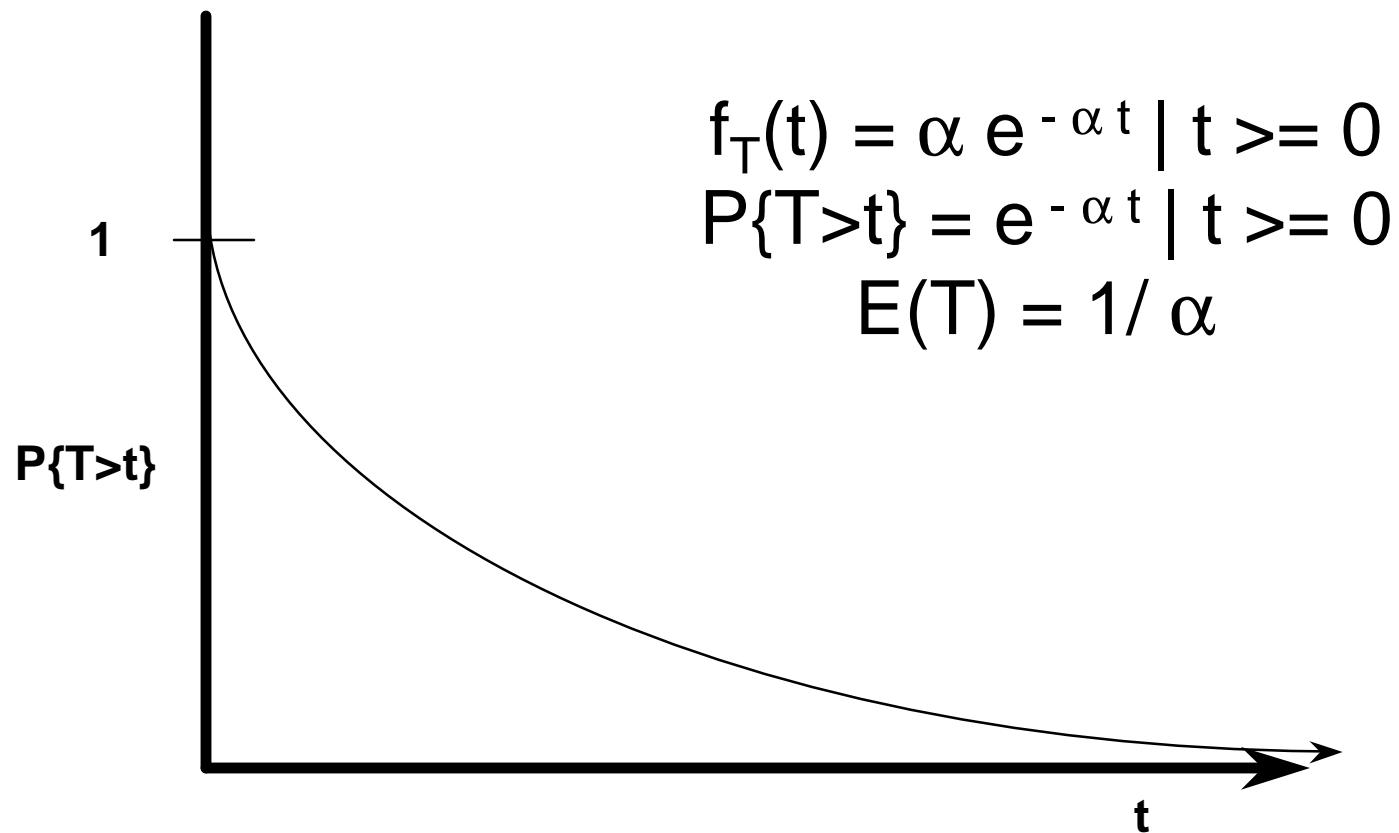
Basic Model: Single Server M/M/1



In steady state (requires $\lambda < \mu$):

$$p_1 = (\lambda / \mu) p_0 \quad p_n = (\lambda / \mu)^n p_0 \quad p_0 = 1 - \lambda / \mu$$

Exponential Distribution



Queueing Formulas

Exponential distribution

$$f_T(t) = \mathbf{a} e^{-at} \mid t \geq 0$$

Number in system

$$\mathbf{N} = \mathbf{r} / (1 - \mathbf{r}) = \mathbf{l} / (\mathbf{m} - \mathbf{l})$$

this formula is only true for M/M/1

Waiting time in queue

$$\mathbf{W} = \mathbf{T} - \mathbf{1} / \mathbf{m}$$

Utilization factor

$$\mathbf{r} = \mathbf{l} / \mathbf{m} \mid \mathbf{l} \leq \mathbf{m}$$

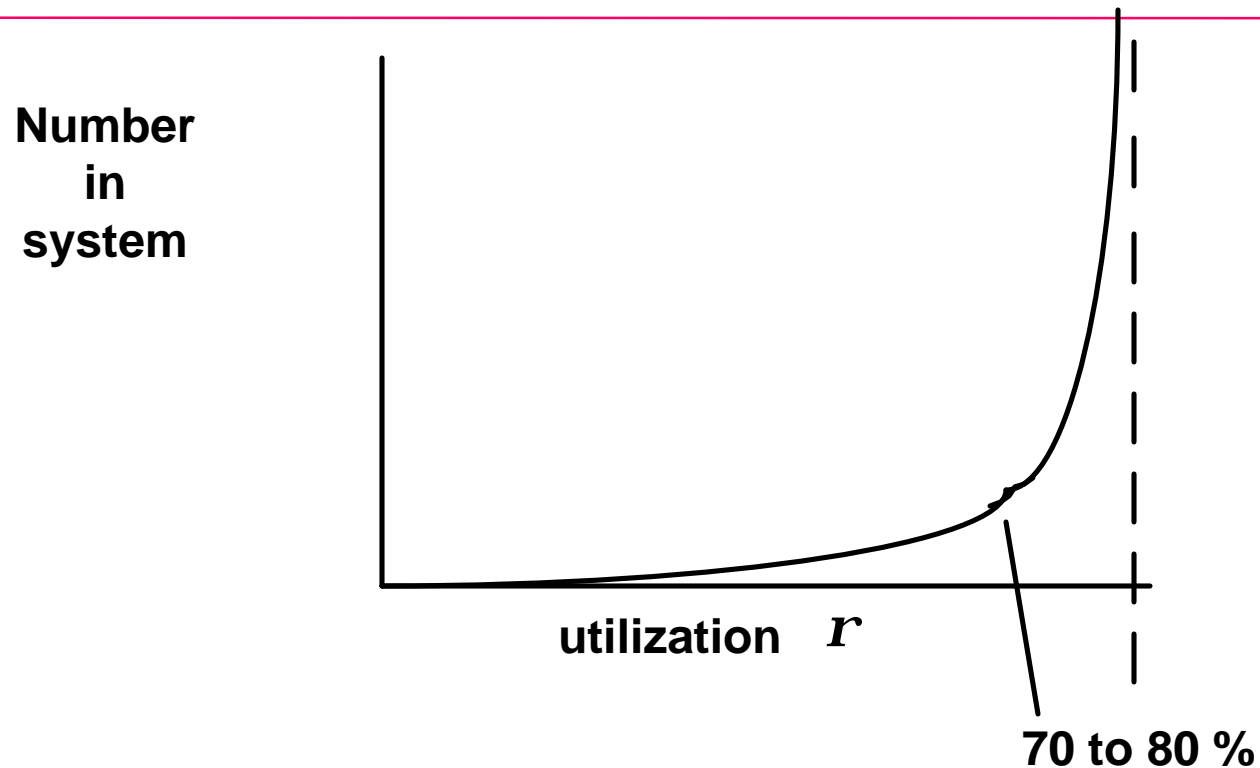
Time in system
(Little's formula)

$$\mathbf{T} = \mathbf{N} / \mathbf{l}$$

Number in queue

$$\mathbf{N}_q = \mathbf{l} \mathbf{W}$$

Packet Switching M/M/1 Queueing



$$r = \frac{l}{m} = \frac{\text{average arrival rate}}{\text{average service rate}}$$

M/M/1 Queueing Theory Example

An IP router is sending packets through a 64 kbps link. The length of the packets, when framed, is an exponential random variable with mean 400 bytes. The interval between arrival of packets (from the router's other inputs) is an exponential random variable with mean 15 packets per second. When a new packet is routed through this link and another packet is already being sent, the new packet is stored in a first-come first-served queue.

Calculate the average length of the queue in the router in bytes, and the average time from availability of a packet to be transmitted, until its last bit has been transmitted.

$$\mu = 64000 / (400 * 8) = 20 \text{ pps}$$

$$N = \rho / (1 - \rho) = .75 / .25 = 3$$

$$W = T - 1 / \mu = .2 - 1 / 20 = .15 \text{ s}$$

$$\rho = \lambda / \mu = 15 / 20 = .75$$

$$T = N / \lambda = 3 / 15 = .2 \text{ s}$$

$$N_q = \lambda W = 15 * .15 \\ = 2.25 \text{ packets}$$

M/M/1 Queueing Theory Example

continued

Repeat for average arrival rate 19 pps:

$$\rho = \lambda / \mu = 19 / 20 = .95 \quad N = \rho / (1 - \rho) = .95 / .05 = 19$$

$$T = N / \lambda = 19 / 19 = 1 \text{ s} \quad W = T - 1 / \mu = 1 - 1 / 20 = .95 \text{ s}$$

$$N_q = \lambda W = 19 * .95 = 18.05 \text{ packets}$$

Repeat for arrival rate 25:

The M/M/1 model is not useful, it only works when:

arrival rate < service rate.

The queue buffer will fill up and the router will start dropping packets at a rate of $25 - 20 = 5$ pps.

A More General Result
The Pollaczek- Khinchin (P- K) Formula
(M/G/1: any single server with Poisson arrivals)

Total time in a queueing node:

$$T = \bar{x} + \frac{l \bar{x}^2}{2(1-r)}$$

Where \bar{x} is average service time

l is average arrival rate

r is utilization = $\frac{\text{average arrival rate}}{\text{average service rate}}$

\bar{x}^2 is second moment of service (grows with variability of packet size)

P-K Formula Example

Two packet systems are used to transmit real-time data. Both systems have the same Poisson-distributed input, and same processor and link capacity, limited by the outgoing link speed of 1.0 Mbps. Both systems have average packet length 1000 bits and average arrival rate 900 pps. System A does not constrain the packets, their length is exponentially distributed and service therefore has second moment $2/\mu^2$. System B has fixed-length packets and therefore has second moment $1/\mu^2$. Find the mean time from arrival in queue to end of transmit for each system.

$$I = 900 \text{ pps} \quad m = 1000000 / 1000 = 1000 \text{ pps} \quad \bar{X} = 1 / m = .001 \text{ sec}$$

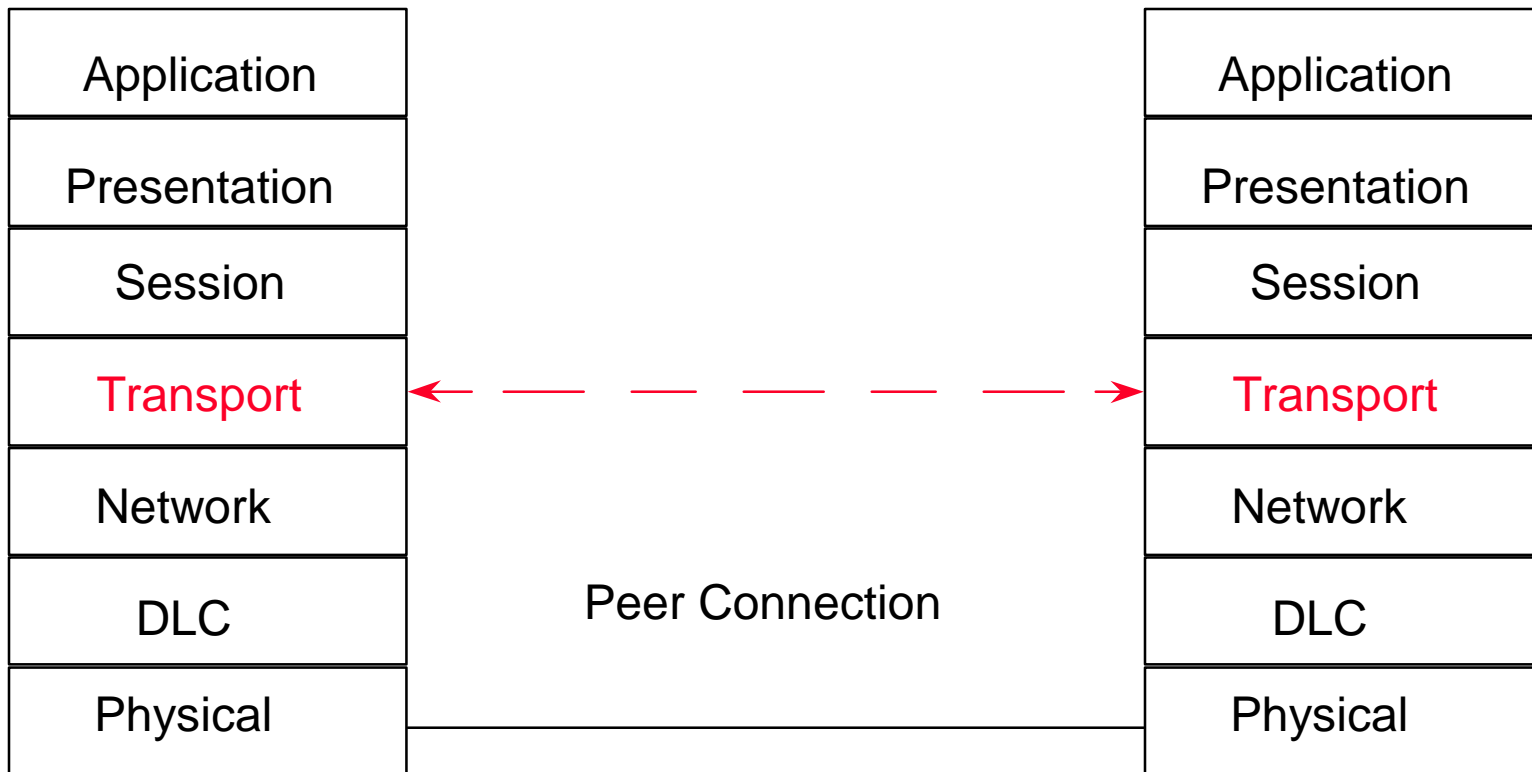
$$r = I / m = .9 \quad \overline{X_A^2} = 2 / m^2 = .000002 \quad \overline{X_B^2} = 1 / m^2 = .000001$$

$$T_A = \bar{X} + (I \overline{X^2}) / (2(1 - r)) = .001 + .009 = .010 \text{ sec}$$

$$T_B = .001 + .0045 = .0055 \text{ sec}, \text{ A's delay is nearly twice as large!}$$

Transport Protocols

Seven Layer Model



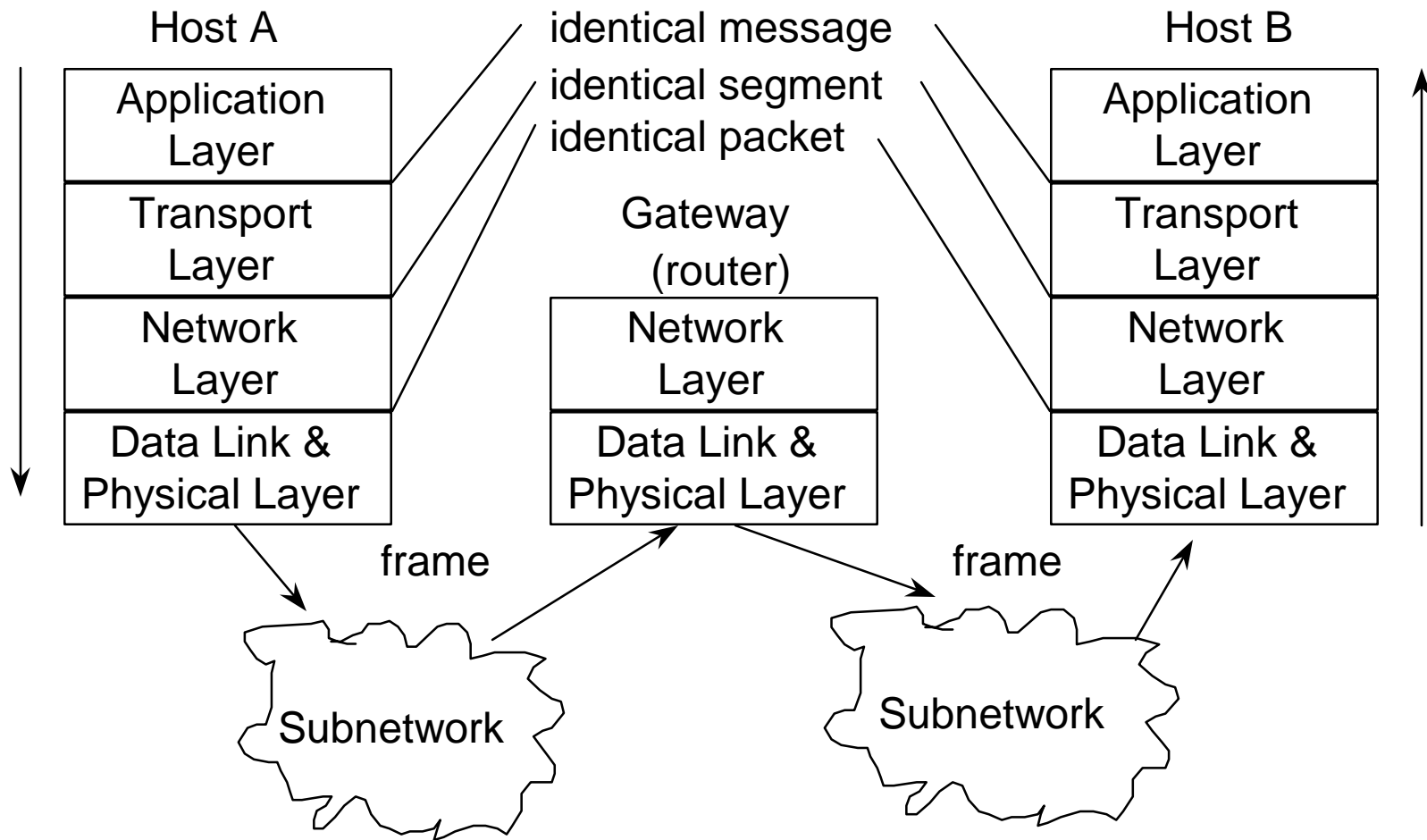
Seven Layer ISO Reference Model

1. Physical - communications medium: wire, fiber, radio, etc.
2. Data Link Control - framing, link error & flow control
medium access control (MAC) sublayer
used for multi-access e.g. Ethernet
3. Network - routing, congestion control
internet sublayer - interconnects networks
4. Transport - end-to-end error and flow control
5. Session - connection management
6. Presentation - recoding transmitted form to display form
e.g. compression, encryption
7. Application - doing what the user needs
e.g. e-mail

Transport Layer Characteristics

- Transport Layer protocols reside in the source and destination nodes.
- Transport Layer provides a consistent service interface to the network.
- Transport Layer must provide reliable connection despite
 - ▶ Connectionless Network services
 - ▶ Virtual Circuit Resets on Connection-oriented Network Services
 - ▶ packet reordering that can occur when a Transport Connection is split across multiple Virtual Circuits

Transport Protocol End-to-end Operation



Transport Layer vs DLC Layer

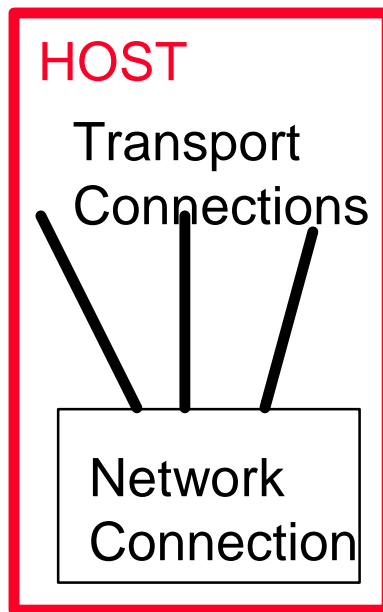
- Transport Layer and Data Link Layer bear some resemblance:
 - ▶ both perform error control, resequencing, and flow control.
 - ▶ Data Link layer operates on a link basis
 - ▶ Transport Layer operates end-to-end across a network

Elements Of Transport Protocols

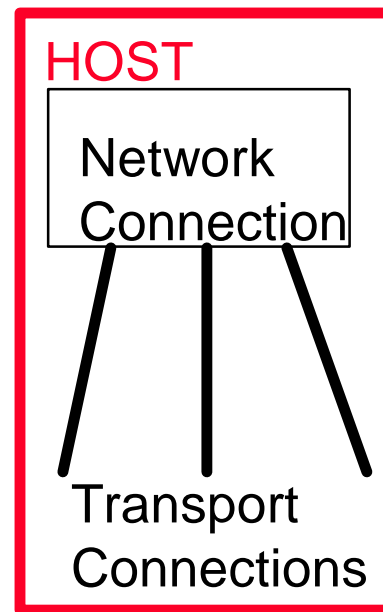
- Connection Management - establishment, refusal, and release. Manage mapping between transport and network connections or build a connection-oriented service from a connectionless network service. Link applications across a network.
- Fragmentation and reassembly of application data - fragment data stream into packet sizes suitable for the network. Where multiple transport connections exist packets must be correctly associated with each connection. Receiver must reassemble packets and resequence the data stream to mirror what was sent.
- Recovery from network failures - reassignment after network disconnects and resynchronization after network resets.
- Error Control and Flow Control - especially for LANs.

Elements Of Transport Protocols (continued)

- Multiplexing - mapping multiple Transport Connections to one Network Connection
- Splitting and Recombining - mapping one Transport Connection to multiple Network Connections.



Multiplexing



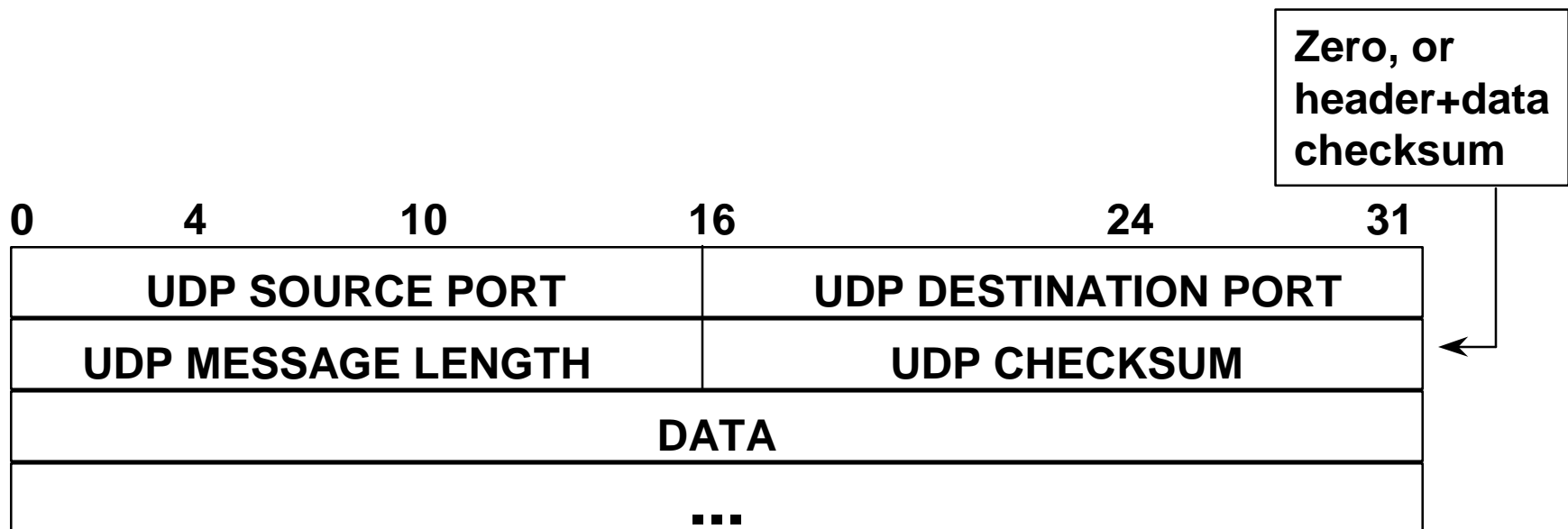
Splitting &
Recombining

Internet Transport Protocols

UDP - User Datagram Protocol
("unreliable data protocol")
used for best-effort services

TCP - Transmission Control Protocol
reliable transport

UDP Message Format



UDP Well-Known Ports

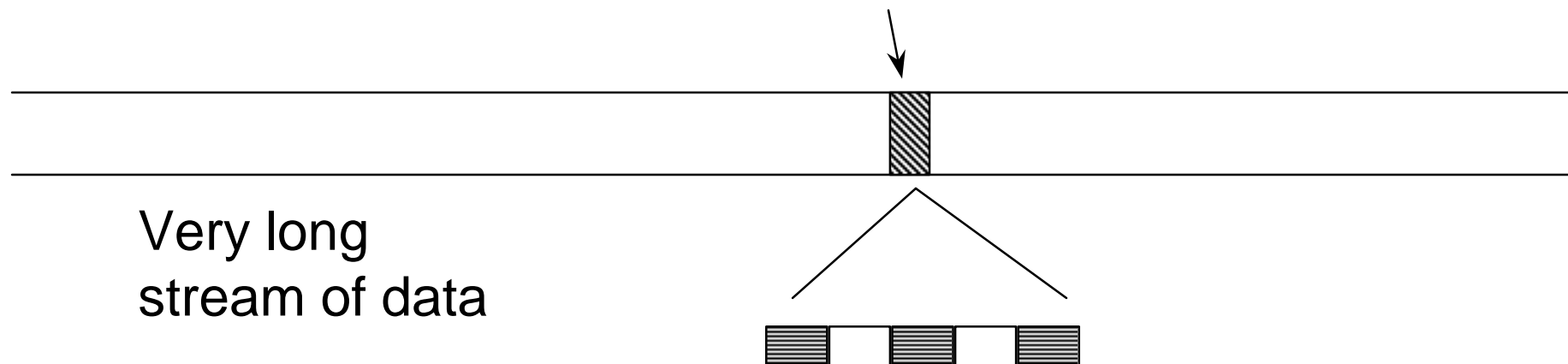
Decimal	Keyword	Unix Keyword	Description
0	ECHO	echo	Echo
7	DISCARD	discard	Discard
11	USERS	systat	Active Users
13	DAYTIME	daytime	Daytime
15	-	netstat	Who is up or NETSTAT
17	QUOTE	qotd	Quote of the Day
19	CHARGEN	chargen	Character generator
37	TIME	time	Time
42	NAMESERVER	nameserver	Domain Name Server
43	NICNAME	whois	Who is
53	DOMAIN	nameserver	Domain Name Server
67	BOOTPS	bootps	Bootstrap Protocol Server
68	BOOTPC	bootpc	Bootstrap Protocol Client
69	TFTP	tftp	Trivial File Transfer

TCP

- TCP provides connection for connectionless IP
- reliable data transfer
- stream model of data
- maintains order
- flow control via sliding window
- multiplexing multiple sessions via “ports”
(Berkeley implementation: socket = IP address + port number)
- full-duplex transmission (ACK piggyback on reverse stream)
- precedence and security options - not widely used
- disconnect

TCP Sliding Window

Window of data to be transported



Segments filled in as received

Connection Setup

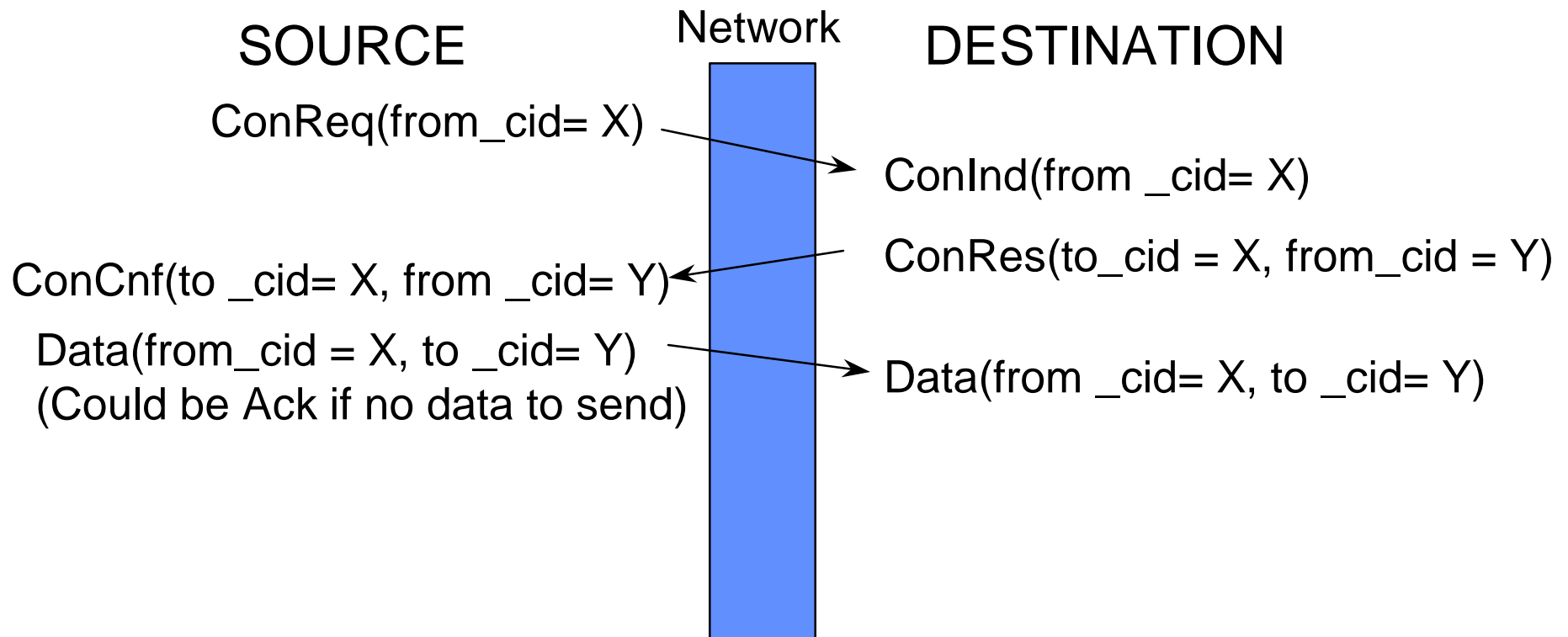
Problem: Delayed Duplicate Sequence Numbers

- Packets may be delayed in the network and arrive out of order at the destination.
- This presents the possibility that an old packet from an earlier transport connection might arrive and appear to be in sequence for some current transport connection.
- Two mechanisms are commonly employed to guard against this:
 - Discarding packets within the network after a certain time (or number of hops).
 - Prohibiting sequence number reuse for a certain time. (In this case, the sequence number is a combination of a connection identifier and a packet identifier within the connection; thus, really the connection identifiers must be recycled in a time-sensitive manner.)

Transport Layer Connection Establishment

Three-Way Handshake

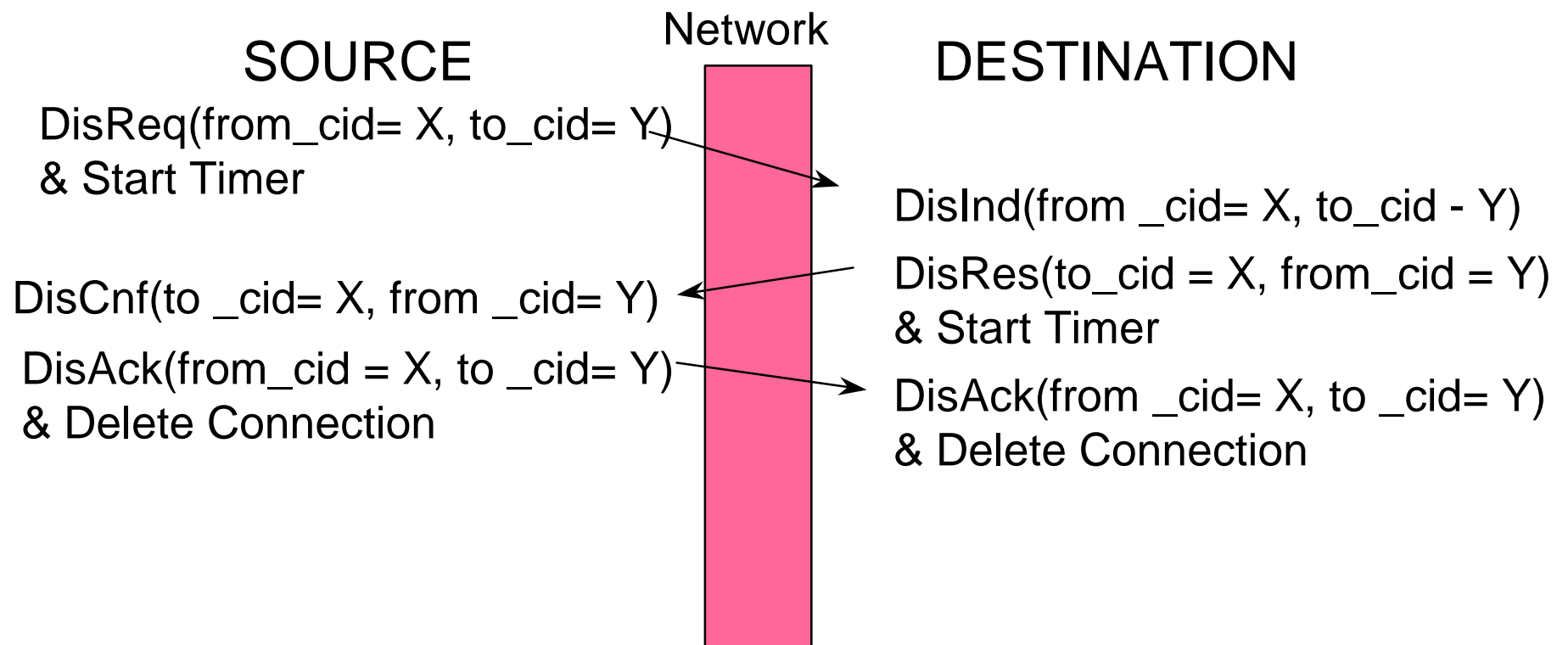
- The three-way handshake is used to avoid problems that can arise from delayed duplicate packets in connection setup.



Transport Layer Connection Release

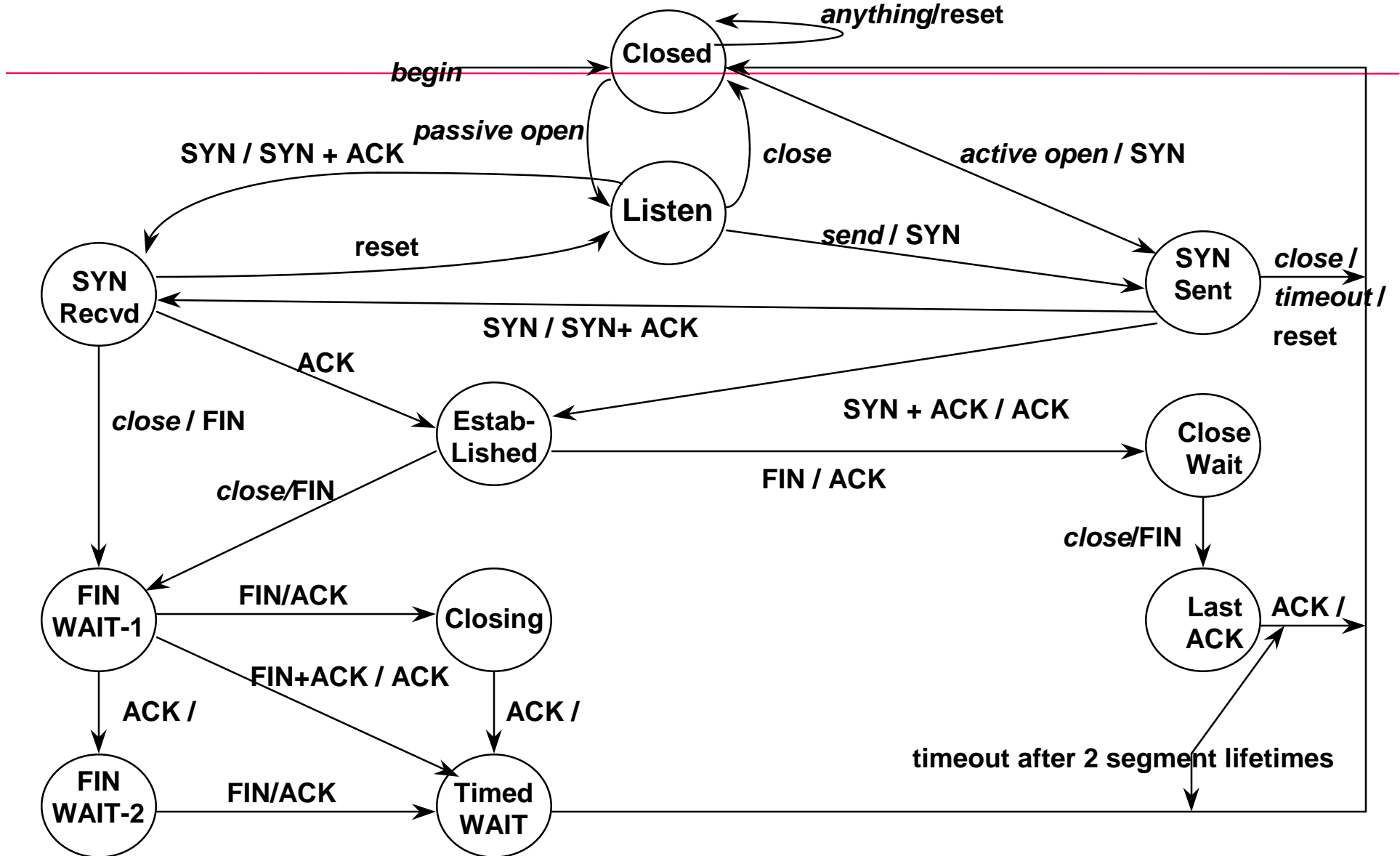
Three-Way Handshake

- To prevent problems with old duplicates of DISCONNECT messages, a three-way handshake is also used for connection release.

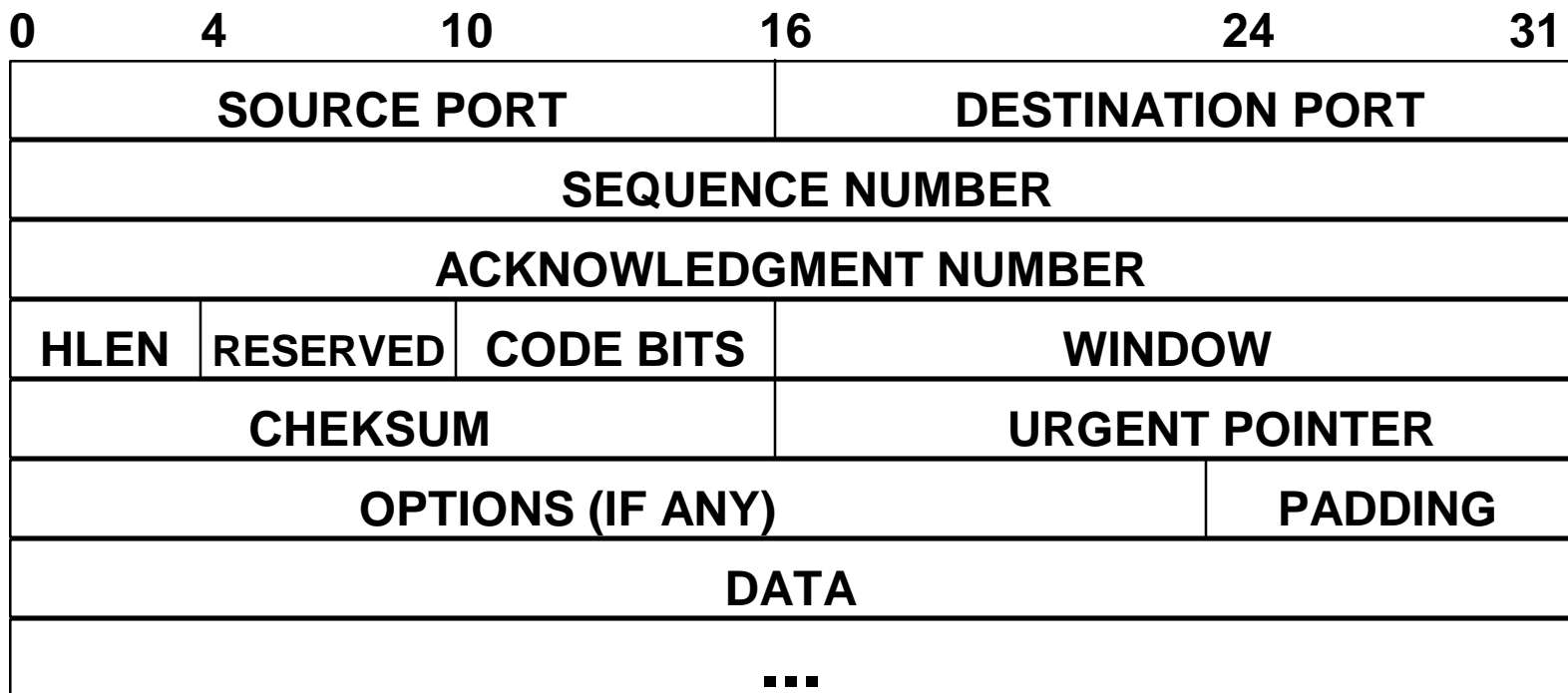


TCP Connection State Machine

from Comer *Internetworking with TCP/IP*



TCP Segment Format



TCP Well-Known Ports

Decimal	Keyword	Unix Keyword	Description
0			Reserved
5	RJE	-	Remote job entry
11	USERS	systat	Active Users
13	DAYTIME	daytime	Daytime
15	-	netstat	Network status program
17	QUOTE	qotd	Quote of the Day
20	FTP-DATA	ftp-data	File Transfer Protocol (data)
21	FTP	ftp	File Transfer Protocol
23	TELNET	telnet	network terminal emulator
25	SMTP	smtp	Simple Mail Transfer Protocol
37	TIME	time	Time
42	NAMESERVER	nameserver	Host Name Server
43	NICNAME	whois	Who Is
53	DOMAIN	nameserver	Domain Name Server
80	WWW-HTTP	http	World Wide Web HTTP

TCP Operation

- Code bits: used to originate connection
- Window: size varied for congestion control (reduce multiplicatively, increase additively)
- Round-trip time (RTT): TCP keeps track of time from SEND to ACK
- Timeout: if some multiple of RTT passes with no ACK, send again
- Establishing connection: requires 3-way handshake (one end must perform “passive open”)
- Well-known ports: reserved for functions (e.g. email) or “meeting points”
- Push: issued by sending application to cause TCP to send a segment (important for interactive applications)

TCP Startup

Port selection:

- application has pre-established “meeting” port
- if multiple sessions are possible for the application, can use initial connection to agree on a new “working” port

Slow-start:

- to avoid congestion
- start with minimum window
- double window size each round-trip
- if no congestion seen (by increased RTT), continue doubling until configuration max reached

TCP Timer Management

- Multiple timers are needed to manage TCP's complexity.
 - Retransmission timer
 - Persistence timer
 - Used to prevent deadlock if window size is 0 and the receiver update is lost
 - Keep alive timer
 - Timer for TIMED WAIT state.

TCP Retransmission Timer

- Used to decide when to retransmit segment.
- Problem: difficult to determine correct value for time-out over WAN.
 - if the timer is too short, unnecessary retransmissions occur, clogging up the network.
 - If the timer is too long, performance suffers due to long retransmission delays.
- Solution: use a highly dynamic algorithm that constantly adjusts the timeout interval.

TCP Timeout Algorithm

- The first step is to maintain a variable, RTT , which represents the best current roundtrip time estimate. Suppose an acknowledgement gets back in time M , where M is before the timeout expires. Then set

$$RTT = \alpha RTT + (1 - \alpha) M$$

where

$$\alpha < 1$$

TCP Timeout Algorithm

continued

- Next, to account for large variability, determine a deviation estimator D as

$$D = a D + (1 - a) |RTT - M|$$

- Finally, set the timeout as

$$Timeout = RTT + 4 D$$

TCP Timeout Algorithm

continued

- Problem: when an acknowledgement comes in, it is unclear whether the acknowledgement refers to the first transmission or the later one.
- Solution: do not update RTT on any segments that have been retransmitted. Instead, double the timeout.

TCP/IP - Some Observations

- X.25 deals with reliability at all layers; TCP/IP deals with it only at transport layer
- OSI has a strict boundary between layers 3 and 4; some TCP/IP implementations get better results by “cheating” and sharing data between layers 3 and 4
- TCP/IP software normally resides in the host operating system (e.g. Unix Kernel)

OSI Transport Layer Design Issues

- The main question for the Transport Layer protocol designer: “What can be assumed about the Network Service?”

Network Service Class	Description From OSI Transport Layer Standard
A	Error-free Service Without Network Resets
B	Perfect Packet Delivery With Network Resets
C	Unreliable Service (lost and duplicated packets possible) and Network Resets

OSI Transport Layer Answer

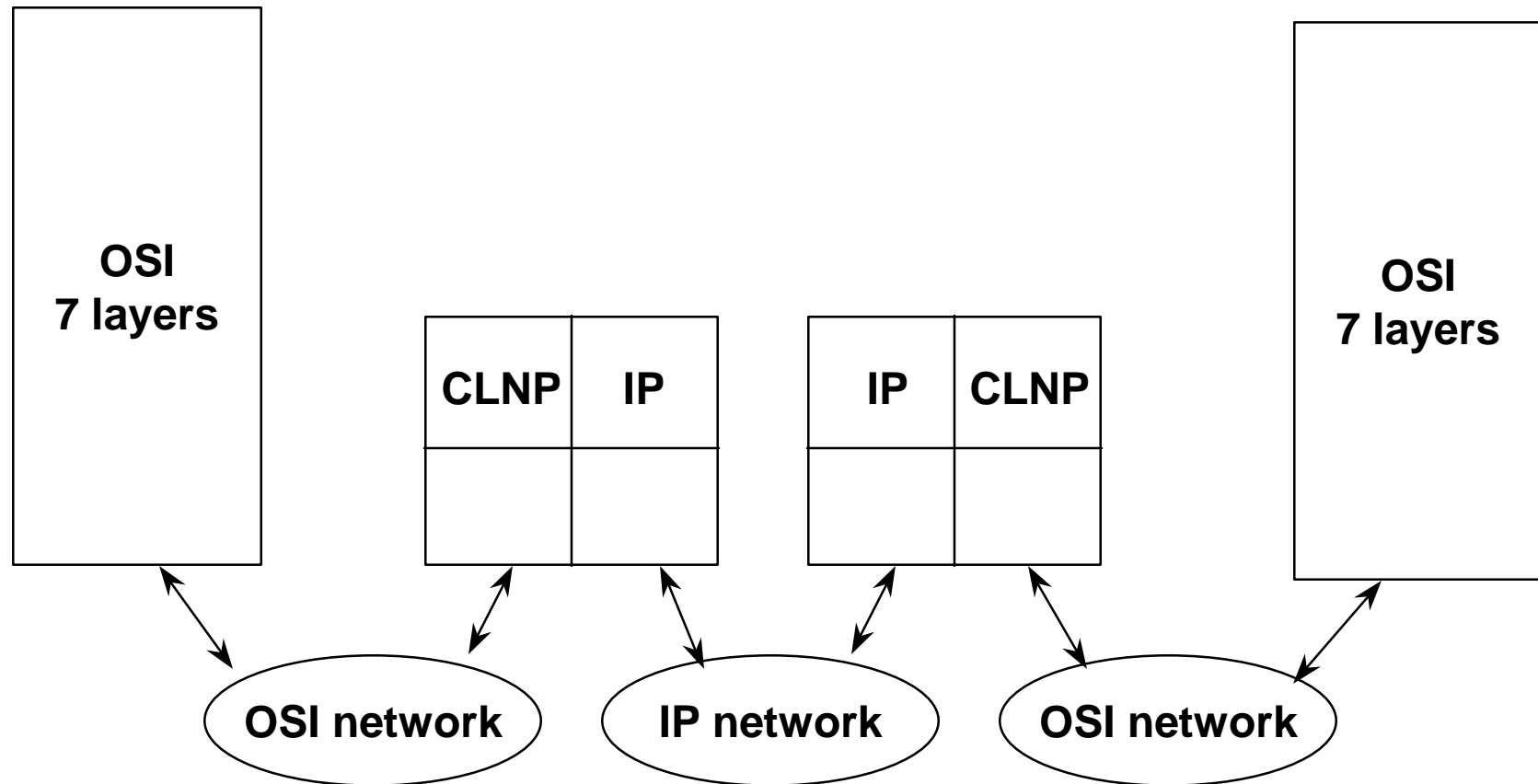
Five Transport Classes

- Class 0 & 2 Transports - assume Type A network service, but Class 2 adds the ability to multiplex transport connections onto a single network connection
- Class 1 & 3 Transports - assume Type B network service, but Class 3 adds the ability to multiplex transport connections onto a single network connection
- Class 4 - assumes Type C network service, includes ability to multiplex transport connections onto a single network connection and the ability to split a single transport connection across multiple network connections.
- Transmission Control Protocol (TCP) - similar to Class 4 OSI Transport protocol, but has differences in details.

OSI Transport Class 4 vs TCP

- TCP checksums computed with 32-bit words while **TP4** checksum computed with 8-bit words.
- TCP acknowledges bytes while Class 4 acknowledges transport packets.
- TCP operates only over connectionless network service.
- TCP includes functions for session synchronization and graceful connection close (in OSI these functions are reserved to the Session Layer).

Footnote: IP/OSI Interoperation



Project TRN1: Reliable Transport

- Simplified from TCP
 - only one connection open at a time
 - half duplex transmission
 - very short segment (20 bytes)
 - no DLC ARQ
- Four send states: closed, syn_sent, sending, close_wait
- Study tl.cpp pseudo-code comments, then program send_rtl_segments in stub tsend.cpp
- Might want to change profile for debugging:
 - print_at[] to see details of network operation
 - interactive = TRUE so output comes a few lines at a time
- Successful run will pass all email messages