King Fahd University of Petroleum & Minerals Computer Engineering Dept

COE 541 – Design and Analysis of Local Area Networks

Term 041

Dr. Ashraf S. Hasan Mahmoud

Rm 22-148-3

Ext. 1724

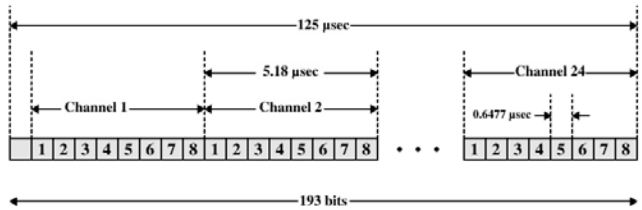
Email: ashraf@ccse.kfupm.edu.sa

Fixed Assignment Access

- **Schemes:**
  - **Time Division Multiple Access (Time)**
  - **Frequency Division Multiple Access (BW)**
  - **Code Division Multiple Access (Code)**
- **Access to common channel is independent of user demand – static and predetermined**
  - **Contrast to asynchronous time multiplexing**

TDM – Example2: Digital Carrier Systems

- Voice call is PCM coded ➔ 8 b/sample
- DS-0: PCM digitized voice call – R = 64 Kb/s
- Group 24 digitized voice calls into one frame as shown in figure ➔DS-1: 24 DS-0s
- Note channel 1 has all 1st bits from all of 24 calls; channel 2 has all 2nd bits from all 24 calls; etc.



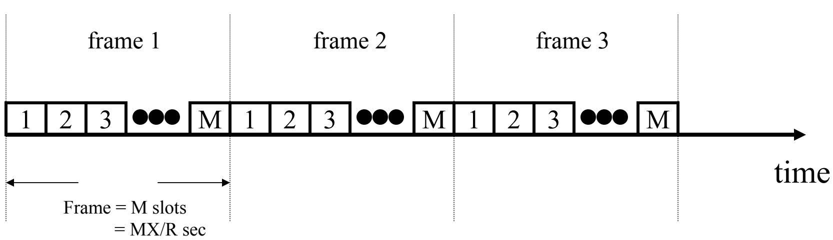
Notes:

1. The first bit is a framing bit, used for synchronization.
2. Voice channels:
   - •8-bit PCM used on five of six frames.
   - •7-bit PCM used on every sixth frame; bit 8 of each channel is a signaling bit.
3. Data channels:
   - •Channel 24 is used for signaling only in some schemes.
   - •Bits 1-7 used for 56 kbps service
   - •Bits 2-7 used for 9.6, 4.8, and 2.4 kbps service.

Figure 8.9 DS-1 Transmission Format

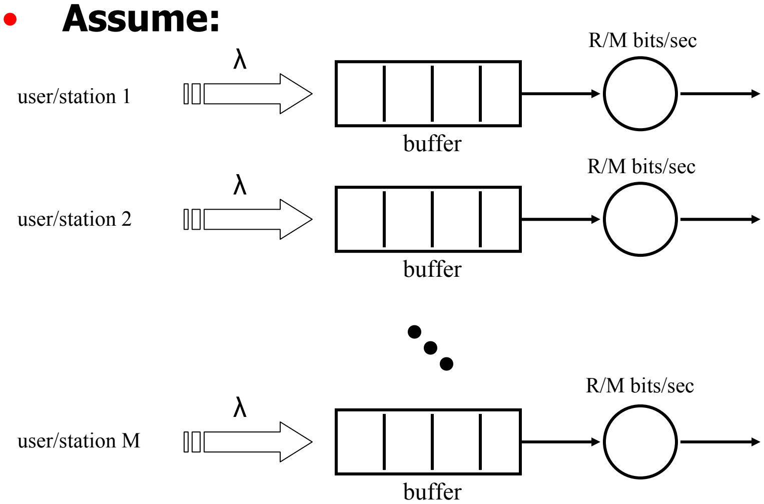
TDMA

- **Assume:**
  - **M users/stations**
  - **Channel of bit rate = R b/s**
  - **Fixed packet size = X bits/packet**
  - **Packet arrival = λ packet/sec**



TDMA – Queueing Model

- **Assume:**



TDMA – Total Delay Analysis

- **Total Packet (Burst) delay:**
  - **Slot Synchronization Delay – Avg = ½ frame duration, plus**
  - **Queueing Delay, plus**
  - **Packet transmission**
- **Slot Synchronization = X M/(2R)**
- **Packet transmission = X/R**
- **Queueing Delay = ?**

TDMA – Packet (Burst) Queueing Delay

- **Each channel can be modeled as an M/D/1 queue**
- **Consider station/user queues individually**
  - **Poisson arrival of packets of rate λ**
  - **Service time – fixed (packet size is fixed and so is the transmission rate)**
    - **From point of view of user queue – packet is transmitted at rate of R/M bits/seconds**
    - **Packet transmission time = X/(R/M) or MX/R seconds**

TDMA – Packet (Burst) Queueing Delay – cont'd

- **For M/D/1 (refer to M/G/1 slides):**

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

$E[\tau]$ is the packet service time = MX/R

$\rho = \lambda E[\tau] = \lambda MX/R$

- **Therefore, the mean waiting time can be written as**

$$E[W] = \frac{\rho}{2(1-\rho)} \frac{MX}{R}$$

TDMA – Throughput

- **Throughput: average number of useful (good) packets transmission per time unit**
- **Each station transmits λ X/R packets per time unit ➔ Station throughput = λ X/R**
- **The M stations community throughput = M λ X/R**
- **Total Throughput, S = M λ X/R**
  = ρ

TDMA – Total Delay

- **Total Delay, T**

$$T = \frac{X}{R} + \frac{MX}{2R} + \frac{MS}{2(1-S)}\frac{X}{R}$$

Normalizing total delay with respect to packet transmission time ➔

$$\check{T} = 1 + \frac{M}{2} + \frac{MS}{2(1-S)}$$

FDMA – Total Delay

- **Assume same traffic parameters (for comparison reasons)**
- **No slot synchronization time – transmission can be always on**
- **Total Packet (Burst) delay:**
  - **Queueing Delay, plus**
  - **Packet transmission**

FDMA – Total Delay – cont'd

- **Total delay, T**

$$T = \frac{M X}{R} + \frac{S}{2(1 - S)} \frac{M X}{R}$$

- **Or**

$$\check{T} = \frac{M(2 - S)}{2(1 - S)}$$

TDMA versus FDMA – Total Delay

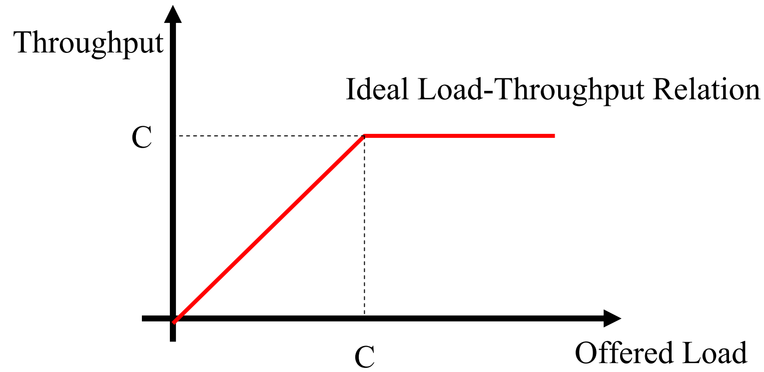
- **Using the previous relations,**

$$\check{T}_{FDMA} = \check{T}_{TDMA} + M/2 - 1$$

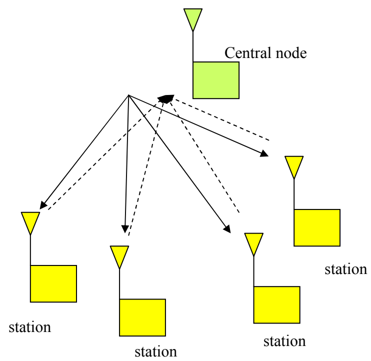
- **i.e. total delay for FDMA is always greater than that for TDMA except for M = 2**

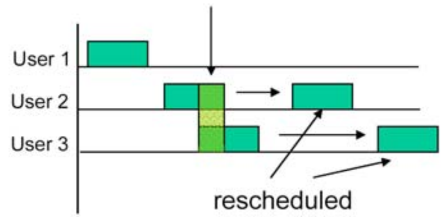
Performance Measures

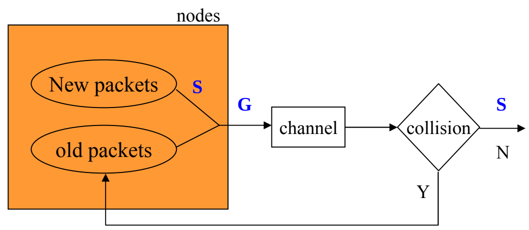
- Throughput
- Delay (packet)



Pure ALOHA



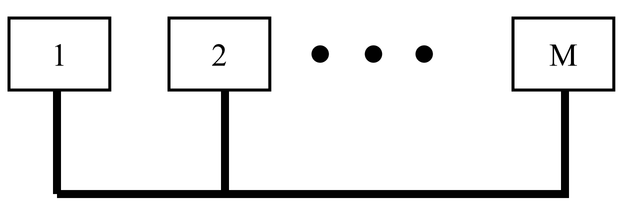




Uplink carrier 413 kHz, 9.6 kb/s
Downlink carrier 407 kHz, 9.6 kb/s

ALOHA Random Access Procedure

- **Assume**
  - **Packet transmission time: P**
  - **Total # of packet arrival (new + retransmission) ~ Poisson with rate λ**



ALOHA - Throughput

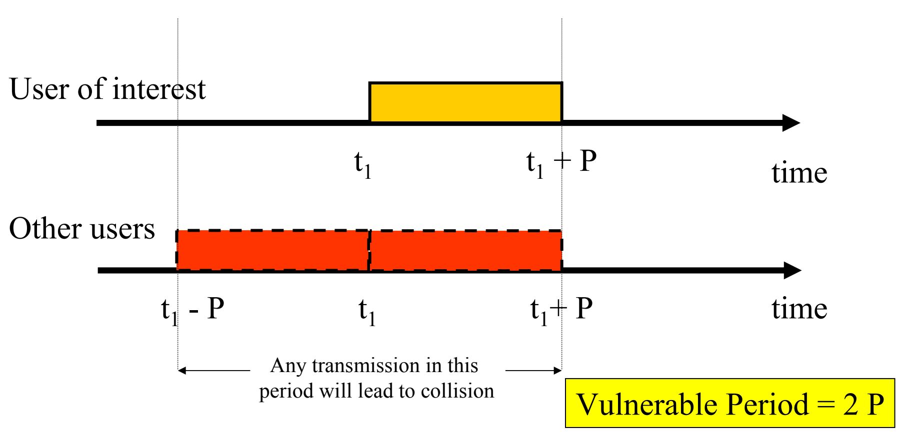
- **Poisson arrival (new + retransmitted) of packets:**

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

- **Offered Load (G): Average number of attempted packet transmissions per packet transmission time, P**
- **Throughput (S): Average number of successful transmissions per packet transmission time, P**

Pure ALOHA – Throughput – cont'd

- **Vulnerable Period**



ALOHA Throughput – cont'd

- Throughput = fraction of attempted transmission that are successful (i.e. did not collide)
- Therefore,

$$S = G \times \text{Prob[no collision in 2 P seconds]}$$
$$= G \times \text{Prob[0 packet arrivals in 2 P seconds]}$$

$$= G \times \frac{(2G)^0}{0!} e^{-2G}$$

Or

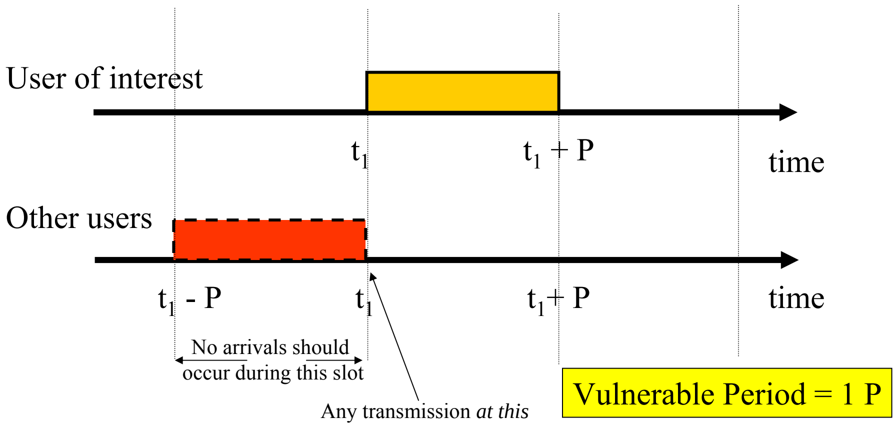
$$S = G e^{-2G}$$ packets/packet transmission time

Slotted ALOHA

- An improvement over pure ALOHA
- Time axis is slotted
- Transmission occur only at the beginning of a time slot
- A packet arriving to buffer has to wait till the beginning of the time slot for transmission
- Cost: common clock signal!

Slotted ALOHA – Throughput – cont'd

- **Vulnerable Period (note time axis is divided into slots – transmissions can only start at the beginning of a time slot)**



Slotted ALOHA – Throughput

- **Throughput = fraction of attempted transmission that are successful (i.e. did not collide)**
- **Therefore,**

S = G X Prob[no collision in 1 P seconds]
= G X Prob[0 packet arrivals in 1 P seconds]

$$= G \times \frac{(G)^0}{0!} e^{-1G}$$

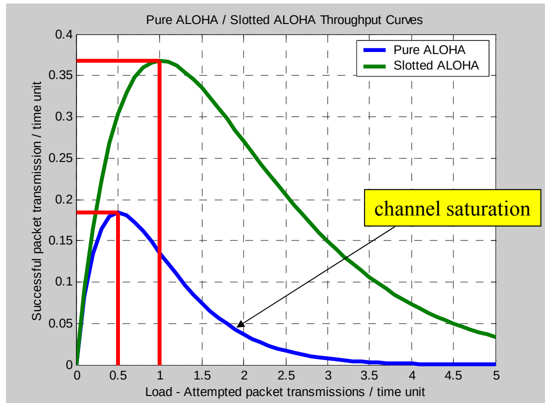
Or

$S = G e^{-1G}$ **packets/packet transmission time**

ALOHA – Throughput – cont'd

- Pure ALOHA: Max throughput, $S = 0.5\,e^{-1}$ or ~ 18% at G = ½
- Slotted ALOHA: Max throughput, $S = e^{-1}$ or ~ 36% at G = 1

- **For Pure ALOHA:**
  - Stable operation range: 0 < G < 0.5
  - Unstable operation range: G > 0.5
- **For Slotted ALOHA:**
  - Stable operation range: 0 < G < 1.0
  - Unstable operation range: G > 1.0



Pure ALOHA – (Approximate) Delay Analysis

- **Average number of attempts per successfully transmitted packet = G/S**
- **From throughput relation,**

$$G/S = e^{2G}$$

- **Therefore, average number unsuccessful attempts = G/S – 1**

$$= e^{2G} - 1$$

Pure ALOHA – (Approximate) Delay Analysis – cont'd

- **Cost for each collision**
  - **Backoff time - assume duration B on average**
  - **Retransmission**
- **Therefore, total delay, T**

$$T = P + (e^{2G} - 1)(P + B)$$

- **Normalizing the total delay yields,**

$$\check{T} = 1 + (e^{2G} - 1)(1 + B/P)$$

Example 1

- **Problem:** A centralized network providing a maximum of 10 Mbps and services a large set of user terminal with pure ALOHA protocol

a) What is the maximum throughput for network?

b) What is the offered traffic in the medium and how is it composed?

c) If a packet length is 64KBytes, what is the average packet delay? Assume average backoff time = 1 second.

Example 1 – cont'd

- **Solution**:

a) Smax = 18%
==> Network throughput = 0.18 X 10 = 1.8 Mbps

b) At S = Smax, G = 0.5,
Offered load = 0.5 X 10 = 5 Mbps
Composition of load: 1.8 Mbps of delivered packets
+ 3.2 Mbps of collided packets

c) Packet transmission time P = 64X1024X8 bits/10 Mb/s
= 6.6 msec

$T = P + (e^{2G} - 1)(P + B)$
$= 6.6 + (e^{1} - 1)(6.6 + 1000)$
$= 1736$

Notes On ALOHA Analysis

- **Slotted ALOHA: a modified ALOHA protocol to allow stations to transmit only at known and fixed time instances.**
  - **Time axis is divided into slots – stations can transmit only at the beginning of a time slot**
- **What is the vulnerable period for slotted ALOHA?**
- **Derive the throughput and delay relationship for this protocol?**

Idealized Central Control

- **Idealized = ZERO cost for transfer of channel from one state to another under central node**
  - **Whenever a station has data to transmit, controller knows instantly and the channel assignment is immediate**
  - **Packets arriving while channel is busy are queued (infinite buffer)**
  - **If two stations have queues packets, the one with first arrival is chosen**

Idealized Central Control - Analysis

- **Assumptions (same as before):**
  - Arrival at each station ~ Poisson of λ packets/sec
  - Packets have constant length of X bits
  - M stations
  - Channel bit rate = R b/s
  - Propagation and processing times ≈ 0

Idealized Central Control – Analysis – cont'd

- Total input = M λ packets / second
- Since "no cost" for transfer of channel ==> distributed network behaves like a single queue
- Over all throughput (utilization) is given by

$$\rho = M \lambda * (X/R)$$

where M λ is the total arrival rate to this single queue, and
X/R is the service time

Idealized Central Control – Analysis – cont'd

- This single queue – M/D/1
- Mean number of queued packets ($E[N_q]$) and mean waiting time ($E[W]$)

$$N_q = \frac{\rho^2}{2(1-\rho)}, \quad W = \frac{\rho}{2(1-\rho)} \frac{X}{R}$$

- Therefore, total delay, T, is given by

$$T = \frac{X}{R} + \frac{\rho}{2(1-\rho)} \frac{X}{R}$$

Idealized Central Control – Analysis – cont'd

- **Since there are no collisions ==> throughput = utilization or S = ρ**
- **Hence, total delay is given by**

$$T = \frac{X}{R} + \frac{S}{2(1-S)}\frac{X}{R}$$

Or

$$\check{T} = 1 + \frac{S}{2(1-S)}$$

$$= \frac{(2-S)}{2(1-S)}$$

Idealized Central Control – Analysis – cont'd

- **Also, E[Nq] is given by**

$$E[N_q] = \frac{S^2}{2(1-S)}$$

- **Per station throughput = S/M**
- **Per station n_q = $E[N_q]$ / M**

Polling Networks

- **Central Control Networks: a central node arbitrates access to the network**
- **The access order is predetermined – under the control of the central node**
- **Access is granted when station is polled – Full rate of channel is used**
  - **Stations accumulate traffic in their buffers**
  - **Transmit when given permission (polled)**

Operation Modes

- **Two Modes:**
  1. **Roll-Call**
  2. **Hub polling**
- **For the two modes, the opportunity to transmit is symmetrically rotated from one station to another**

Operation Modes – Roll-Call

- **Central node initiates polling sequence by sending polling message to chosen station**
- **Polled station transmits traffic (if any)**
- **Transmitting station informs central node of transmission end (field in the last transmitted packet)**
- **Central node polls next station in-line**
- **Process repeats**

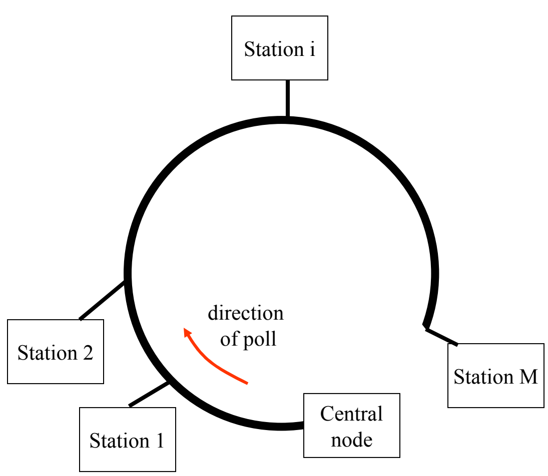
Operation Modes – Hub polling

- **Central node initiates polling sequence by sending polling message to chosen station**
- **Polled station transmits traffic (if any)**
- **Last transmitted packet contains a go-ahead signal with the next inline station address**
- **Next inline station (which is continuously monitoring traffic) identifies its address and starts transmitting (if there is traffic in buffer) immediately**

Roll-Call vs Hub based

- **Response time**
- **Complexity – Cost**
- **?**

Logical Structure



Performance Analysis

- **Assume:**
  - **Arrival process is Poisson with rate λ packets/sec**
  - **The walk time, w, between station stations is constant**
    - **Includes processing and propagation time**
  - **Average packet length = Xavg bits/packet**
    - **Will consider fixed and exponentially distributed packet sizes**
  - **Common channel (server) rate = R b/s**

Performance Analysis – cont'd

- **Cycle Time, Tc:**
  - **Total time to poll each station and return to the starting station in the polling sequence**
  - **Random variable**
    - **Amount of data transmitted by each station is random**
- **Other performance measures:**
  - **Average queue length, N, in station (packets)**
  - **Average time, W, that packets wait in the station buffer before being transmitted**
  - **Average transfer delay, T, from packet entry into station buffer till delivery to central node**

Cycle Time

- **Let Nm be the average number of packets stored in station buffer**
  - **Nm includes packets arriving to buffer while station is in service**
- **Time to empty buffer = Nm Xavg /R**
- **Cycle Time, Tc**

$$Tc = M [Nm\, Xavg / R + w]$$

Cycle Time – cont'd

- **At steady state, Nm is given by**

$$Nm = \lambda\, Tc$$

- **Substituting in the previous equation yields**

$$Tc = \frac{Mw}{1 - M \lambda\, Xavg / R}$$

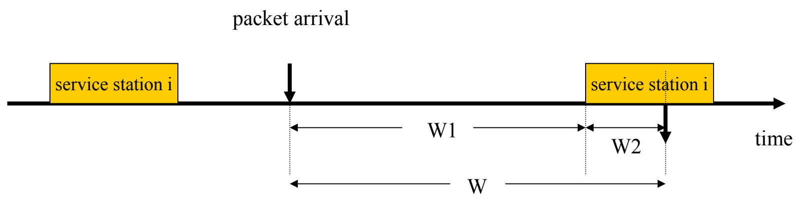
Or

$$Tc = \frac{Mw}{1 - S} \text{ seconds}$$

where
throughput S = (M λ) / (R/Xavg) < 1 or (M λ) < (R/Xavg)!!

Delay Analysis

- **Packet waiting time, W, in queue:**
  - **Waiting time in queue, W1, while other stations are being served, plus**
  - **Waiting time in queue, W2, while its station is being served and till packet reaches head of queue**



Delay Analysis – cont'd

- **Avg number of packets transmitted by station in a cycle: Nm = λ Tc**
  - **remember we serve till buffer is empty**

- **Average service time for station equals to**

  λ Tc Xavg/R

- **Define ρ as**

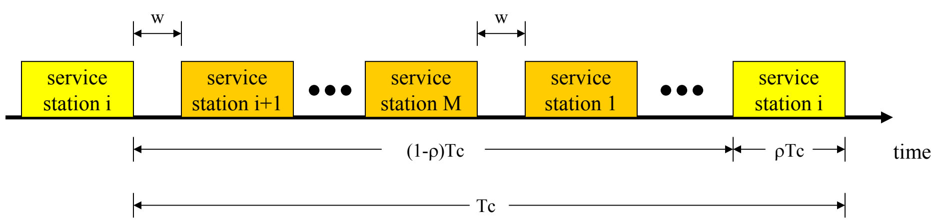
$$\rho = \lambda \text{ Xavg / R}$$

- **Therefore, average service time per station is given by**

  ρ Tc

Cycle for a Polling Network

- **Note the cycle time Tc partitioning**



Delay Analysis – cont'd

- **$(1-\rho)$Tc is the average time station I waits to be served**
  - **Packet arrive at random during $(1-\rho)$Tc**
  - **Average waiting time W1 = $(1-\rho)$Tc/2**
- **Substitute the expression for Tc, yields**

$$W1 = \frac{Mw(1-\rho)}{2(1-M\rho)}$$

It remains to compute W2!

Delay Analysis – cont'd

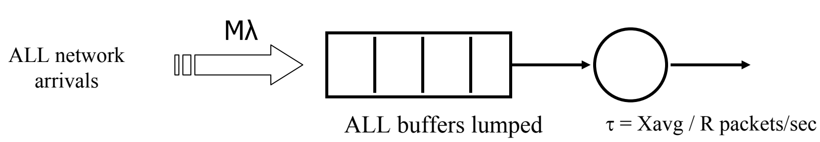
- **Writing W1 in terms of S = M λ Xavg/R**

$$W1 = \frac{Mw(1 - S/M)}{2(1 - S)}$$

It remains to compute W2!

Delay Analysis – cont'd

- **To determine W2 – consider the following equivalent queueing system**
  - **Server never goes idle (no walk time) – switches instantly from one buffer to the next**
  - **All arrivals aggregated**
  - **All buffers lumped**
- **This model: M/G/1**



Delay Analysis – cont'd

- **For an M/G/1 with arrival rate λ and service time, τ: average waiting time, E[W], is given by**

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

- **For our hypothetical queue:**
  - $\lambda \rightarrow M\lambda$
  - $E[\tau] = Xavg/R$; $E[\tau^2] = E[X^2]/R^2$

- **Therefore, W2 is given by**

$$W2 = \frac{M \lambda E[X^2]/R^2}{2(1 - M\rho)}$$

Delay Analysis – cont'd

- **Writing W2 in terms of $S = M \lambda Xavg/R$**

$$W2 = \frac{S \; E[X^2]}{2 \, Xavg \, R (1 - S)}$$

Therefore, overall waiting time for the packet:

$$W = W1 + W2$$

$$W = \frac{Mw(1 - S/M)}{2(1 - S)} + \frac{S \, E[X^2]}{2E[X] \, R (1 - S)}$$

Delay Analysis – Constant Packet Size

- For constant packet size X

- ➔ $E[X] = X$
- ➔ $E[X^2] = X^2$

- Therefore, overall waiting time for the packet:

$$W = \frac{Mw(1 - S/M)}{2(1 - S)} + \frac{S\,E[X]}{2R\,(1 - S)}$$

Delay Analysis – Exponential Packet Size

- For exponentially distributed packet sizes, X

- ➔ $E[X] = Xavg$
- ➔ $E[X^2] = 2\,(Xavg)^2 = 2\,E[X]^2$

- Therefore, overall waiting time for the packet:

$$W = \frac{Mw(1 - S/M)}{2(1 - S)} + \frac{S\,E[X]}{R\,(1 - S)}$$

Example 2:

- **Problem: Consider a metropolitan area network with a single central processor located at the headend of a broadband CATV system that has a tree topology. The following are specified:**
  - **Maximum distance from headend to subscriber station = 20 km**
  - **Access technique – roll-call polling**
  - **Length of polling packet = 8 Bytes**
  - **Length of go-ahead packet = 1 Bytes**
  - **Data rate of channel = 56 kb/s**
  - **Number of subscribers = 1000**
  - **Packet length distribution for packets from subs to headend – exponential**
  - **Mean packet length = 200 Bytes**
  - **Propagation delay = 6 μsec/km**
  - **Modern sync time = 10 msec**

A. **Find the mean waiting delay for arriving packets at the stations if each user generates an average of one packet per minute**

B. **If the channel rate is reduced to 9600 b/s what is the longest possible mean packet length that will not overload the system?**

C. **For mean packet lengths of two-thirds the result of (B) determine the mean waiting delay**

Example 2: cont'd

- **Solution:**

Mean walking time, w:

w = transmission time of go-ahead packet* +
propagation delay from station to headend +
transmission of polling packet +
propagation delay from headend to next station +
modern sync time

One way propagation = 20 X 6 = 120 μsec

Transmission time for go-ahead packet = 1 X 8 /56 = 0.14 msec

Transmission time for polling packet = 8 X 8 / 56 = 1.14 msec

Therefore: w = 0.14 + 2 X 0.120 + 1.14 +10 = 11.52 msec

*This decomposition of the walk time assumes there is a separate go-ahead packet indicating end of traffic condition – alternatively, the last traffic packet could convey the same information by setting a flag

Example 2: cont'd

- **Solution:**

A) Mean waiting delay, W is given by

$$W = \frac{Mw(1 - S/M)}{2(1 - S)} + \frac{S\,E[X]}{R\,(1 - S)}$$

We need to compute S first –
$S = M\,\lambda\,Xavg/R$
$= 1000 \times (1/60) \times 200 \times 8 / 56 = 0.476$

Substituting in the formula for W, yields

$W = 10.99 + 0.026$
$= 11.02$ seconds

Example 2: cont'd

- **Solution:**

B) $S_{max} <= 1 \rightarrow M\,\lambda\,Xavg_{max}/R <= 1$
For R = 9600 b/s
$\rightarrow Xavg_{max} <= R/(M\,\lambda) = 72$ Bytes

C) For $Xavg = 2/3\ Xavg_{max}$
$= 2/3\ (72) = 48$ Bytes
$S = M\,\lambda\,Xavg/R = 0.667$

The new walking time, w is given by

$w = 8X8/9.6 + 1X8/9.6 + 2X0.12 + 10 = 17.74$ msec

Use the new values for S and w and sub in the expression for W

$W = 26.62 + 0.01 = 26.63$ seconds

Average Number of Packets Per Station

- **Using Little's formula:**

$$N = \frac{M \lambda w(1 - S/M)}{2(1 - S)} + \frac{S \lambda E[X^2]}{2E[X]\, R\, (1 - S)}$$

Average Number of Packets Per Station – Constant Packet Size

- **For constant packet size X**
- **➔ E[X] = X**
- **➔ E[X²] = X²**
- **Therefore, overall waiting time for the packet:**

$$N = \frac{M \lambda w(1 - S/M)}{2(1 - S)} + \frac{S \lambda E[X]}{2R\,(1 - S)}$$

Average Number of Packets Per Station – Exponential Packet Size

- **For exponentially distributed packet sizes, X**
- **➔ E[X] = Xavg**
- **➔ $E[X^2] = 2\,(Xavg)^2 = 2\,E[X]^2$**
- **Therefore, overall waiting time for the packet:**

$$N = \frac{M\,\lambda\,w(1 - S/M)}{2(1 - S)} + \frac{S\,\lambda E[X]}{R\,(1 - S)}$$

Example 3:

- **<u>Problem</u>: For the network specified in Example 2, find the average number of packets per station for parts (A) and (C).**
- **<u>Solution</u>:**

(A) w = 11.52 msec, S = 0.476, M = 1000 – exponential packet sizes

$$N = \frac{1000X\ 0.01152/60(1-0.476/1000)}{2\,(1 - 0.476)} + \frac{0.476/60X200X8}{56000\ X(1 - 0.476)}$$

= 0.183 + 0.000432
= 0.183 packets / station

(C) w = 17.74 msec, S = 0.667, M = 1000 – exponential packet sizes

N = 0.444 + 0.00133
= 0.445 packets / station

Adaptive Polling

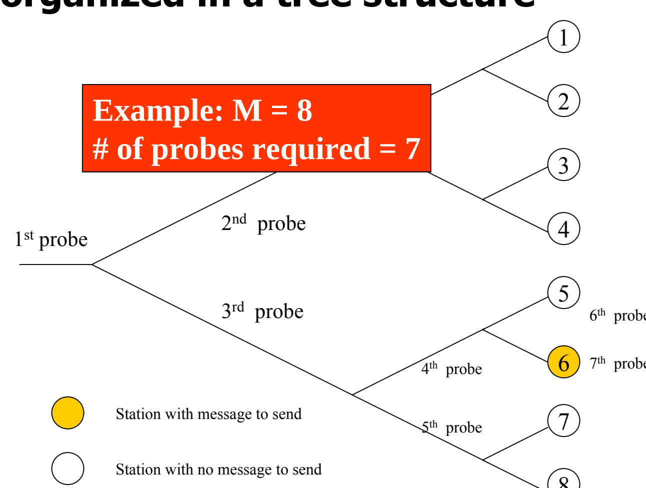
- **Using waiting time and buffer size equations: under light to moderate loading (i.e. S is small) – performance depends mainly on Number of stations, M and walking time, w**

$$W \sim Mw/2$$

- **Try to reduce number of polls ➔ Adaptive cycles**

Adaptive Polling: Pure Probing

- **Nodes are organized in a tree structure**
  - **Controller carries out probing procedure by separating stations into 2 groups that are probed at one at a time by a signal broadcast to all stations in that group**
  - **If a +ve response is received from a group, it is further divided into 2 subgroups**
  - **Process continues till station is identified**



Adaptive Polling: Pure Probing – cont'd

- Designed for low load conditions – i.e. a small fraction of terminals are transmitting
- Controller does not know that only one station wants to transmit

- If the number of stations = M
  - ➔ $2X\log_2(M) + 1$ probes are needed to locate a single ready user
  - ➔ Remember a standard polling requires $M = 2^n$ polls at most ($M/2 = 2^{n-1}$ on average to locate the single ready user)
  - Example: M = 256 stations:
    - Pure probing: 17 probes
    - Standard polling: 256 polls

Adaptive Polling: Pure Probing – cont'd

- When more than one station has data – number of probes increase

- Under heavy load (i.e. all stations have data to transmit) – number of probes becomes equal or greater than number of polls for standard polling

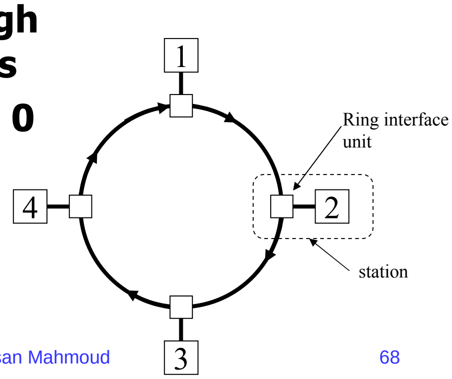
Ring Networks

- **Based on network geometry**
- **Characterized as a sequence of point-to-point links between stations, closed on itself.**
- **All messages travel over a fixed route from station to station around the loop**
- **Interface unit connects station to ring**
  - **Regenerate messages and identifies addresses**
  - **Does not store messages**

Ring Networks

- **Station latency – few bit times for all traffic passing through message (processing time)**

- **Typically – ring = high speed directional bus**
- **Propagation delay ~ 0**



Ring Networks - Advantages

- Simple implementation
- No routing is required
- Only a small latency added
- Can cover large distances (metropolitan area networks) – signal/message regeneration
- Efficiency does not degrade rapidly with load

Ring Networks - Disadvantages

- Single point failure – if a single station interface fails …
- Not so easy to expand/modify – ring must be broken
- Propagation delay is proportional to number of stations

Types of Ring Networks

- **Three Basic Types:**
  1. **Token Rings: control access to ring through passing of ring from station to station – almost same as hub polling**
  2. **Slotted Rings: a small number of fixed-sized slots are circulated; when empty they are available for use by any station**
  3. **Register Insertion Rings: two shift registers for each station node as switches to control traffic into and out of the ring – long packets can be served**

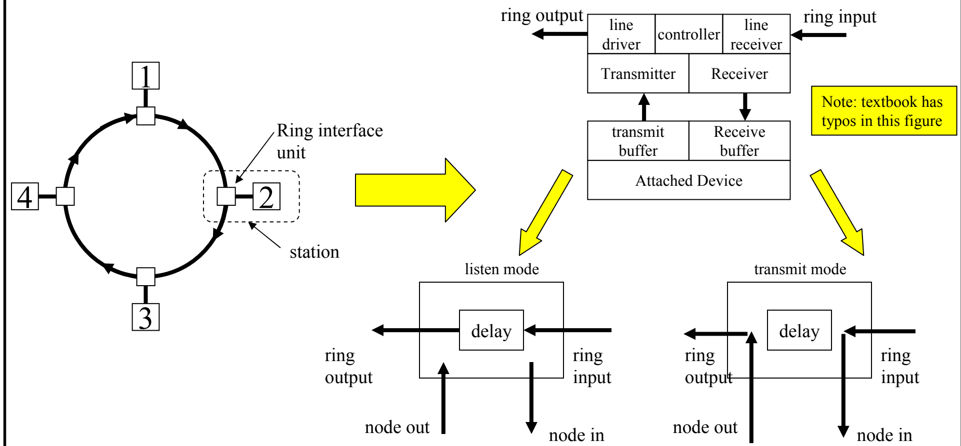
Token Ring Networks

- **Access to ring is controlled by a token**
- **Token states: busy or idle**
- **When ring is first activate – a master station circulates an idle token**
- **To transmit data, a station must:**
  - **Capture token**
  - **Set token to busy**
  - **Transmit data**
  - **Set token to idle**

See http://www.datacottage.com/nch/troperation.htm for a basic introduction and animated operation or better check http://www.macs.hw.ac.uk/~pjbk/commbook/lans/

Token Ring Networks

- **Same basic structure for all rings**



Token Pattern

- **Token:**
  - **A dedicated pattern of several bits, or**
  - **A single bit transmitted in a format different that that used for data bits**
- **Example: IEEE802 – token = several bytes long**
- ***Bit stuffing* is used to prevent occurrence of similar patterns**
- **Usually, one bit in this pattern is used to indicate whether the token is busy or free**
  - **To set the token bit – station latency = 1 bit time**
- **Can be used to add priority functionality**

Service Discipline

- **Exhaustive**
  - **Station retains use of ring until it has transmitted all the data stored in transmit buffer**
- **Non-exhaustive**
  - **Station is allowed to transmit only a specified number of bits each time it captures the token**
- **Two disciples provide same performance for light-medium loads**
- **The analysis in this package assumes exhaustive**

Idle Operation

- **Synchronization**
  - **Use of Manchester encoding**
- **All stations in listen mode**
- **Token circulates around the ring**
  - **Ring Latency = propagation time + sum of station latencies**

Normal Operation

- **One station captures token**
- **Station transmits data**
- **Station produces a modified token (or a control field in the header of the data packet) to indicate to other stations that ring is not free (i.e. token is part of packet)**
- **Transmitting station is responsible for:**
  - **removing its packet from the ring, and**
  - **generating a new token**

Normal Operation – cont'd

- **When the new token is generated – leads to three different modes of operation**
  - **Multiple token,**
  - **Single token, and**
  - **Single packet operation**

Multiple Token Operation

- **The transmitting station generates a new FREE token and places it on the ring immediately following the last bit of transmitted data**
- **This permits several busy tokens on the ring!!**
  - **What are the packet times in relation to ring latency required to achieve this?**
- **But only one free token exists!!**

Single Token Operation

- **The transmitting station generates a new FREE token and places it on the ring immediately ONLY after it removes its BUSY token**
- **Two Cases arise:**
  - **Packet time > ring latency: station will receive (and erase) its busy token before it has finished transmitting its packet – new FREE token generated after packet is completed – looks the same as multiple token operation**
  - **Packet time < ring latency: station will finish transmission of packet – must wait till it receives (and erase) busy token – new FREE is then generated**
- **Only a single token exits on the ring at any time**

Single Packet Operation

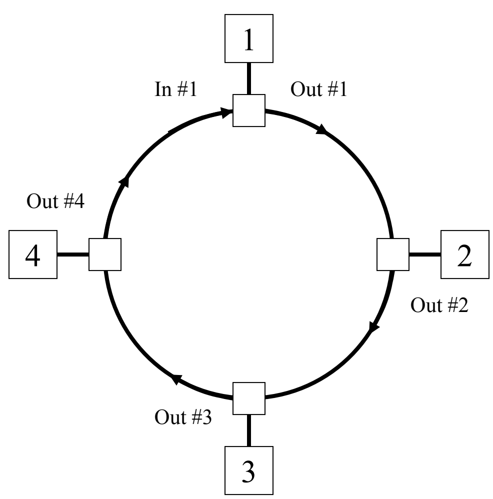
- **The transmitting station does not issue a new FREE token until after it has circulated completely around the ring and erased all of its transmitted packet**
  - **Same as single token operation except here also the packet has to be removed before the new token is generated**

- **Only a single token exits on the ring at any time**
- **Very conservative behavior – no two simultaneous transmissions on the ring**

Example 4: Four-Station Token Ring

- **<u>Example</u>:**

- **Only stations 1 and 4 have traffic to send**
- **Station 1 has 6 bits to tx**
- **Station 4 has 3 bits to tx**
- **Station 1 captures the token first, and then 4**



Ring Networks – Token Ring

time	Multiple Token					Single Token					Single Packet					time
	In #1	Out #1	Out #2	Out #3	Out #4	In #1	Out #1	Out #2	Out #3	Out #4	In #1	Out #1	Out #2	Out #3	Out #4	
0																0
	□				□	□				□	□				□	
2		■					■					■				2
		d	■				d	■				d	■			
4		d	d	■			d	d	■			d	d	■		4
	■	d	d	d	■	■	d	d	d	■	■	d	d	d	■	
6	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	6
	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	
8	d	□	d	d	d	d	□	d	d	d	d		d	d	d	8
	d		□	d	d	d		□	d	d	d			d	d	
10	d			□	d	d			□	d	d				d	10
	■				■	■				■		□				
12	d	■			d	d	■			d			□			12
	d	d	■		d	d	d	■		d				□		
14	□	d	d	■	□		d	d	■		■				■	14
		□	d	d		□		d	d	□	d	■			d	
16			□	d			□		d		d	d	■		d	16
				□				□				d	d	■		
18	□				□				□				d	d		18
						□				□				d		
20											□				□	20
												□				
22													□			22
														□		
24											□				□	24
26																26

□ free token

■ busy token

d data bit

Token Ring - Delay Analysis

- **Assumptions**
  - **All stations are identical load-wise**
  - **Arrival process ~ Poisson with λ packets / second / station**
  - **There are M stations**
  - **The average distance between stations ≈ one-half the distance around the ring**
  - **Propagation delay between consecutive stations = τ/M – where τ is the total ring propagation time**
  - **Packet size: random (uniform or exp) – average packet size = Xavg bits / packet**
  - **Exhaustive service time**

Token Ring - Delay Analysis

- **Assumptions – cont'd**
  - **Channel bit rate, R bits / second**
  - **Latency per station B bits**
  - **Round trip propagation = τ seconds**
  - **Ring Latency = τ'**

- **Required: Determine the transfer delay for token passing rings (multiple token, single token, and single packet)**

Token Rings vs Hub polling

- **Difference:**
  - **Token ring has no central station/controller**

- **Similarities:**
  - **Walk time in hub polling equivalent to time from packet transmission finish till instant when next station receives free token**

- **Therefore we will adapt the hub polling performance equations to our case here**

Review: Hub polling Performance

- It was shown previously, the packet waiting time for a polling network is given by

$$W = \frac{Mw(1 - S/M)}{2(1 - S)} + \frac{S\,E[X^2]}{2E[X]\,R\,(1 - S)}$$

Where S – is the network throughput

- The average Transfer delay (i.e. Waiting plus service time) is given by

$$T = Xavg/R + \tau avg + W$$

Where τavg is the average propagation delay from station to the central computer in the polling network

Token Ring Performance

- **Ring Latency:**
  τ' = total propagation time + sum station latencies (refer to slide 76)
  $\tau' = \tau + M\ B/R$
- One average a transmission will face $\tau'/2$ of latency before being received
- Therefore, for token ring, transfer delay T is given by

$$T = Xavg/R + \tau'/2 + W$$

Token Ring Performance – cont'd 2

- **To compute W for token ring, we need to find:**
  - **The equivalent walk time**
  - **The network throughput**
  - **The moments for service time: $E[X]/R$, and $E[X^2]/R^2$**

- **Walk time,**

  w = propagation delay from station to the next + station latency

  $$= \tau/M + B/R$$

  $$= \tau'/M$$

Token Ring Performance – cont'd 3

- **Define "effective throughput", S' to be**

  $$S' = M\lambda\, E[EST]$$

(remember throughput for the polling network is defined as $S = M\lambda\, Xavg/R$)

where E[EST] is the average effective service time for a terminal on the ring

E[EST] = total time consumed by the ring to process one packet and become free to process the next packet

Token Ring Performance – cont'd 4

- **Therefore, total transfer delay, T is given by**

$$T = Xavg/R + \tau'/2 + W$$

and W is given by

OUR MAIN RESULTS for RING NETWORKS

$$W = \frac{\tau'(1 - S'/M)}{2(1 - S')} + \frac{S'\,E[EST^2]}{2E[EST]\,(1 - S')}$$

Token Ring Performance – Multiple Token Operation

- **For multiple token operation, a free token is generated immediately after the last data bit is transmitted**

➔ $E[EST_{multiple\_token}] = ESTavg = Xavg/R$
➔ $E[EST_{multiple\_token}^2] = E[X^2]/R^2$

- **Therefore, the total transfer delay, T is given by**

$$T_{multiple\_token} = Xavg/R + \tau'/2 + W_{multiple\_token}$$

where

$$W_{multiple\_token} = \frac{\tau'(1 - S/M)}{2(1 - S)} + \frac{S\,E[X^2]}{2E[X]\,R\,(1 - S)}$$

We all know by now how to evaluate $E[X^2]$ and $E[X]$ for constant/uniform/exponentially distributed packet sizes – refer to slides 53-54.

Token Ring Performance – Single Token Operation

- For single token operation, a free token is generated when the busy token has circulated the ring completely!
- To evaluate $E[EST_{signle\_token}]$ let us define the normalized ring latency parameter a'

$$
\begin{aligned}
a' &= \tau' \ / \ (Xavg \ / \ R) \\
&= \text{propagation time} / \ (Xavg/R) + (M\ B\ /R) \ / \ (Xavg \ / \ R) \\
&= a \ + M\ B \ / \ Xavg
\end{aligned}
$$

a is the normalized ring propagation time (i.e. Tprop / Tframe)

Token Ring Performance – Single Token Operation – cont'd

- The two cases that arise:
  - a' < 1 ➔ busy token will be received before packet transmission is completed
  - a' > 1 ➔ packet transmission time finishes before start of packet circulates the ring
- This is related to the packet size X
  - X can be constant
  - X can be exponentially distributed
- Each of these cases will be considered separately

Token Ring Performance – Single Token Operation – Constant Packet Size and a' < 1

- **Packet size = X = constant ➔ Xavg = X**
- **Single token operation is the same as multiple token operation**
- **Transfer delay, T_{single_token} is the same as that for $T_{multiple\_token}$**

Token Ring Performance – Single Token Operation – Constant Packet Size and a' > 1

- **Packet size = X = constant ➔ Xavg = X**
- **Single token operation is *different* than the operation of multiple token**
- **Effective Service Time (EST) = τ' which is the time for the busy token to circulate the ring**

Token Ring Performance – Single Token Operation – Constant Packet Size and a' < 1 – cont'd

- Therefore,

$$S' = M\lambda\, E[EST]$$
$$= M\lambda\, \tau'$$
$$= M\lambda\, (Xavg/R)\, a'$$
$$= S\, a'$$

Maximum achievable throughput	= 1	if a' < 1
	1/a'	if a' > 1

- Hence, transfer delay, Tsingle_token, is given by

$$T_{single\_token} = Xavg/R + \tau'/2 + W_{single\_token}$$

where

$$W_{single\_token} = \frac{\tau'(1 - Sa'/M)}{2(1 - Sa')} + \frac{Sa'\tau'}{2\,(1 - Sa')}$$

Token Ring Performance – Single Token Operation – Exponential Packet Size

- **The packet size is random with exponential distribution**
  - **i.e. For some packets a' > 1, and for others a' < 1**
  - **Therefore, we will use the pdf of the packet size to find the pdf (or cdf) of EST and then the E[EST] and E[EST²]**

Token Ring Performance – Single Token Operation – Exponential Packet Size – cont'd

- **Packet size X is exponentially distributed ➔**

$$f_X(x) = \begin{cases} 0 & x < 0 \\ \frac{1}{\overline{X}} \exp\left(-x/\overline{X}\right) & x \geq 0 \end{cases}$$ PDF for X

Or

$$F_X(x) = \begin{cases} 0 & x < 0 \\ 1 - \exp\left(-x/\overline{X}\right) & x \geq 0 \end{cases}$$ CDF for X i.e. Prob[$X \leq x$]

Where E[X] = Xavg = $\overline{X}$

Token Ring Performance – Single Token Operation – Exponential Packet Size – cont'd 2

- **Service time = X/R ➔**

$$f_{X/R}(x) = R \times f_X(Rx) = \begin{cases} 0 & x < 0 \\ \frac{R}{\overline{X}} \exp\left(-Rx/\overline{X}\right) & x \geq 0 \end{cases}$$ PDF for X/R

Or

$$F_{X/R}(x) = \begin{cases} 0 & x < 0 \\ 1 - \exp\left(-Rx/\overline{X}\right) & x \geq 0 \end{cases}$$ CDF for X/R i.e. Prob[$X/R \leq x$]

Where E[X/R] = Xavg/R = $\overline{X}/R$

Token Ring Performance – Single Token Operation – Exponential Packet Size – cont'd 3

- **Effective Service time, EST**

Or

$$f_{EST}(x) = \begin{cases} 0 & x < \tau' \\ 1 - \exp\left(-R\tau'/\overline{X}\right) & x = \tau' \\ \frac{R}{\overline{X}}\exp\left(-Rx/\overline{X}\right) & x > \tau' \end{cases}$$

PDF for EST

$$F_{EST}(x) = \begin{cases} 0 & x < \tau' \\ 1 - \exp\left(-Rx/\overline{X}\right) & x \ge \tau' \end{cases}$$

CDF for EST
i.e. Prob[$EST \le x$]

$$E[EST] = \int_0^\infty x f_{EST}(x) = \frac{\overline{X}}{R}\exp(-a') + \tau'$$

$$E[EST^2] = \int_0^\infty x^2 f_{EST}(x) = 2\left(\frac{\overline{X}}{R}\right)^2 (1+a')\exp(-a') + (\tau')^2$$

Token Ring Performance – Single Token Operation – Exponential Packet Size – cont'd 4

- **Hence, transfer delay, T_{single_token}, is given by**

$$T_{single\_token} = Xavg/R + \tau'/2 + W_{single\_token}$$

where

Maximum achievable throughput $= 1/(e^{-a'} + a')$

$$W_{single\_token} = \frac{\tau'[1 - S(e^{-a'} + a')/M]}{2[1 - S(e^{-a'} + a')]} + \frac{Xavg}{R}\,\frac{S[(a')^2 + 2(1+a')e^{-a'}]}{2\,[1 - S(e^{-a'} + a')]}$$

Token Ring Performance – Single Packet Operation

- For single packet operation, a free token is not generated until the sending station has received and erased all of the packet it has transmitted
- Therefore, EST_{signle_packet} is always equal to $X/R + \tau'$
- Hence,

$$E[EST_{signle\_packet}] = Xavg / R + \tau'$$

$$E[EST_{signle\_packet}{}^2] = E[(X/R)^2] + 2\,\tau' E[X]/R + (\tau')^2$$

Token Ring Performance – Single Packet Operation – cont'd

- Hence, transfer delay, T_{single_packet}, is given by

$$T_{single\_packet} = Xavg/R + \tau'/2 + W_{single\_packet}$$

where

Maximum achievable throughput = 1/(1+a')

$$W_{single\_packet} = \frac{\tau'[1-(1+a')S/M]}{2[1-(1+a')S]} + \frac{Xavg}{R}\,\frac{S\,(1+a')^2}{2\,[1-(1+a')S]}$$

Token Ring Performance – Summary

Ring Parameters:	τ = total round trip propagation time (seconds) τ' (ring latency) = $\tau + MB/R$ (seconds) w (equivalent walk time) = τ'/M a' (normalized ring latency) = $\tau'/(Xavg/R)$	M = number of stations B = token size (bits) R = channel bit rate (b/s) EST – effective service time
Performance:	$T = Xavg / R + \tau'/2 + W$ $W = \frac{\tau' (1-S'/M)}{2(1-S')} + \frac{S' E[EST^2]}{2 E[EST] (1-S')}$	
Multiple Tokens	$EST = X/R \rightarrow E[EST] = Xavg/R$; $E[ESR^2] = E[X^2]/R^2$ $S' \rightarrow S$	
Single Token – Constant X	If $X/R > \tau' \rightarrow$ same as multiple tokens If $X/R < \tau' \rightarrow EST = \tau'$, $E[EST] = \tau'$ and $E[EST^2] = \tau'^2$ $S' \rightarrow Sa'$	
Single Token – Exponential X	$EST = \tau'$ if $X/R < \tau'$ X/R if $X/R > \tau'$ $\rightarrow E[EST] = (Xavg/R) e^{-a'} + \tau'$, $E[EST^2] = (\tau')^2 + 2(Xavg/R)^2 e^{-a'} (1+a')$ $S' \rightarrow S(e^{-a'} + a')$	
Single Packet	$EST = X/R + \tau' \rightarrow E[EST] = (Xavg/R) + \tau'$, $E[EST^2] = (\tau')^2 + 2\tau'(Xavg/R) + E[X^2]/R^2$ $S' \rightarrow S(1+a')$	

Example 5:

Problem: For both constant and exponential packets, evaluate the mean transfer delay for a single-token ring, that has the following parameters:

- **Ring length of 1 km**
- **Bit rate of 4 Mb/s**
- **Mean packet length of 1000 bits**
- **M = 40 stations**
- **Poisson arrival process to each station with 10 packets/second arrival rate; and**
- **Station latency of 1 bit**

Repeat this calculation for a ring in which the latency is 10 bits.

If the number of stations on the ring is increased from 40 to 120 with the same ring length, evaluate the mean transfer delay for cases of 1- and 10-bit station latency; All other network parameters are unchanged

Example 5: cont'd

$$W = \frac{\tau'(1 - S'/M)}{2(1 - S')} + \frac{S'\,E[EST^2]}{2E[EST]\,(1 - S')}$$

<u>B = 1 bit, M = 40</u>

τ = 5 μsec/km
$\tau' = \tau + MB/R$
= 5 + 40X1/4 = 15 μsec
Xavg/R = 1000/4 = 250 μsec
$a' = \tau'/(Xavg/R)$
= 15/(1000/4) = 0.06 < 1 ➔ multiple token op
➔ S' = S
S = MλXavg/R
= 40X10X1000/(4X10^6) = 0.1
E[EST] = Xavg/R
E[ESR^2] = E[X^2]/R^2
T = Xavg/R + τ'/2 + W_single_token_constant
Constant packet size:
E[EST] = Xavg/R = 1000/4 = 250 μsec
E[ESR^2] = E[X^2]/R^2 = $(1000/4)^2$ = 62.5X10^{-9} sec^2
T = 250 + 15/2 + 8.29 + 13.89
= 279.68 μsec
Exponential packet size:
E[EST] = Xavg/R = 1000/4 = 250 μsec
E[ESR^2] = E[X^2]/R^2 = 2$(1000/4)^2$ = 125X10^{-9} sec^2
T = 250 + 15/2 + 8.31 + 27.78
= 293.59 μsec

<u>B = 10 bits, M = 40</u>

τ = 5 μsec/km
$\tau' = \tau + MB/R$
= 5 + 40X10/4 = 105 μsec
Xavg/R = 1000/4 = 250 μsec
$a' = \tau'/(Xavg/R)$
= 105/(1000/4) = 0.42 < 1 ➔ multiple token op
➔ S' = S
S = MλXavg/R
= 40X10X1000/(4X10^6) = 0.1
E[EST] = Xavg/R
E[ESR^2] = E[X^2]/R^2
T = Xavg/R + τ'/2 + W_single_token_constant
Constant packet size:
E[EST] = Xavg/R = 1000/4 = 250 μsec
E[ESR^2] = E[X^2]/R^2 = $(1000/4)^2$ = 62.5X10^{-9} sec^2
T = 250 + 105/2 + 58.19 + 13.89
= 374.58 μsec
Exponential packet size:
E[EST] = Xavg/R = 1000/4 = 250 μsec
E[ESR^2] = E[X^2]/R^2 = 2$(1000/4)^2$ = 125X10^{-9} sec^2
T = 250 + 15/2 + 58.72 + 28.61
= 389.83 μsec

Example 5: cont'd

$$W = \frac{\tau'(1 - S'/M)}{2(1 - S')} + \frac{S'\,E[EST^2]}{2E[EST]\,(1 - S')}$$

<u>B = 1 bit, M = 120</u>

τ = 5 μsec/km
$\tau' = \tau + MB/R$
= 5 + 120X1/4 = 35 μsec
Xavg/R = 1000/4 = 250 μsec
$a' = \tau'/(Xavg/R)$
= 35/(1000/4) = 0.14 < 1 ➔ multiple token op
➔ S' = S
S = MλXavg/R
= 120X10X1000/(4X10^6) = 0.3
E[EST] = Xavg/R
E[ESR^2] = E[X^2]/R^2
T = Xavg/R + τ'/2 + W_single_token_constant
Constant packet size:
E[EST] = Xavg/R = 1000/4 = 250 μsec
E[ESR^2] = E[X^2]/R^2 = $(1000/4)^2$ = 62.5X10^{-9} sec^2
T = 250 + 35/2 + 24.94 + 53.57
= 346.01 μsec
Exponential packet size:
E[EST] = Xavg/R = 1000/4 = 250 μsec
E[ESR^2] = E[X^2]/R^2 = 2$(1000/4)^2$ = 125X10^{-9} sec^2
T = 250 + 35/2 + 25.04 + 107.67
= 400.21 μsec

<u>B = 10 bits, M = 120</u>

τ = 5 μsec/km
$\tau' = \tau + MB/R$
= 5 + 120X10/4 = 305 μsec
Xavg/R = 1000/4 = 250 μsec
$a' = \tau'/(Xavg/R)$
= 305/(1000/4) = 1.22 > 1 ➔ **<u>NOT</u>** multiple token op
➔ S' = Sa'
S = MλXavg/R
= 120X10X1000/(4X10^6) = 0.3
E[EST] = Xavg/R
E[ESR^2] = E[X^2]/R^2
T = Xavg/R + τ'/2 + W_single_token_constant
Constant packet size:
E[EST] = Xavg/R = 1000/4 = 250 μsec
E[ESR^2] = E[X^2]/R^2 = $(1000/4)^2$ = 62.5X10^{-9} sec^2
T = 250 + 305/2 + 239.80 + 88.04
= 730.34 μsec
Exponential packet size:
E[EST] = Xavg/R = 1000/4 = 250 μsec
E[ESR^2] = E[X^2]/R^2 = 2$(1000/4)^2$ = 125X10^{-9} sec^2
T = 250 + 305/2 + 278.54 + 192.45
= 873.49 μsec

Slotted Rings

- **Bits are transferred in serial fashion in one direction from one station to station around the ring**
- **Constant number of bit positions grouped into fixed-lengths slots – circulate continuously around the ring**
  - **i.e ring latency measure in bits ≥ total number of bit positions circulating the ring**
- **Bit spaces are grouped into mini packets**
  - **Each minipacket contains a bit in the header – bit = 1 ➔ occupied; bit = 0 ➔ free**
- **If the slot is empty, it is available for use by a station with data to transmit**

Slotted Rings – Example 6

- **Assume:**
  - **Ring speed (R) = 10 Mb/s (or bit time = 0.1 μsec)**
  - **M = 50 stations**
  - **B = 1 bit**
  - **2 km ring**
- **Propagation delay = 5 μsec / km ➔ total round trip 10 μsec**
- **Ring latency (t') = 10 + MB/R = 15 μsec**
  = 150 bit times

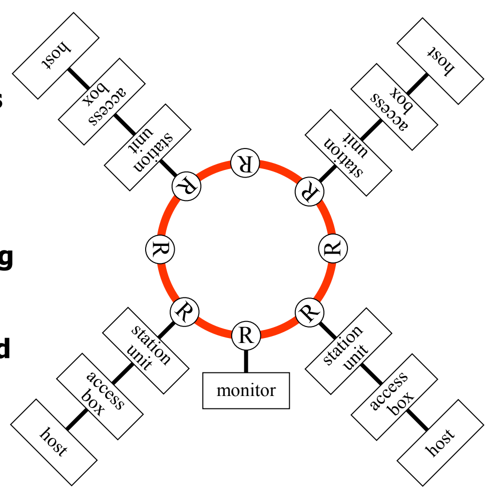
Therefore, ring can support: 3 X 50 bit slots, or
4 X 35 bit slots (with 10 bit gap), or
etc.

Slotted Rings – Characteristics

- **Designed to transmit relatively few bits at a time from each station!!**
- **Minimum access delay**

Cambridge Slotted

- **Ring sections coupled with repeaters**
- **Data rate ~ 10 Mb/s**
- **Voice grade twisted pairs cable – max section length = 100 meters**
- **Can use coaxial or fiber**
- **Monitor station – setup and maintain ring framing – ring manager**
- **Station unit – independent transmit and receive modules**
- **Access box – interface logic to host**



Cambridge Slotted – cont'd

- **Receive module**
  - **Continuously reading signal from repeater**
  - **When a minipacket is addressed to station, minipacket is saved in receive register**
  - **Minipacket maybe marked to indicate "station is busy" if station did not copy into receive register – i.e. was busy**
- **Transmit module**
  - **Shift register in station unit coupled in parallel to the access box**
  - **Data and destination bytes are written in parallel to register – source & control bits added automatically**
  - **A signal from access box sends the content of register onto the ring to fill the first empty slot**
  - **Transmit register retains a copy of the transmitted minipacket**

Cambridge Slotted – Minipacket Format

1	F	M	Destination	Source	Data	Data	R	R	P

- **Total length = 38 bits – 16 bits of data**
- **Four slots + a short gap (several digits)**
- **Frame circuit in station – synchronizes with the gap and leading 1 of each minipacket**
- **Destination – 1 byte**
- **Source – 1 byte**
- **Data – 1 byte (for each data field)**
- **M – monitor**
- **F – Full/empty bit**
- **R – Response bits (dest absent, packet accepted, dest deaf, or dest busy) – read by transmitting station before it decided to discard its copy of minipacket – no need to ANK/NAK packets**
- **P – Parity bit**

Fairness Requirement

- **The full/empty indicator must be changed to empty after the minipacket has made a complete circulation of the ring**

Slotted Ring Operation Example 7

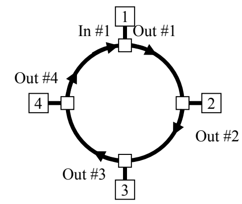
- **Two conditions:**
  - **One station has large data packet to transmit**
  - **Two stations have large data packet to transmit**
- **M = 4**
- **B = 1 – 1 bit station latency**
- **Propagation delay is ignored**
- **One slot on the ring**
- **1st bit of the 4 bit slot is used the full/empty indicator**

Slotted Ring Operation Example 7

time	Only #1 is transmitting					#1 and #3 transmitting				
	In #1	Out #1	Out #2	Out #3	Out #4	In #1	Out #1	Out #2	Out #3	Out #4
0	□				□	□				□
		■					■			
2		d1	■				d1	■		
		d1	d1	■			d1	d1	■	
4	■	d1	d1	d1	■	■	d1	d1	d1	■
	d1	□	d1	d1	d1	d1	□	d1	d1	d1
6	d1		□	d1	d1	d1		□	d1	d1
	d1			□	d1	d1			■	d1
8	□				□	■			d3	■
		■				d3	■		d3	d3
10		d1	■			d3	d3	■	d3	d3
		d1	d1	■		d3	d3	d3	□	d3
12	■	d1	d1	d1	■	□	D3	D3		□
	d1	□	d1	d1	d1		■	d3		
14	d1		□	d1	d1		d1	■		
	d1			□	d1		d1	d1	■	
16	□				□	■	d1	d1	d1	■
		■				d1	□	d1	d1	d1
18		d1	■			d1		□	d1	d1
		d1	d1	■		d1			■	d1
20	■	d1	d1	d1	■	■			d3	■
	d1	□	d1	d1	d1	d3	■		d3	d3

Empty slot

Full slot



□ free token

■ busy token

d1 data bit from #1

d3 data bit from #3

Performance of Slotted Ring

- **Assumptions**
  - **All stations are identical load-wise**
  - **Arrival process ~ Poisson with λ packets / second / station**
  - **There are M stations**
  - **Channel bit rate, R bits / second**
  - **Station latency = B bits**
  - **τ is the total ring propagation time**
  - **τ' is the total ring latency = τ + MB/R**
  - **Packet size: exponential – average packet size = Xavg bits / packet**
  - **The minipacket length is much less than the packet size**

Distributed M/M/1 Queue

- **Since the time for a slot to circulate the ring (opportunity to transmit) is very small compared to the packet transmission time ➔ modeled as a distributed M/M/1 queue**
  - **Arrival rate = $M\lambda$**
  - **Service time = Xavg/R**
  - **From station perspective: effective channel rate = R/2 – caused by the strategy to prevent ring hogging**
  - **R is used to compute overall throughput**

Distributed M/M/1 Queue – cont'd

- **Network throughput, S is given by**

$$S = M\lambda \, X_{avg}/R$$

- **For slotted ring:**
  - **ρ ➔ S**
  - **R ➔ R/2**
  - **T ➔ T + τ'/2**

Remember:
Average transfer delay for M/M/1 (total delay) is given by

$$T = \frac{E[\text{service time}]}{1-\rho} = \frac{X_{avg}/R}{1-\rho}$$

where ρ is the utilization of the queueing system

- **Therefore, for slotted ring, transfer delay, T is given by**

$$T = \frac{2}{1-S}\,\frac{X_{avg}}{R} + \frac{\tau'}{2}$$

- **Result valid for arbitrary packet length distribution**

Refined Results

- **The previous model does not account for the huge overhead in each minipacket!!**
- **Let the minipacket or slot size be Lh (overhead bits) + Ld (data bits)**
  - **Define h = Lh / Ld**
  - **Using the above definitions, one can write**

$$\tau' = m(Lh + Ld)/R + g = \tau + MB/R$$

where m is the number of slots on the ring and g is the gap in seconds

- **Therefore:**

$$X \rightarrow (1 + h)\, X$$
$$S \rightarrow (1 + h)\, S$$

Refined Results – cont'd

- **Substituting in the previous result, yields**

$$T = \frac{2(1+h)}{1 - S(1+h)} \; \frac{Xavg}{R} + \frac{\tau'}{2}$$

- **Now – maximum throughput = 1/(1+h) or Ld/(Lh + Ld) – which is the correct result**

Example 8:

- **Problem: A slotted ring is 1 kilometer long, has 50 stations attached and has a bit rate of 10 Mb/s. Each slot contains 3 bytes of data, a source byte, a destination byte, and another byte that includes the monitor and indicator bits. It may be assumed that each station latency is 1 bit**

A) **How many slots this ring hold without adding any artificial delays? What is the gap time? If packets of length 1200 bits are to be transmitted on this ring, find the mean transfer delay when packets arrive at each station at a rate of (i) 1 packet / second (ii) 40 packets / second**

B) **Increase the number of station on the network to 100. (i) How many slots can the ring now hold without adding artificial delays? (ii) What is the gap time? Again, evaluate the mean transfer delay for the same arrival rates and same packet length.**

Example 8: solution

A) For M = 50 stations
Propagation delay, $\tau = 5\ \mu sec$
Ring latency, $\tau' = \tau + MB/R$
$= 5 + 50X1/10 = 10\ \mu sec$
Slot length, = 6 bytes or 48 bits
Since $\tau' = m(48)/10 + g = 10$
Therefore, $m \leq 2$ – if m = 2, then g = 0.4 μsec
$h = Lh/Ld = 24/24 = 1$
$Xavg / R = 1200 / 10 = 120\ \mu sec$
(i) $S = M\lambda Xavg/R = 50X1X120X10^{-6} = 0.006$

$$T = \frac{2(1+h)}{1 - S(1+h)} \frac{Xavg}{R} + \frac{\tau'}{2}$$

$$= \frac{2 X 2}{1 - 0.006 X 2} 120 + 10/2 = 490.8\ \mu sec$$

(ii) $S = M\lambda Xavg/R = 50X40X120X10^{-6} = 0.24$
$T = 928.1\ \mu sec$

Example 8: solution – cont'd

B) For M = 100 stations
Propagation delay, τ = 5 μsec
Ring latency, τ' = τ + MB/R
= 5 + 100X1/10 = 15 μsec
Slot length, = 6 bytes or 48 bits
Since τ' = m(48)/10 + g = 10
Therefore, m $\leq$ 3 – if m = 3, then g = 0.6 μsec
h = Lh/Ld = 24/24 = 1
Xavg / R = 1200 / 10 = 120 μsec
(i) S = MλXavg/R = 100X1X120X10^{-6} = 0.012

$$T = \frac{2(1+h)}{1 - S(1+h)} \cdot \frac{Xavg}{R} + \frac{\tau'}{2}$$

$$= \frac{2 \times 2}{1 - 0.012 \times 2} \; 120 + 15/2 = 499.3 \; \mu sec$$

(ii) S = MλXavg/R = 100X40X120X10^{-6} = 0.48
T = 12007.5 μsec