# Lectures 15 & 16

# Local Area Networks

**Eytan Modiano**

# Carrier Sense Multiple Access (CSMA)

- **In certain situations nodes can hear each other by listening to the channel - "Carrier Sensing"**

- **CSMA: Polite version of Aloha**
  - **Nodes listen to the channel before they start transmission**

    **Channel idle => Transmit**

    **Channel busy => Wait (join backlog)**

  - **When do backlogged nodes transmit?**

    **When channel becomes idle backlogged nodes attempt transmission with probability $q_r = 1$**

    **Persistent protocol, $q_r = 1$**

    **Non-persistent protocol, $q_r < 1$**

# CSMA

- **Let $\tau$ = the maximum propagation delay on the channel**
  - **When a node starts/stops transmitting, it will take this long for all nodes to detect channel busy/idle**

- **For initial understanding, view the system as slotted with "mini-slots" of duration equal to the maximum propagation delay**
  - **Normalize the mini-slot duration to $\beta = \tau/D_{tp}$ and packet duration = 1**


- **Actual systems are not slotted, but this hypothetical system simplifies the analysis and understanding of CSMA**

# Rules for slotted CSMA

- When a new packet arrives
  - If current mini-slot is idle, start transmitting in the next mini-slot
  - If current mini-slot is busy, node joins backlog
  - If a collision occurs, nodes involved in collision become backlogged
- Backlogged nodes attempt transmission after an idle mini-slot with probability $q_r < 1$ (non-persistent)
  - Transmission attempts only follow an idle mini-slot
  - Each"busy-period" (success or collision) is followed by an idle slot before a new transmission can begin
- Time can be divided into epochs:
  - A successful packet followed by an idle mini-slot (duration = $\beta$+1)
  - A collision followed by an idle mini-slot (duration = $\beta$+1)
  - An idle minislot (duration = $\beta$)

# Analysis of CSMA

- Let the state of the system be the number of backlogged nodes
- Let the state transition times be the end of idle slots
  - Let T(n) = average amount of time between state transitions when the system is in state n

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

When qr is small $(1-q_r)^n \sim e^{-q_r n} \Rightarrow T(n) = \beta + (1 - e^{-\lambda\beta - nq_r})$

- At the beginning of each epoch, each backlogged node transmits with probability $q_r$
- New arrivals during the previous idle slot are also transmitted
- With backlog n, the number of packets that attempt transmission at the beginning of an epoch is approximately Poisson with rate

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

# Analysis of CSMA

- The probability of success (per epoch) is

  $P_s = g(n)\, e^{-g(n)}$

- The expected duration of an epoch is approximately

  $T(n) \sim \beta + (1 - e^{-g(n)})$

- Thus the success rate per unit time is

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

# Maximum Throughput for CSMA

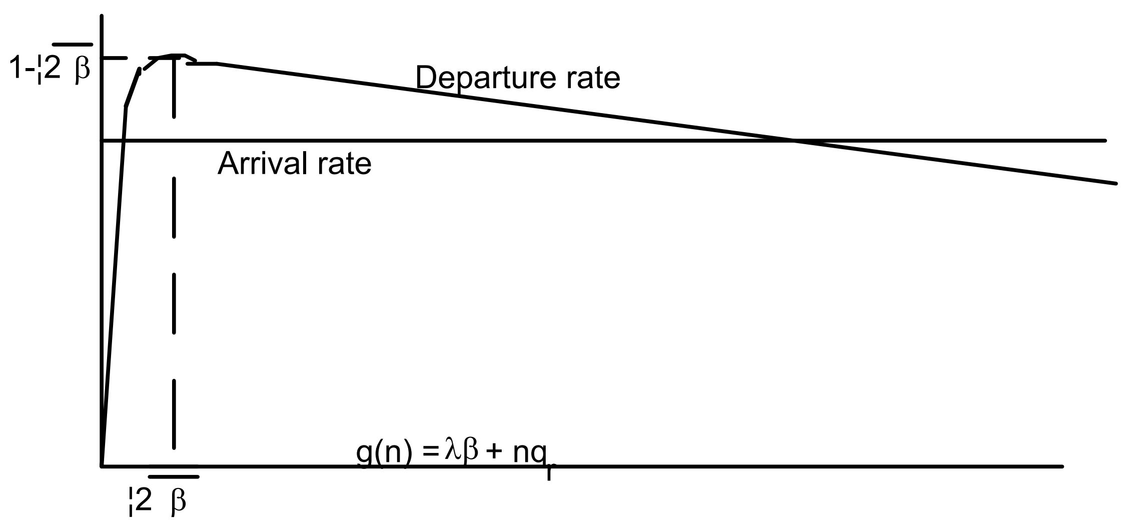
- **The optimal value of g(n) can again be obtained:**

$$g(n) \approx \sqrt{2\beta} \qquad \lambda < \frac{1}{1+\sqrt{2\beta}}$$

- **Tradeoff between idle slots and time wasted on collisions**

- **High throughput when β is small**

- **Stability issues similar to Aloha (less critical)**

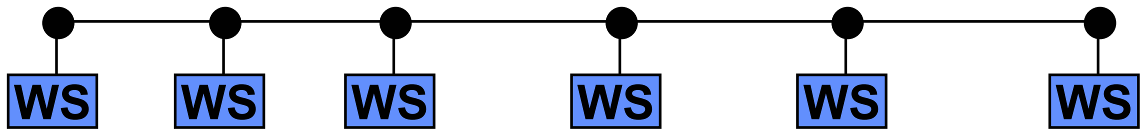


# Unslotted CSMA

- **Slotted CSMA is not practical**
  - Difficult to maintain synchronization
  - Mini-slots are useful for understanding but not critical to the performance of CSMA
- **Unslotted CSMA will have slightly lower throughput due to increased probability of collision**
- **Unslotted CSMA has a smaller effective value of $\beta$ than slotted CSMA**
  - Essentially $\beta$ becomes average instead of maximum propagation delay

# CSMA/CD and Ethernet

Two way cable




- CSMA with Collision Detection (CD) capability
  - Nodes able to detect collisions
  - Upon detection of a collision nodes stop transmission
    - Reduce the amount of time wasted on collisions

- Protocol:
  - All nodes listen to transmissions on the channel
  - When a node has a packet to send:
    - Channel idle => Transmit
    - Channel busy => wait a random delay (binary exponential backoff)
  - If a transmitting node detects a collision it stops transmission
    - Waits a random delay and tries again

# Time to detect collisions

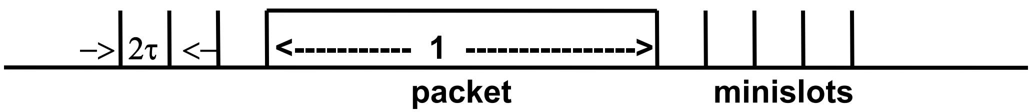

- A collision can occur while the signal propagates between the two nodes
- It would take an additional propagation delay for both users to detect the collision and stop transmitting

- If $\tau$ is the maximum propagation delay on the cable then if a collision occurs, it can take up to $2\tau$ seconds for all nodes involved in the collision to detect and stop transmission

# Approximate model for CSMA/CD

- **Simplified approximation for added insight**

- **Consider a slotted system with "mini-slots" of duration $2\tau$**




- **If a node starts transmission at the beginning of a mini-slot, by the end of the mini-slot either**
  - **No collision occurred and the rest of the transmission will be uninterrupted**
  - **A collision occurred, but by the end of the mini-slot the channel would be idle again**

- **Hence a collision at most affects one mini-slot**

# Analysis of CSMA/CD

- Assume N users and that each attempts transmission during a free "mini-slot" with probability p
  - P includes new arrivals and retransmissions

$$P(\text{i users attempt}) = \binom{N}{i} P^i (1-P)^{N-i}$$

$$P(\text{exactly 1 attempt}) = P(\text{success}) = NP(1-P)^{N-1}$$

To maximize P(success),

$$\frac{d}{dp}[NP(1-P)^{N-1}] = N(1-P)^{N-1} - N(N-1)P(1-P)^{N-2} = 0$$

$$\Rightarrow P_{opt} = \frac{1}{N}$$

$\Rightarrow$ Average attempt rate of one per slot

$\Rightarrow$ Notice the similarity to slotted Aloha

# Analysis of CSMA/CD, continued

$$P(\text{success}) = NP(1-p)^{N-1} = (1-\frac{1}{N})^{N-1}$$

$$P_s = \text{limit } (N \rightarrow \infty)\ P(\text{success}) = \frac{1}{e}$$

Let X = Average number of slots per succesful transmission

$$P(X=i) = (1-P_s)^{i-1} P_s$$

$$\Rightarrow E[X] = \frac{1}{P_s} = e$$

- Once a mini-slot has been successfully captured, transmission continues without interruption
- New transmission attempts will begin at the next mini-slot after the end of the current packet transmission

# Analysis of CSMA/CD, continued

- **Let S = Average amount of time between successful packet transmissions**

$$S = (e-1)2\tau + D_{Tp} + \tau$$

Idle/collision Mini-slots ($(e-1)2\tau$)

Packet transmission time ($D_{Tp}$)

Ave time until start of next Mini-slot ($\tau$)

- **Efficiency = $D_{Tp}/S = D_{Tp} / (D_{Tp} + \tau + 2\tau(e-1))$**

- **Let $\beta = \tau / D_{Tp}$ => Efficiency $\approx 1/(1+4.4\beta) = \lambda < 1/(1+4.4\beta)$**

- **Compare to CSMA without CD where** $\lambda < \dfrac{1}{1+\sqrt{2\beta}}$

# Notes on CSMA/CD

- Can be viewed as a reservation system where the mini-slots are used for making reservations for data slots
- In this case, Aloha is used for making reservations during the mini-slots
- Once a users captures a mini-slot it continues to transmit without interruptions
- In practice, of course, there are no mini-slots
  - Minimal impact on performance but analysis is more complex

# CSMA/CD examples

- **Example (Ethernet)**
  - Transmission rate = 10 Mbps
  - Packet length = 1000 bits, $D_{Tp} = 10^{-4}$ sec
  - Cable distance = 1 mile, $\tau = 5 \times 10^{-6}$ sec
  - ➨ $\beta$ = 5x10-2 and E = 80%

- **Example (GEO Satellite) - propagation delay 1/4 second**
  - $\beta$ = 2,500 and E ~ 0%

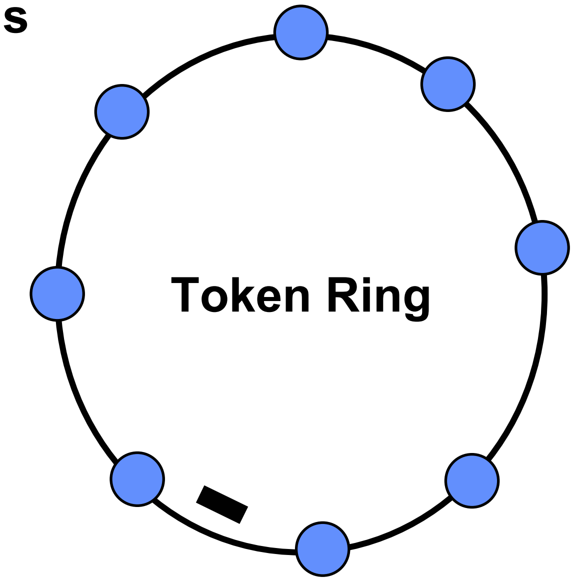
- **CSMA/CD only suitable for short propagation scenarios!**

- **How is Ethernet extended to 100 Mbps?**

- **How is Ethernet extended to 1 Gbps?**

# Token rings

- **Token rings were developed by IBM in early 1980's**
- **Token: a bit sequence**
  - **Token circulates around the ring**
    - **Busy token: 01111111**
    - **Free token: 01111110**
- **When a node wants to transmit**
  - **Wait for free token**
  - **Remove token from ring (replace with busy token)**
  - **Transmit message**
  - **When done transmitting, replace free token on ring**
  - **Nodes must buffer 1 bit of data so that a free token can be changed to a busy token**
- **Token ring is basically a polling system**
    - **Token does the polling**



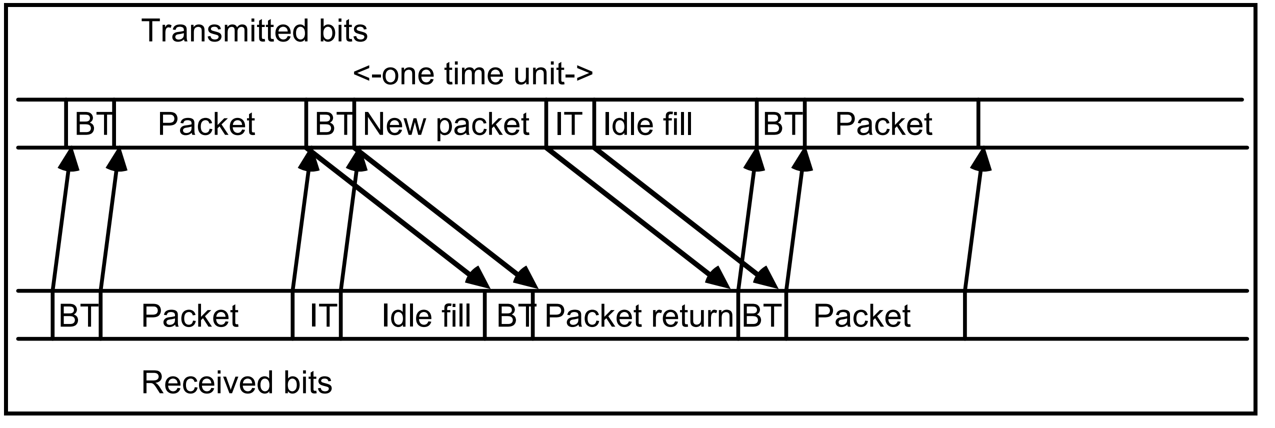
# Release of token

- Release after transmission
  - Node replaces token on ring as soon as it is done transmitting the packet
  - Next node can use token after  short propagation delay

- Release after reception
  - Node releases token only after its own packet has returned to it
    - Serves as a simple acknowledgement mechanism

# PACKET TRANSMISSION (release after transmission)

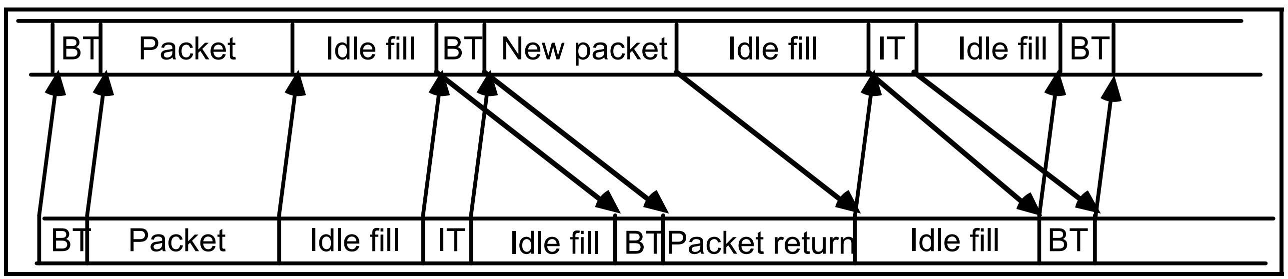
- **When not transmitting their own packets nodes relay whatever they receive**
- **After receiving an idle token a node can start sending a new packet (discard incoming bits)**
- **After a node sends a packet and the idle token, it sends idle fill until:**
  - **The packet followed by idle, or**
  - **busy token, returns around the ring**



# PACKET TRANSMISSION
# (release after reception)

- **In many implementations (including IEEE802.5, but not including FDDI), a node waits to check its packet return before sending the idle token.**

  **This increases packet transmission time by one round trip delay.**



# Delay analysis

- System can be analyzed using multi-user reservation results
- Exhaustive system - nodes empty their queue before passing token on to the next node
- Assume m nodes and each with Poisson arrivals of rate λ/m
- Let v = average propagation and token transmission delay
- System can be viewed as a reservation system with m users and average reservation interval (see reservation system results)

$$W = \frac{\lambda E[X^2] + v(m-\rho)}{2(1-\rho)}, \quad \rho = m(\lambda/m)E[X] = \lambda E[X]$$

- Notice that 100% throughput can be achieved for exhaustive system

# Throughput analysis (non-exhaustive)

- Gated system with limited service - each node is limited to sending one packet at a time
  - When system is heavily loaded nodes are always busy and have a packet to send

- Suppose each node transmits one packet and then releases the token to the next node
  - $V_i$ = propagation and transmission time for token between two nodes (transmission time is usually negligible)
- The amount of time to transmit N packets

$$T_N = N*E[X] + V_1 + V_2 + \ldots + V_N = N*E[X] + N*E[V]$$

$$\lambda < N*E[X]/(N*E[X] + N*E[V]) = 1/(1+E[V]/E[X])$$

- Compare to CSMA/CD, but notice that V is the delay between two nodes and not the maximum delay on the fiber

# Throughput analysis
# (token release after reception)

- **Nodes release token only after it has returned to it**
- **Again assume each node sends one packet at a time**

- **Total time to send ONE packet**

- $T = E[X] + V_1 + V_2 + \ldots + V_m + V_i$

  Time to send token to next node ($V_i$)

  M nodes on the ring ($V_m$)

- $T = E[X] + (m+1)E[V] \Rightarrow$

$$\lambda < E[X]/T = 1/(1+(m+1)E[V]/E[X])$$

# Delay Analysis

- **Release after transmission**
  - **Partially gated limited service system (sec. 3.5.2)**

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

- **Release after reception**
  - **Homework problem 4.27**
  - **Additional round-trip time can be added to the packet transmission time**

$$W = \frac{\lambda(E[X^2] + 2mv + m^2v^2) + v(m + \lambda(E[X] + mv))}{2(1 - \lambda(E[X] + (m+1)v))}$$

# Token ring issues

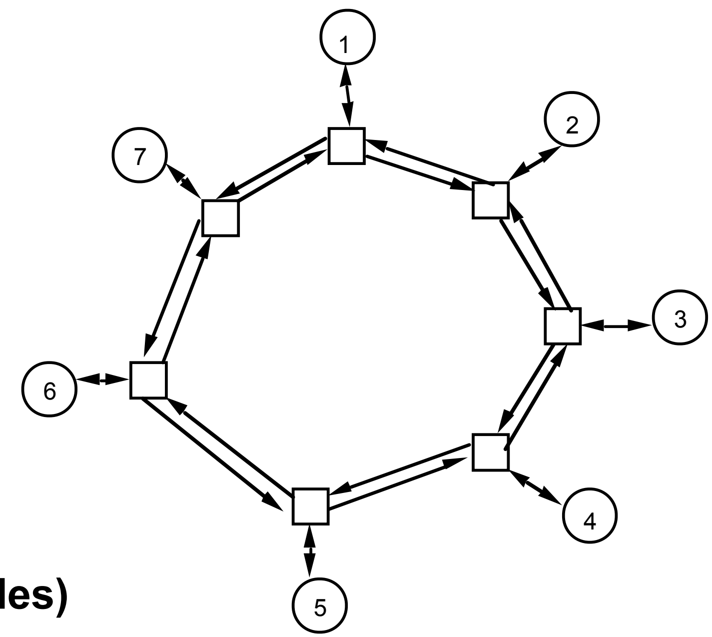
- **Fairness: Can a node hold the token for a long time**
  - **Solution: maximum token hold time**
- **Token failures: Tokens can be created or destroyed by noise**
  - **Distributed solution:**

    **Nodes are allowed to recognize the loss of a token and create a new token**

    **Collision occurs when two or more nodes create a new token at the same time => need collision resolution algorithms**
- **Node failures: Since each node must relay all incoming data, the failure of a single node will disrupt the operation of the ring**
- **Token ring standard: IEEE 802.5**

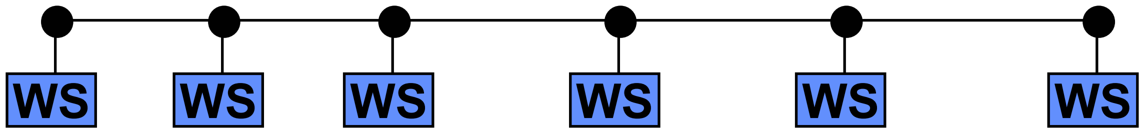
# FDDI

- **Fiber distributed data interface (FDDI) is a 100 Mbps Fiber Optic Token Ring local area network standard**
- **FDDI uses two counter-rotating rings**
  - **Single faults can be isolated by switching from one ring to the other on each side of fault.**
- **Token release after transmission**
- **Limit on token hold time**
- **Upper-bound on time between token visits at a node**
  - **Support for guaranteed delays**
  - **Imposes a limit on the size of a ring (distance between nodes, number of nodes)**




- **FDDI designed to be a metro or campus area network technology**

# TOKEN BUSES




- **Special control packet serves as a token**
- **Nodes must have token to transmit**
- **Token is passed from node to node in some order**
  – **Conceptually, a token bus is the same as a token ring**
  – **When one node finishes transmission, it sends an idle token to the next node (by addressing the control packet properly)**
  – **Similar to a polling system**
- **Issues**
  – **Efficiency lower than token rings due to longer transmission delay for the packets and longer propagation delays**
  – **Need protocol for joining and leaving the bus**

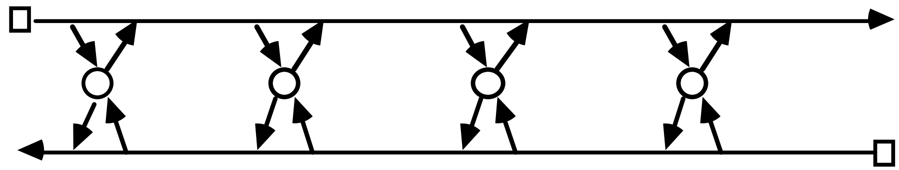
# IMPLICIT TOKENS

- The idle tokens on a token bus can be replaced with silence
- The next node starts to transmit a packet after hearing the bus become silent
- If the next node has no packet, successive nodes start with successively greater delay
- If the bus propagation delay is much smaller than the time to transmit a token, this can reduce delay

- This scheme is used for wireless LANs (IEEE 802.11) and it goes by the name of CSMA/CA (collision avoidance)

# DISTRIBUTED QUEUE DUAL BUS (DQDB)

- Metropolitan area network using two oppositely directed unidirectional 150 Mbps buses
- All frames are the same length (53 bytes); empty frames are generated at the head ends of the buses and are filled by the nodes "on the fly"

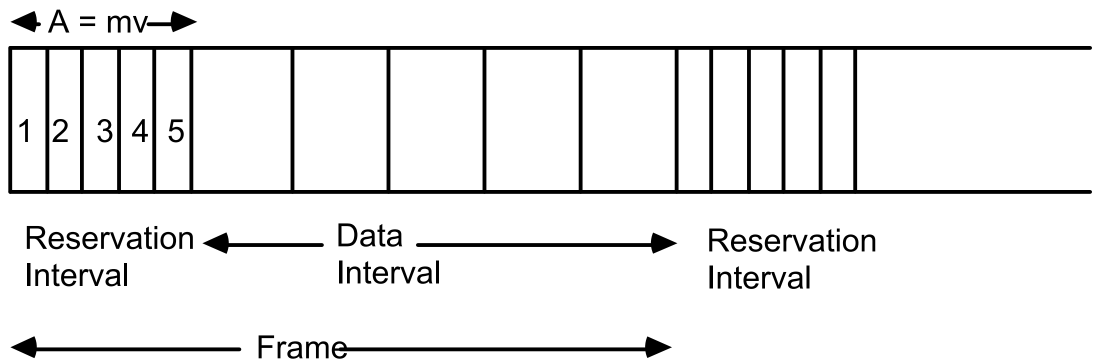


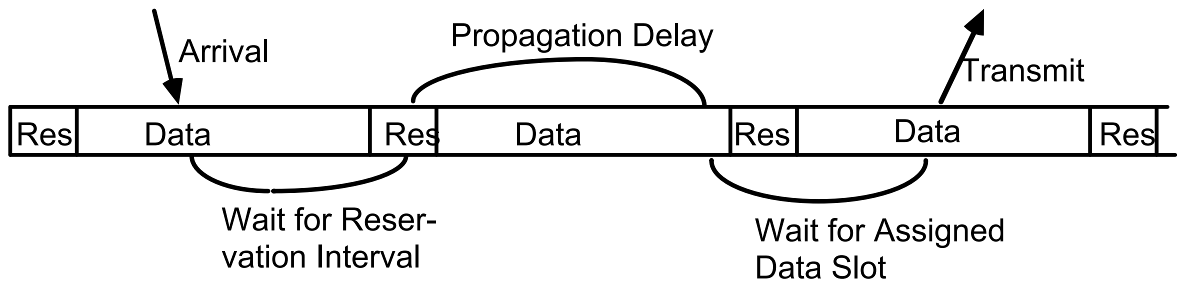
- A node uses the right moving bus to send frames to nodes on the right and the left moving bus for nodes on its left
- DQDB was standardized as IEEE 802.6 and was intended to be compatible with ATM

# DQDB Reservations

- **Greedy algorithm: Each node uses a free slot when it has something to send**
  - **Thus an efficiency of 100% is possible**
- **The trouble with this trivial approach is unfairness - nodes at the tail of the bus can be "starved"**
- **DQDB uses a reservations systems whereby nodes send requests upstream so that empty slots can be reserved**
  - **If a node has a frame to send on the right bus, it sets the request bit in a frame on the left bus**
  - **Nodes maintain an "implicit" queue of requests that can be served on a FCFS basis (hence the name distributed queue)**

# Large propagation delay (satellite networks)




- **Satellite reservation system**
  - **Use mini-slots to make reservation for longer data slots**
  - **Mini-slot access can be inefficient (Aloha, TDMA, etc.)**
- **To a crude approximation, delay is 3/2 times the propagation delay plus ideal queueing delay.**

# Satellite Reservations

- Frame length must exceed round-trip delay
  - Reservation slots during frame j are used to reserve data slots in frame j+1
  - Variable length: serve all requests from frame j in frame j+1
    - Difficult to maintain synchronization
    - Difficult to provide QoS (e.g., support voice traffic)
  - Fixed length: Maintain a virtual queue of requests
- Reservation mechanism
  - Scheduler on board satellite
  - Scheduler on ground
  - Distributed queue algorithm
    - All nodes keep track of reservation requests and use the same algorithm to make reservation
- Control channel access
  - TDMA: Simple but difficult to add more users
  - Aloha: Can support large number of users but collision resolution can be difficult and add enormous delay

# Aloha Reservations

- Use Aloha to capture a slot
- After capturing a slot user keeps the slot until done
  - Other users observe the slot busy and don't attempt
- When done other users can go after the slot
  - Other users observe the slot idle and attempt using Aloha
- Method useful for long data transfers or for mixed voice and data

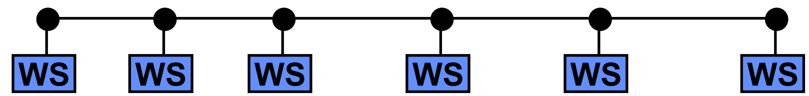
| Slot | 1 | 2 | 3 | 4 | 5 | 6 | |
|---|---|---|---|---|---|---|---|
| | 15 | idle | 3 | 20 | | 2 | frame 1 |
| | 15 | 7 | 3 | idle | 9 | 2 | frame 2 |
| | idle | 7 | 3 | | 9 | idle | frame 3 |
| | 18 | 7 | 3 | | 9 | 6 | frame 4 |
| | 18 | 7 | 3 | 15 | 9 | 6 | frame 5 |

# Packet multiple access summary

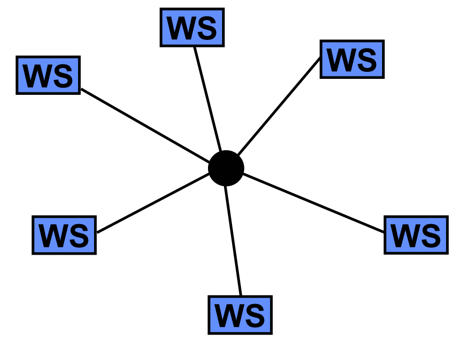
- Latency: Ratio of propagation delay to packet transmission time
  - GEO example: Dp = 0.5 sec, packet length = 1000 bits, R = 1Mbps
    - Latency = 500 => very high
  - LEO example: Dp = 0.1 sec
    - Latency = 100 => still very high
  - Over satellite channels data rate must be very low to be in a low latency environment
- Low latency protocols
  - CSMA, Polling, Token Rings, etc.
  - Throughput ~ $1/(1+a\alpha)$, $\alpha$ = latency, a = constant
- High latency protocols
  - Aloha is insensitive to latency, but generally low throughput
    - Very little delays
  - Reservation system can achieve high throughput
    - Delays for making reservations
  - Protocols can be designed to be a hybrid of Aloha and reservations
    - Aloha at low loads, reservations at high loads

# Migration to switched LANs

- **Traditional Ethernet**
  - Nodes connected with coax
    - Long “runs” of wire everywhere
  - CSMA/CD protocol




- **“Hub” Ethernet**
  - Nodes connected to hub
    - Hub acts as a broadcast repeater
    - Shorted cable “runs”, Useful for 100 Mbps
  - CSMA/CD protocol
  - Easy to add/remove users
  - Easy to localize faults
  - Cheap cabling (twisted pair, 10baseT)




- **Switched Ethernet**
  - No CSMA/CD
    - Easy to increase data rate (e.g., Gbit Ethernet)
  - Nodes transmit when they want
  - Switch queues the packets and transmits to destination
  - Typical switch capacity of 20-40 ports
  - Each node can now transmit at the full rate of 10/100/Gbps
  - <u>Modularity:</u> Switches can be connected to each other using high rate ports