

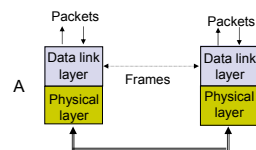
Lecture 5 Peer-to-Peer Protocols and Data Link Layer



PART II: Data Link Controls
Framing
Point-to-Point Protocol
High-Level Data Link Control
Link Sharing Using Statistical Multiplexing



Data Link Protocols



Data Links Services

- Framing
- Error control
- Flow control
- Multiplexing
- Link Maintenance
- Security: Authentication & Encryption

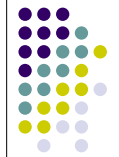
Examples

- Directly connected, wire-like
- Losses & errors, but no out-of-sequence frames
- Applications: Direct Links; LANs; Connections across WANs
- PPP
- HDLC
- Ethernet LAN
- IEEE 802.11 (Wi Fi) LAN

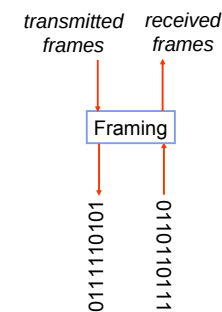
Lecture 5 Peer-to-Peer Protocols and Data Link Layer



Framing

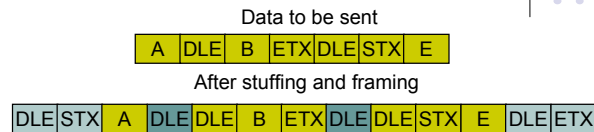


Framing



- Mapping stream of physical layer bits into frames
- Mapping frames into bit stream
- Frame boundaries can be determined using:
 - Character Counts
 - Control Characters
 - Flags
 - CRC Checks

Character-Oriented Framing

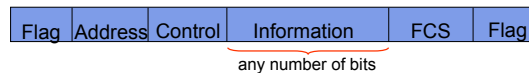


- Frames consist of integer number of bytes
 - Asynchronous transmission systems using ASCII to transmit printable characters
 - Octets with HEX value <20 are nonprintable
- Special 8-bit patterns used as control characters
 - STX (start of text) = 0x02; ETX (end of text) = 0x03;
- Byte used to carry non-printable characters in frame
 - DLE (data link escape) = 0x10
 - DLE STX (DLE ETX) used to indicate beginning (end) of frame
 - Insert extra DLE in front of occurrence of DLE STX (DLE ETX) in frame
 - All DLEs occur in pairs except at frame boundaries

Framing & Bit Stuffing

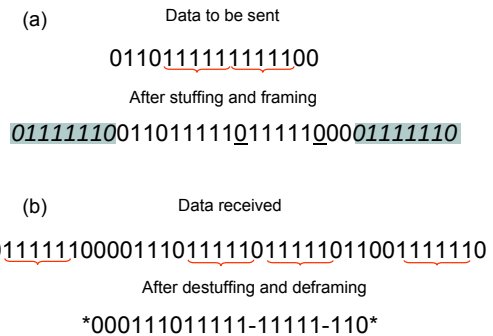


HDLC frame

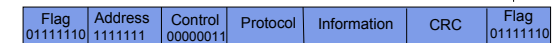


- Frame delineated by flag character
- HDLC uses *bit stuffing* to prevent occurrence of flag 01111110 inside the frame
- Transmitter inserts extra 0 after each consecutive five 1s *inside* the frame
- Receiver checks for five consecutive 1s
 - if next bit = 0, it is removed
 - if next two bits are 10, then flag is detected
 - If next two bits are 11, then frame has errors

Example: Bit stuffing & de-stuffing



PPP Frame



All stations are to accept the frame

Unnumbered frame

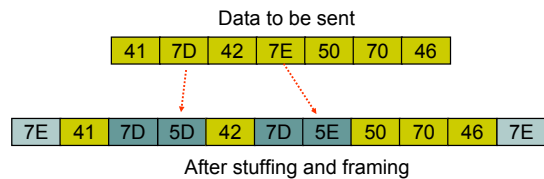
integer # of bytes

Specifies what kind of packet is contained in the payload, e.g., LCP, NCP, IP, OSI CLNP, IPX

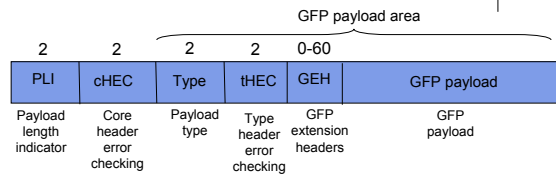
- PPP uses similar frame structure as HDLC, except
 - Protocol type field
 - Payload contains an *integer* number of bytes
- PPP uses the same flag, but uses *byte stuffing*
- Problems with PPP byte stuffing
 - Size of frame varies unpredictably due to byte insertion
 - Malicious users can inflate bandwidth by inserting 7D & 7E

Byte-Stuffing in PPP

- PPP is character-oriented version of HDLC
- Flag is 0x7E (01111110)
- Control escape 0x7D (01111101)
- Any occurrence of flag or control escape inside of frame is replaced with 0x7D followed by original octet XORed with 0x20 (00100000)



Generic Framing Procedure



- GFP combines frame length indication with CRC
 - PLI indicated length of frame, then simply count characters
 - cHEC (CRC-16) protects against errors in count field (single-bit error correction + error detection)
- GFP designed to operate over octet-synchronous physical layers (e.g. SONET)
 - Frame-mapped mode for variable-length payloads: Ethernet
 - Transparent mode carries fixed-length payload: storage devices

GFP Synchronization & Scrambling

- Synchronization in three-states
 - *Hunt state*: examine 4-bytes to see if CRC ok
 - If no, move forward by one-byte
 - If yes, move to pre-sync state
 - *Pre-sync state*: tentative PLI indicates next frame
 - If N successful frame detections, move to sync state
 - If no match, go to hunt state
 - *Sync state*: normal state
 - Validate PLI/cHEC, extract payload, go to next frame
 - Use single-error correction
 - Go to hunt state if non-correctable error
- Scrambling
 - Payload is scrambled to prevent malicious users from inserting long strings of 0s which cause SONET equipment to lose bit clock synchronization (as discussed in line code section)

Lecture 5 Peer-to-Peer Protocols and Data Link Layer



Point-to-Point Protocol

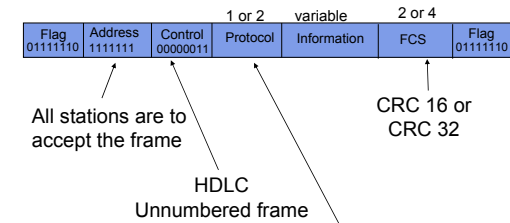
PPP: Point-to-Point Protocol

- Data link protocol for point-to-point lines in Internet
 - Router-router; dial-up to router
- 1. Provides *Framing and Error Detection*
 - Character-oriented HDLC-like frame structure
- 2. *Link Control Protocol*
 - Bringing up, testing, bringing down lines; negotiating options
 - Authentication**: key capability in ISP access
- 3. A family of *Network Control Protocols* specific to different network layer protocols
 - IP, OSI network layer, IPX (Novell), Appletalk

PPP Applications

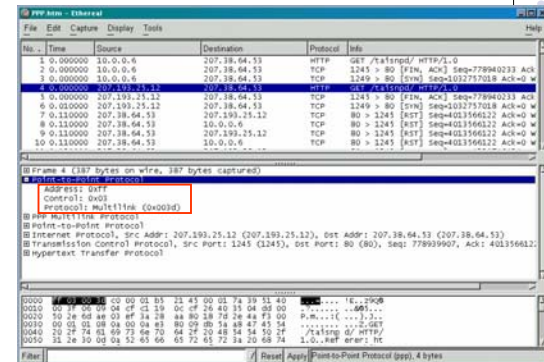
- PPP used in many point-to-point applications
- Telephone Modem Links 30 kbps
 - Packet over SONET 600 Mbps to 10 Gbps
 - IP→PPP→SONET
 - PPP is also used over shared links such as Ethernet to provide LCP, NCP, and authentication features
 - PPP over Ethernet (RFC 2516)
 - Used over DSL

PPP Frame Format

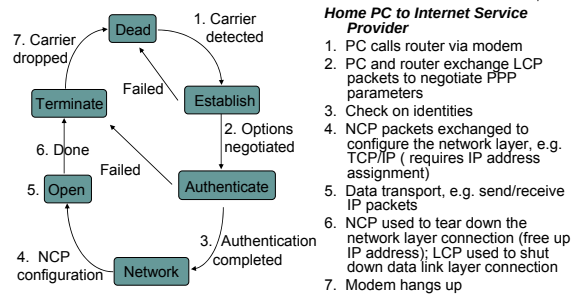


- PPP can support multiple network protocols simultaneously
- Specifies what kind of packet is contained in the payload
 - e.g. LCP, NCP, IP, OSI CLNP, IPX...

PPP Example



PPP Phases



PPP Authentication

- Password Authentication Protocol
 - Initiator must send ID & password
 - Authenticator replies with authentication success/fail
 - After several attempts, LCP closes link
 - Transmitted unencrypted, susceptible to eavesdropping
- Challenge-Handshake Authentication Protocol (CHAP)
 - Initiator & authenticator share a secret key
 - Authenticator sends a challenge (random # & ID)
 - Initiator computes cryptographic checksum of random # & ID using the shared secret key
 - Authenticator also calculates cryptographic checksum & compares to response
 - Authenticator can reissue challenge during session

Example: PPP connection setup in dialup modem to ISP

No.	Time	Source	Destination	Protocol	Info
1	0.000000	2013:45:44:44:44:44	2013:45:44:44:44:44	PPP LCP	PPP LCP Configuration Request
2	0.000026	2013:45:44:44:44:44	2013:45:44:44:44:44	PPP LCP	PPP LCP Configuration Request
3	0.000040	2013:45:44:44:44:44	2013:45:44:44:44:44	PPP LCP	PPP LCP Configuration Request
4	0.000054	2013:45:44:44:44:44	2013:45:44:44:44:44	PPP LCP	PPP LCP Configuration Request
5	0.000068	2013:45:44:44:44:44	2013:45:44:44:44:44	PPP LCP	PPP LCP Configuration Request
6	0.000082	2013:45:44:44:44:44	2013:45:44:44:44:44	PPP LCP	PPP LCP Configuration Request
7	0.000096	2013:45:44:44:44:44	2013:45:44:44:44:44	PPP LCP	PPP LCP Configuration Request
8	0.000110	2013:45:44:44:44:44	2013:45:44:44:44:44	PPP LCP	PPP LCP Configuration Request
9	0.000124	2013:45:44:44:44:44	2013:45:44:44:44:44	PPP LCP	PPP LCP Configuration Request
10	0.000138	2013:45:44:44:44:44	2013:45:44:44:44:44	PPP LCP	PPP LCP Configuration Request
11	0.000152	2013:45:44:44:44:44	2013:45:44:44:44:44	PPP LCP	PPP LCP Configuration Request
12	0.000166	2013:45:44:44:44:44	2013:45:44:44:44:44	PPP LCP	PPP LCP Configuration Request
13	0.000180	2013:45:44:44:44:44	2013:45:44:44:44:44	PPP LCP	PPP LCP Configuration Request
14	0.000194	2013:45:44:44:44:44	2013:45:44:44:44:44	PPP LCP	PPP LCP Configuration Request
15	0.000208	2013:45:44:44:44:44	2013:45:44:44:44:44	PPP LCP	PPP LCP Configuration Request
16	0.000222	2013:45:44:44:44:44	2013:45:44:44:44:44	PPP LCP	PPP LCP Configuration Request
17	0.000236	2013:45:44:44:44:44	2013:45:44:44:44:44	PPP LCP	PPP LCP Configuration Request
18	0.000250	2013:45:44:44:44:44	2013:45:44:44:44:44	PPP LCP	PPP LCP Configuration Request
19	0.000264	2013:45:44:44:44:44	2013:45:44:44:44:44	PPP LCP	PPP LCP Configuration Request
20	0.000278	2013:45:44:44:44:44	2013:45:44:44:44:44	PPP LCP	PPP LCP Configuration Request

Frame 9 (128 bytes on wire) (96 bytes captured) on interface 0: Ethernet II, Src: 2013:45:44:44:44:44, Dst: 2013:45:44:44:44:44
 Encapsulated: LCP (0x0021)
 Code: Configuration Ack (0x0021)
 Identifier: 0x00000000
 Length: 12
 Options: LCP bytes
 Async Control Character Map: 0x00000000 (DC1 (DCH), DC3 (DORP))
 Authentication protocol: Password Authentication Protocol (0x0023)
 Magic number: 0x00000000
 Protocol field compression
 Address/control field compression

LCP Setup
PAP
IP NCP setup

Lecture 5 Peer-to-Peer Protocols and Data Link Layer

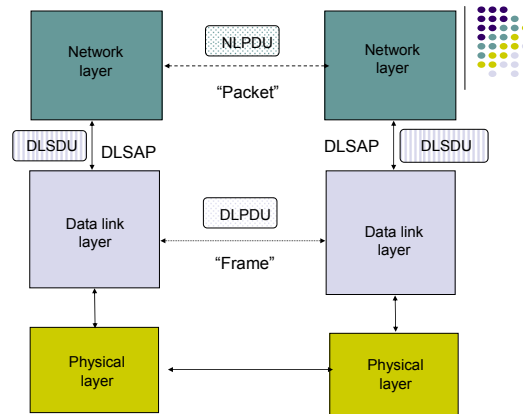


High-Level Data Link Control



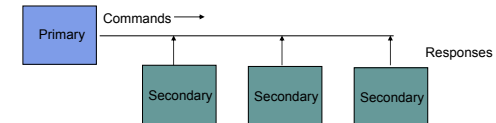
High-Level Data Link Control (HDLC)

- Bit-oriented data link control
- Derived from IBM Synchronous Data Link Control (SDLC)
- Related to Link Access Procedure Balanced (LAPB)
 - LAPD in ISDN
 - LAPM in cellular telephone signaling

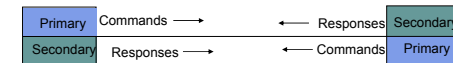


HDLC Data Transfer Modes

- Normal Response Mode
 - Used in polling multidrop lines

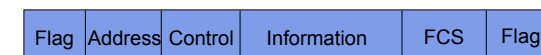


- Asynchronous Balanced Mode
 - Used in full-duplex point-to-point links



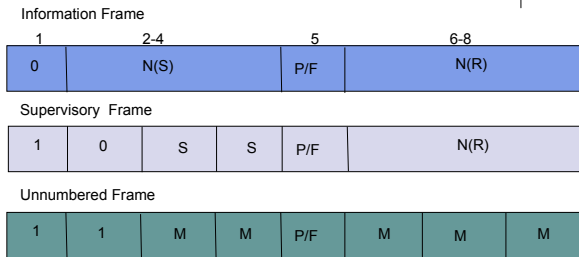
- Mode is selected during connection establishment

HDLC Frame Format



- Control field gives HDLC its functionality
- Codes in fields have specific meanings and uses
 - Flag: delineate frame boundaries
 - Address: identify *secondary* station (1 or more octets)
 - In ABM mode, a station can act as primary or secondary so address changes accordingly
 - Control: purpose & functions of frame (1 or 2 octets)
 - Information: contains user data; length not standardized, but implementations impose maximum
 - Frame Check Sequence: 16- or 32-bit CRC

Control Field Format



- S: Supervisory Function Bits
- N(R): Receive Sequence Number
- N(S): Send Sequence Number
- M: Unnumbered Function Bits
- P/F: Poll/final bit used in interaction between primary and secondary

Information frames

- Each I-frame contains sequence number N(S)
- Positive ACK piggybacked
 - N(R)=Sequence number of *next* frame expected acknowledges all frames up to and including N(R)-1
- 3 or 7 bit sequence numbering
 - Maximum window sizes 7 or 127
- Poll/Final Bit
 - NRM: Primary polls station by setting P=1; Secondary sets F=1 in *last* I-frame in response
 - Primaries and secondaries always interact via *paired* P/F bits

Error Detection & Loss Recovery

- Frames lost due to loss-of-synch or receiver buffer overflow
- Frames may undergo errors in transmission
- CRCs detect errors and such frames are treated as lost
- Recovery through ACKs, timeouts & retransmission
- Sequence numbering to identify out-of-sequence & duplicate frames
- HDLC provides for options that implement several ARQ methods

Supervisory frames

Used for error (ACK, NAK) and flow control (Don't Send):

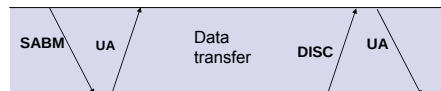
- *Receive Ready* (RR), SS=00
 - ACKs frames up to N(R)-1 when piggyback not available
- *REJECT* (REJ), SS=01
 - Negative ACK indicating N(R) is first frame not received correctly. Transmitter must resend N(R) and later frames
- *Receive Not Ready* (RNR), SS=10
 - ACKs frame N(R)-1 & requests that no more I-frames be sent
- *Selective REJECT* (SREJ), SS=11
 - Negative ACK for N(R) requesting that N(R) be selectively retransmitted

Unnumbered Frames

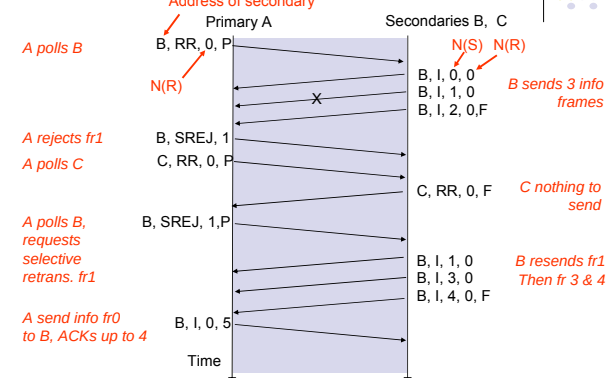
- Setting of Modes:
 - SABM: Set Asynchronous Balanced Mode
 - UA: acknowledges acceptance of mode setting commands
 - DISC: terminates logical link connectio
- Information Transfer between stations
 - UI: Unnumbered information
- Recovery used when normal error/flow control fails
 - FRMR: frame with correct FCS but impossible semantics
 - RSET: indicates sending station is resetting sequence numbers
- XID: exchange station id and characteristics

Connection Establishment & Release

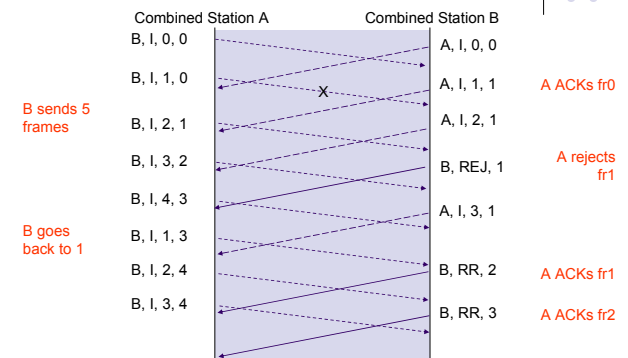
- Supervisory frames used to establish and release data link connection
- In HDLC
 - Set Asynchronous Balanced Mode (SABM)
 - Disconnect (DISC)
 - Unnumbered Acknowledgment (UA)



Example: HDLC using NRM (polling)

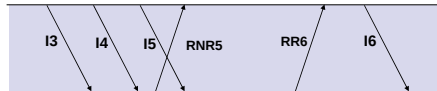


Frame Exchange using Asynchronous Balanced Mode



Flow Control

- Flow control is required to prevent transmitter from overrunning receiver buffers
- Receiver can control flow by delaying acknowledgement messages
- Receiver can also use supervisory frames to explicitly control transmitter
 - Receive Not Ready (RNR) & Receive Ready (RR)



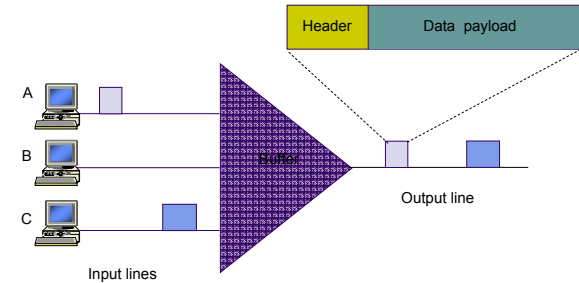
Lecture 5 Peer-to-Peer Protocols and Data Link Layer

Link Sharing Using Statistical Multiplexing

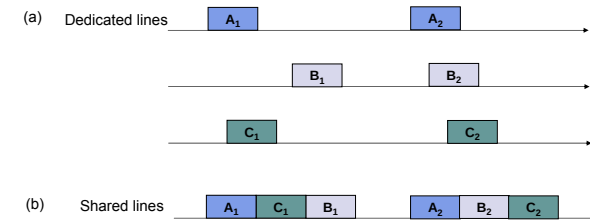


Statistical Multiplexing

- Multiplexing concentrates bursty traffic onto a shared line
- Greater efficiency and lower cost

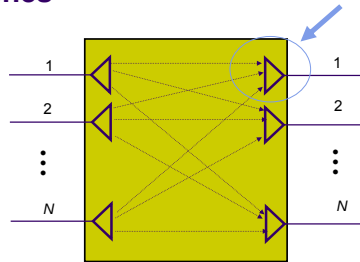


Tradeoff Delay for Efficiency



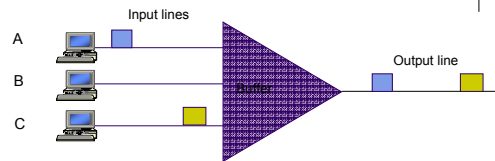
- Dedicated lines involve not waiting for other users, but lines are used inefficiently when user traffic is bursty
- Shared lines concentrate packets into shared line; packets buffered (delayed) when line is not immediately available

Multiplexers inherent in Packet Switches



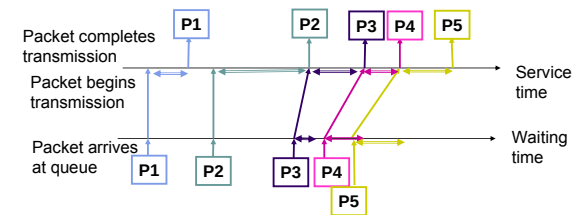
- Packets/frames forwarded to buffer prior to transmission from switch
- Multiplexing occurs in these buffers

Multiplexer Modeling



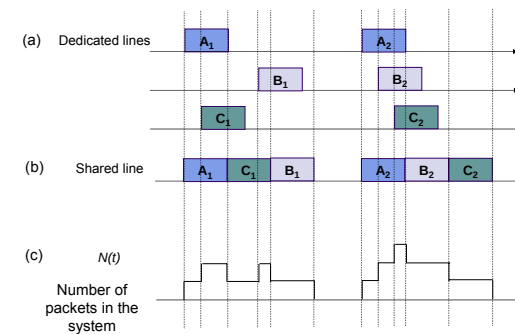
- Arrivals: What is the packet interarrival pattern?
- Service Time: How long are the packets?
- Service Discipline: What is order of transmission?
- Buffer Discipline: If buffer is full, which packet is dropped?
- Performance Measures:
 - Delay Distribution; Packet Loss Probability; Line Utilization

Delay = Waiting + Service Times



- Packets arrive and wait for service
- Waiting Time: from arrival instant to beginning of service
- Service Time: time to transmit packet
- Delay: total time in system = waiting time + service time

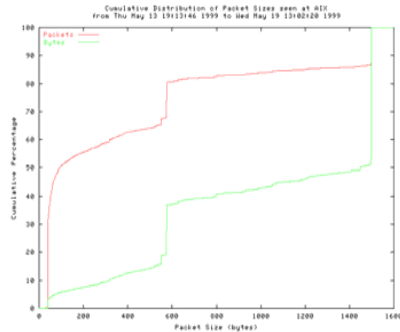
Fluctuations in Packets in the System



Packet Lengths & Service Times

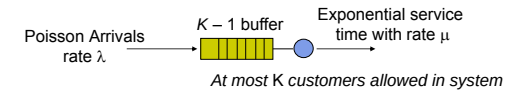
- R bits per second transmission rate
- L = # bits in a packet
- $X = L/R$ = time to transmit ("service") a packet
- Packet lengths are usually variable
 - Distribution of lengths $\rightarrow$ Dist. of service times
 - Common models:
 - Constant packet length (all the same)
 - Exponential distribution
 - Internet Measured Distributions fairly constant
 - See next chart

Measure Internet Packet Distribution



- Dominated by TCP traffic (85%)
- ~40% packets are minimum-sized 40 byte packets for TCP ACKs
- ~15% packets are maximum-sized Ethernet 1500 frames
- ~15% packets are 552 & 576 byte packets for TCP implementations that do not use path MTU discovery
- Mean=413 bytes
- Stand Dev=509 bytes
- Source: caida.org

M/M/1/K Queueing Model



- 1 customer served at a time; up to $K-1$ can wait in queue
- Mean service time $E[X] = 1/\mu$
- Key parameter Load: $\rho = \lambda/\mu$
- When $\lambda \ll \mu$ ($\rho \approx 0$), customers arrive infrequently and usually find system empty, so delay is low and loss is unlikely
- As λ approaches μ ($\rho \rightarrow 1$), customers start bunching up and delays increase and losses occur more frequently
- When $\lambda > \mu$ ($\rho > 1$), customers arrive faster than they can be processed, so most customers find system full and those that do enter have to wait about $K-1$ service times

Poisson Arrivals

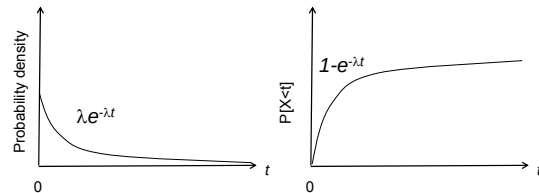
- Average Arrival Rate: λ packets per second
- Arrivals are equally-likely to occur at any point in time
- Time between consecutive arrivals is an exponential random variable with mean $1/\lambda$
- Number of arrivals in interval of time t is a Poisson random variable with mean λt

$$P[k \text{ arrivals in } t \text{ seconds}] = \frac{(\lambda t)^k}{k!} e^{-\lambda t}$$

Exponential Distribution



$$P[X > t] = e^{-t/E[X]} = e^{-\lambda t} \text{ for } t > 0.$$



M/M/1/K Performance Results

(From Appendix A)



Probability of Overflow:

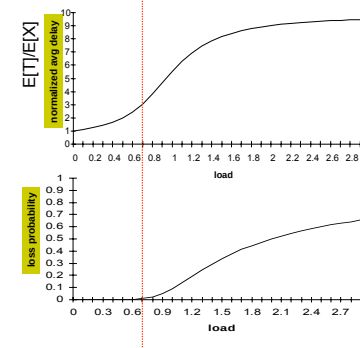
$$P_{loss} = \frac{(1-\rho)\rho^K}{1-\rho^{K+1}}$$

Average Total Packet Delay:

$$E[N] = \frac{\rho}{1-\rho} - \frac{(K+1)\rho^{K+1}}{1-\rho^{K+1}}$$

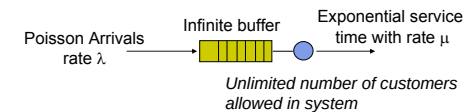
$$E[T] = \frac{E[N]}{\lambda(1-P_K)}$$

M/M/1/10



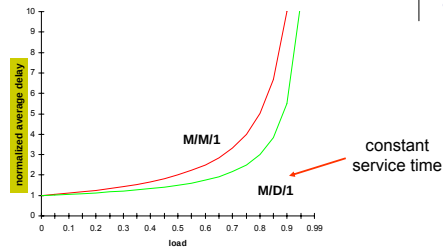
- Maximum 10 packets allowed in system
- Minimum delay is 1 service time
- Maximum delay is 10 service times
- At 70% load delay & loss begin increasing
- What if we add more buffers?

M/M/1 Queue



- $P_b=0$ since customers are never blocked
- Average Time in system $E[T] = E[W] + E[X]$
- When $\lambda \ll \mu$, customers arrive infrequently and delays are low
- As λ approaches μ , customers start bunching up and average delays increase
- When $\lambda > \mu$ customers arrive faster than they can be processed and queue grows without bound (unstable)

Avg. Delay in M/M/1 & M/D/1



$$E[T_M] = \frac{1}{\lambda} \left[\frac{\rho}{1-\rho} \right] = \left[\frac{1}{1-\rho} \right] \frac{1}{\mu} = \left[\frac{\rho}{1-\rho} \right] \frac{1}{\mu} + \frac{1}{\mu} \text{ for M/M/1 model.}$$

$$E[T_D] = \left[1 + \frac{\rho}{2(1-\rho)} \right] \frac{1}{\mu} = \left[\frac{\rho}{2(1-\rho)} \right] \frac{1}{\mu} + \frac{1}{\mu} \text{ for M/D/1 system.}$$

Effect of Scale

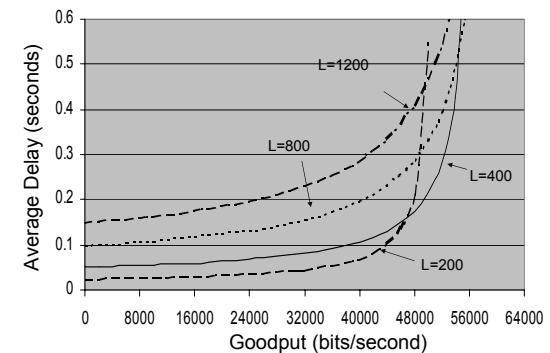
- C = 100,000 bps
- Exp. Dist. with Avg. Packet Length: 10,000 bits
- Service Time: X=0.1 second
- Arrival Rate: 7.5 pkts/sec
- Load: $\rho=0.75$
- Mean Delay: $E[T] = 0.1/(1-.75) = 0.4 \text{ sec}$
- C = 10,000,000 bps
- Exp. Dist. with Avg. Packet Length: 10,000 bits
- Service Time: X=0.001 second
- Arrival Rate: 750 pkts/sec
- Load: $\rho=0.75$
- Mean Delay: $E[T] = 0.001/(1-.75) = 0.004 \text{ sec}$
- Reduction by factor of 100

Aggregation of flows can improve Delay & Loss Performance

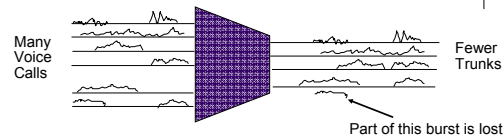
Example: Header overhead & Goodput

- Let R=64 kbps
- Assume IP+TCP header = 40 bytes
- Assume constant packets of total length
 - L= 200, 400, 800, 1200 bytes
- Find avg. delay vs. goodput (information transmitted excluding header overhead)
- Service rate $\mu = 64000/8L \text{ packets/second}$
- Total load $\rho = \lambda 8L/64000$
- Goodput = $\lambda \text{ packets/sec} \times 8(L-40)$

Header overhead limits maximum goodput

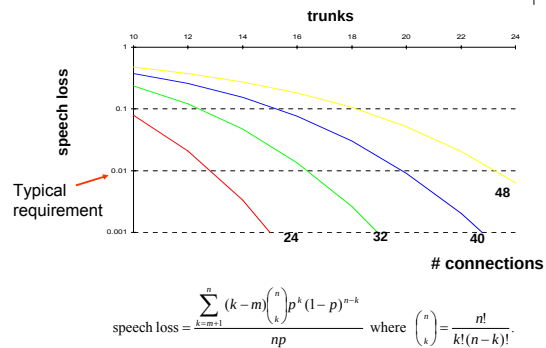


Burst Multiplexing / Speech Interpolation



- Voice active < 40% time
- **No buffering**, on-the-fly switch bursts to available trunks
- Can handle 2 to 3 times as many calls
- Tradeoff: Trunk Utilization vs. Speech Loss
 - Fractional Speech Loss: fraction of active speech lost
- Demand Characteristics
 - Talkspurt and Silence Duration Statistics
 - Proportion of time speaker active/idle

Speech Loss vs. Trunks



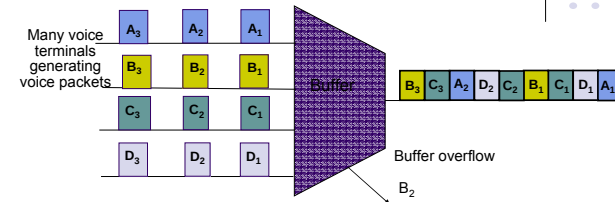
Effect of Scale

- Larger flows lead to better performance
- Multiplexing Gain = # speakers / # trunks

Trunks required for 1% speech loss

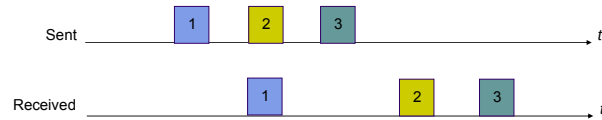
Speakers	Trunks	Multiplexing Gain	Utilization
24	13	1.85	0.74
32	16	2.00	0.80
40	20	2.00	0.80
48	23	2.09	0.83

Packet Speech Multiplexing



- Digital speech carried by fixed-length packets
- No packets when speaker silent
- Synchronous packets when speaker active
- Buffer packets & transmit over shared high-speed line
- Tradeoffs: Utilization vs. Delay/Jitter & Loss

Packet Switching of Voice



- Packetization delay: time for speech samples to fill a packet
- Jitter: variable inter-packet arrivals at destination
- Playback strategies required to compensate for jitter/loss
 - Flexible delay inserted to produce fixed end-to-end delay
 - Need buffer overflow/underflow countermeasures
 - Need clock recovery algorithm

Lecture 5 Peer-to-Peer Protocols and Data Link Layer



ARQ Efficiency Calculations



Stop & Wait Performance



1 successful transmission $i - 1$ unsuccessful transmissions

$$\begin{aligned}
 E[t_{total}] &= t_0 + \sum_{i=1}^{\infty} (i-1) t_{out} P[n_i = i] \\
 &= t_0 + \sum_{i=1}^{\infty} (i-1) t_{out} (1-P_f)^{i-1} P_f \\
 &= t_0 + \frac{t_{out} P_f}{1-P_f} = t_0 \frac{1}{1-P_f}
 \end{aligned}$$

Efficiency:

$$\eta_{SW} = \frac{n_f - n_o}{R} = (1-P_f) \frac{1 - \frac{n_o}{n_f}}{1 + \frac{n_o}{n_f} + \frac{2(t_{prop} + t_{proc})R}{n_f}} = (1-P_f) \eta_0$$

Go-Back-N Performance



1 successful transmission $i - 1$ unsuccessful transmissions

$$\begin{aligned}
 E[t_{total}] &= t_f + \sum_{i=1}^{\infty} (i-1) W_s t_f P[n_i = i] \\
 &= t_f + W_s t_f \sum_{i=1}^{\infty} (i-1) (1-P_f)^{i-1} P_f \\
 &= t_f + \frac{W_s t_f P_f}{1-P_f} = t_f \frac{1 + (W_s - 1) P_f}{1-P_f}
 \end{aligned}$$

Efficiency:

$$\eta_{GBN} = \frac{n_f - n_o}{R} = (1-P_f) \frac{1 - \frac{n_o}{n_f}}{1 + (W_s - 1) P_f}$$