# Computer Networks

Medium Access Sublayer

## Topics

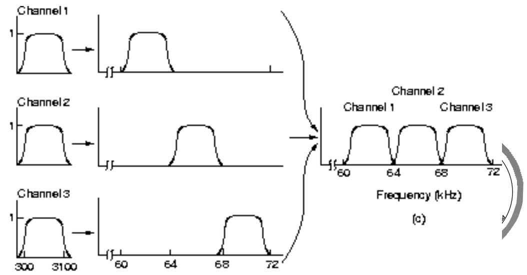
- ✦ Introduction
- ✦ Multiple Access Protocols
- ✦ IEEE 802 Standard
- ✦ Bridges
- ✦ Misc (brief)
  - – High-Speed LANs
  - – Satellite Networks

## Introduction

- ✦ Remember, two categories of networks
  - – point-to-point
  - – broadcast
- ✦ Key issue is who gets channel
  - – example: 6-person conference call
- ✦ Many protocols to decide
- ✦ Medium Access Control sublayer
  - – lower part of data-link layer, but easier here
- ✦ Many LANs multiaccess
  - – satellites, too

## Fixed Channel Allocation

- ✦ Static channel allocation
  - – FDM, TDM




## FDM

- ✦ Time delay T
- ✦ Capacity C bps
- ✦ Arrival rate λ frames/sec
- ✦ Frames mean 1/μ bits

$$T = \frac{1}{\mu C - \lambda}$$

- ✦ Divide into N channels
- ✦ Each channel C/N bps

$$T = \frac{1}{\mu(C/N) - (\lambda/N)} = \frac{N}{\mu C - \lambda} = NT$$

TDM is the same

## Multiple Access

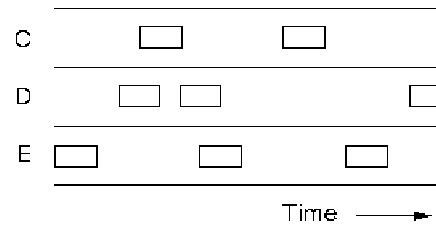
- ✦ So … multiple access can be more efficient
- ✦ Assumptions
  - – N independent stations
  - – One channel
  - – Collision detection
- ✦ Types
  - – contention systems
  - – limited contention systems
  - – collision free systems

## ALOHA - A Family of Contention Protocols

- 1970's, Abramson
- University of Hawaii
- Ground based broadcasting, packet radio
  - generalizes to uncoordinated users competing for single, shared channel
- Pure ALOHA
  - no time slots
- Slotted ALHOA
  - time slots for frames

## Pure ALOHA

- Transmit whenever you want




- Detect collisions after sending
  - checksum error
- If collision, wait random time and retry

## Pure ALOHA == Pure Chaos?

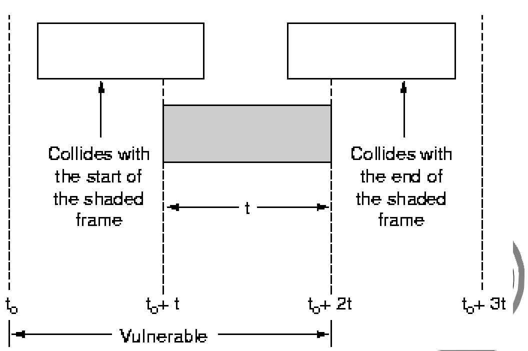
- Assume infinite collection of stations
- Users in two states: *typing* or *waiting*
- User *typing* a line. When done, transmit it.
  - user *waiting* for response. When done, *typing*.
- *frame time* is time to put frame on wire
  - frame length / bit rate
- Mean number of new frames per frame time
  - $N$
  - What does $N > 1$ mean?

## Analysis of Pure ALOHA

- Stations also re-generate collided frames
  - $G$ is old plus new frames
  - $G > N$? $G = N$? $G < N$?
- Low load ($N \approx 0$), few collisions: $G \approx N$
- High load, many collisions: $G > N$
- Throughput per frame time is $G$ times probability of frame having zero collisions:

  $S = G P_0$
  - ex: $G$=.5, $P_0$=.5 so $S$ = .25

## Frame Collisions




## Analysis of Pure ALOHA (cont.)

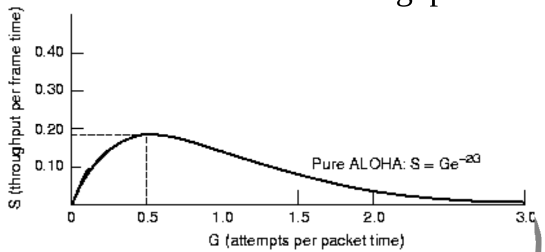
- Probability $k$ frames generated per frame time

  $$\Pr[k] = \frac{G^k e^{-G}}{k!}$$

  $$\Pr[0] = e^{-G}$$
- Need two frame times empty, 2$G$ generated
  - for two slots, $\Pr[0] = e^{-2G}$
- Throughput per frame time

  $$S = Ge^{-2G}$$

# Pure ALOHA
# Offered Load vs. Throughput



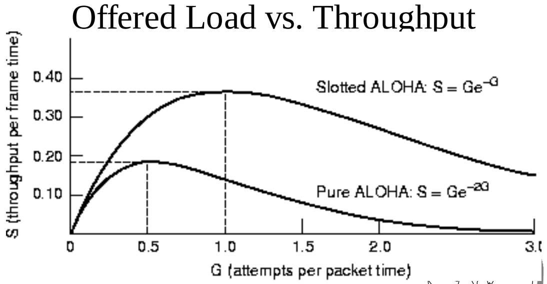

- Max at $G = 0.5$, $S = 1/2e$, only about 0.184 (18%)!
  - Can we do better?

# Slotted ALOHA

- Divide time into intervals, one for each frame
- Stations agree upon time intervals
  - one can "pip" as time keeper, like a clock
- Users transmit only at beginning of slot
- Need one frame time to be empty, $G$ generated
  - for one slot, `Pr[0] = e`$^{-G}$
- Throughput

  `S = Ge`$^{-G}$

# Slotted ALOHA
# Offered Load vs. Throughput




- Max at $G = 1$, $S = 1/e$, only about 0.368 (37%)
  - This is ***not*** Ethernet!

# Last Thoughts on Slotted ALOHA

- Best (G = 1):
  - 37% empty
  - 37% success
  - 26% collisions
- Raising G, reduces empties but increases collisions exponentially
- Expected transmissions (includes original)

  `E = e`$^{G}$
  - G=0, then 1 transmission; G=1 then 2.X trans.
- Small increase in load, big decrease in perf

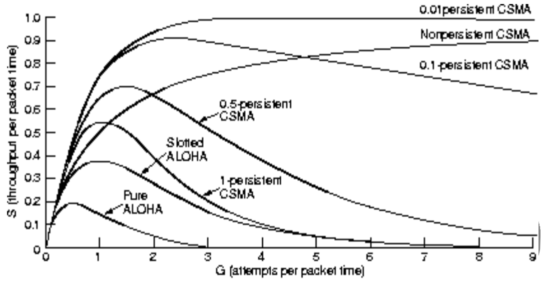
# Carrier Sense Multiple Access - CSMA Protocols

- Sending without paying attention is obviously limiting
- In LANs, can detect what others are doing
- Stations listen for a transmission
  - *carrier sense protocols*

# Persistent and Nonpersistent

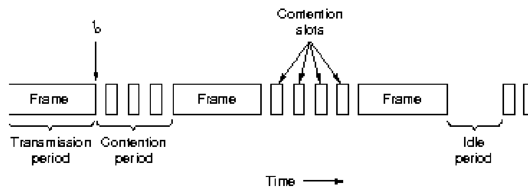
- *1-persistent* CSMA
  - detect, send at first chance
  - wait if another sending
  - longer delay, more collisions
- *non-persistent* CSMA
  - if empty, send
  - if not, less greedy, waits random time then repeats
  - fewer collisions, longer delay
- *p-persistent* CSMA
  - if empty, sends with probability $p$
  - defers with probability $q = 1 - p$

## Carrier Sense Multiple Access




## CSMA with Collision Detection

- If detect collision, stop transmitting
  - frame will be garbled anyway
- CSMA with Collision Detection (CD)




## CSMA/CD Closing Comments

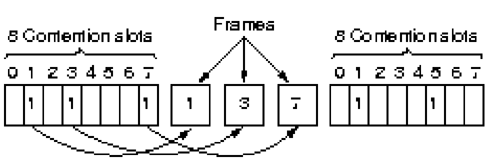
- How long until realize a collision? Time to travel length of cable? Why not?
- Propogation τ, need 2τ to "seize" the line
- Model 2τ slot as slotted ALOHA
- 1-km cable has τ ≈ 5 µsec
- Collision detection *analog*
  - special hardware encoding so can detect
- Does not guarantee reliable delivery
- Basis IEEE 802.3 (*Ethernet*)

## Collision-Free Protocols

- Collisions still occur in CSMA/CD
- More so when "wire" long (large τ)
- Short frames, too, since contention period becomes more significant
- Want *collision free protocols*
- Need to assume *N* stations have numbers, 0 to (N-1) wired in

## Bit-Map Protocol

- Have N contention slots
- Station N puts 1 in slot N-1, else 0
  - ex: station 0 wants to send, 1 in 0th slot




## Bit-Map Protocol Performance

- *N* contention slots, so *N* bits overhead /frame
- *d* data bits
- Station wants to transmit, waits avg *N*/2 slots
- Efficiency under low load (1 sending):
  - d /(N+d)
  - average delay: N/2
- High load (N sending): can prorate overhead
  - d/(d+1)
  - average delay: N(d+1)/2

## Where the Heck Were We?

- Introduction ✓
- Multiple Access Protocols
  - contention ✓
  - collision-free ¬
- IEEE 802 Standard
- Bridges
- Misc (brief)
  - High-Speed LANs
  - Satellite Networks

## Binary Countdown

- Instead of 1 bit per station, encode in binary
  - transmit address in binary
- When multiple transmit, OR together
- When a station sees high-order 1 bit where it has a zero, it gives up

| | Bit time 0 1 2 3 |
|---|---|
| 0 0 1 0 | 0 – – – |
| 0 1 0 0 | 0 – – – |
| 1 0 0 1 | 1 0 0 – |
| 1 0 1 0 | 1 0 1 0 |
| Result | 1 0 1 0 |

## Binary Countdown Performance

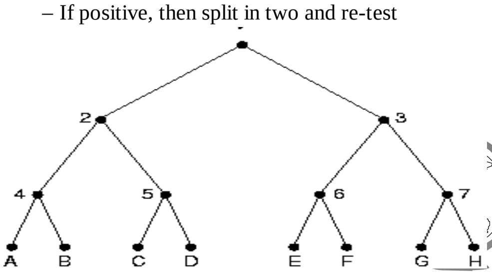
- Efficiency: $d/(d+\log_2 N)$
- Sender address as first field and *no* overhead
- Fairness?
  - Virtual station numbers
  - `C,H,D,A,G,B,E,F` are `7,6,5,4,3,2,1,0`
  - `D` sends: `C,H,A,G,B,E,F,D`

## Contention vs. Collision-Free

- Contention better under low load. *Why*?
- Collision-free better under high load. *Why*?
- Hybrid: *limited contention protocols*
- Instead of symmetric contention, asymmetric
- Divide into groups. Each group contents for same slot.
- How to assign to slots?
  - 1 per slot, then collision free (Binary Countdown)
  - All in same slot, then contention (CSMA/CD)

## Adaptive Tree Walk Protocol

- U.S. Army test for Syphilis
  - Test group, if negative all ok
  - If positive, then split in two and re-test

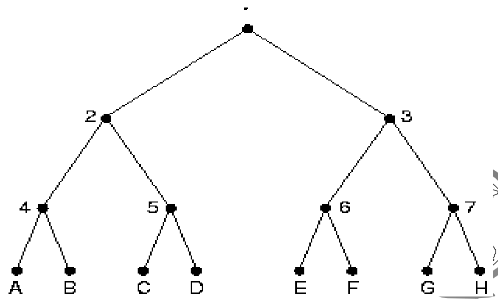



## Adaptive Tree Walk Protocol

- Where to begin searching (entire army?)
  - if heavily loaded, not at the top since there will always be a collision
- Number levels 0, 1, 2 …
- At level $i$, $1/2^i$ stations below it
  - ex: level 0, all stations below it, 1 has 1/2 below…
- If $q$ stations want to transmit, then $q/2^i$ below
- Want number below to be 1 (no collisions)
  - $q/2^i = 1$, $i = \log_2 q$

## Other Improvements

- If collision at 1, 2 idle, do we need to search 3?




## Heck, Here We Are

- Introduction ✓
- Multiple Access Protocols ✓
  - contention ✓
  - collision-free ✓
- IEEE 802 Standard ¬
- Bridges
- Misc (brief)
  - High-Speed LANs
  - Satellite Networks

## IEEE 802 Standard

- 802.3 - Ethernet
- 802.4 - Token Bus
- 802.5 - Token Ring
- Standards differ at the physical layer, but are compatible at the data-link layer

## 802.3 - Ethernet

- Began as ALOHA, added carrier sense
- Xerox PARC built 3 Mbps version for workstations and called it *Ethernet*
  - old scientist dudes thought waves propagated through substance called "ether", so a geeky joke
- Xerox, DEC and Intel made 10 Mbps standard
  - 1 to 10 Mbps
  - not "Ethernet", but close enough

## Ethernet Cabling

- 10Base5 - "Thick Ethernet"
  - 10 Mbps, 500 meters
- 10Base2 - "Thin Ethernet" or "Thinnet"
  - BNC connectors, or T-junctions
  - Easier and more reliable than 10Base5
  - But only 200 meters and 30 stations per segment
- All on one line, then difficult to find break
  - *domain reflectometry*
  - *hubs*

## Three kinds of Ethernet Cabling

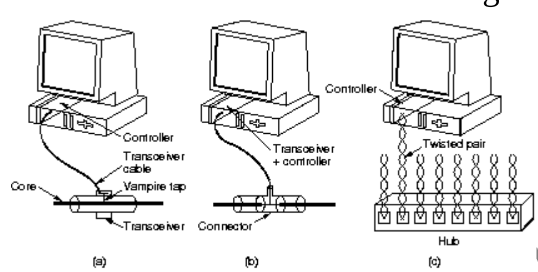
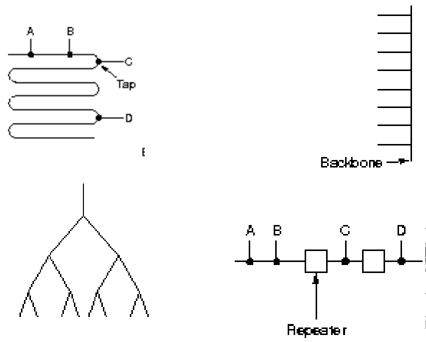


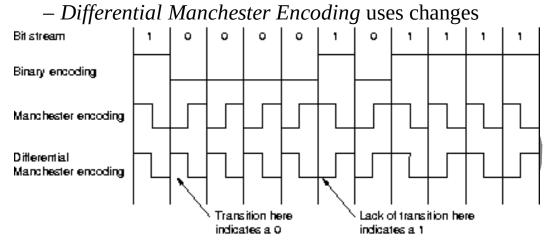

**Fig. 4-18.** Three kinds of 802.3 cabling. (a) 10Base5. (b) 10Base2. (c) 10Base-T.

## Cable Topologies



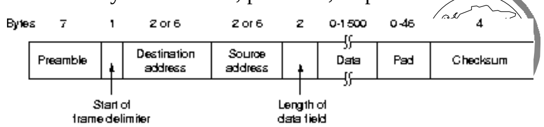

## Encoding

- 0 volts for 0 and 5 volts for 1 can be misleading
- Want start, middle and end of each bit without reference to external clock
  - *Manchester Encoding*
  - *Differential Manchester Encoding* uses changes

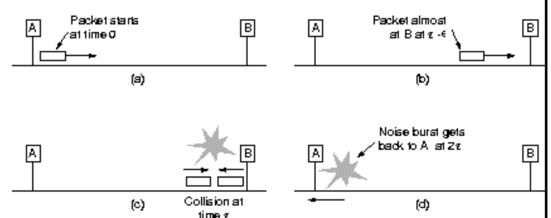



## Ethernet Protocol

- Preamble: 10101010 to allow clock synch
- Start of Frame: 10101011
- Source and Destination addr: 2 or 6 bytes
  - 1 for high order bit means "multicast"
  - all 1's means "broadcast"
- Length: data length, 46 to 1500
  - very small frames, problems, so pad to 46




## Short, Short Frames




- Frame must be > $2\tau$
- Otherwise, how to tell collision from short frame?

## Collision Action?

- If collision, then wait 0 or 1 slot
- If another collision, then wait 0, 1, 2, 3 slots
- If another collision, then wait 0 to $2^3$-1 slots
- After *i* collisions, wait 0 to $2^i$-1 slots
  - called *binary exponential backoff*
  - why is this a good idea? Consider other options
- After 10 collisions, wait 0 to 1023 slots
- After 16 collisions, throw in the towel

## Now,Where Were We?

- Introduction ✓
- Multiple Access Protocols ✓
- IEEE 802 Standard
  - Ethernet (802.3) ¬
  - Token Bus (802.4)
  - Token Ring (802.5)
- Misc

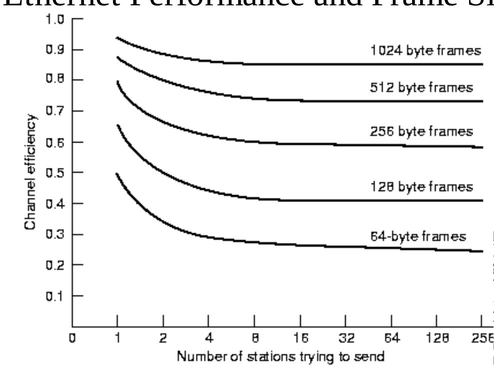
## Ethernet Performance

- Mean frame transmission time, P sec
- Probability that a frame transmits, A
  - (complicated stuff skipped)
- Channel Efficiency = $\frac{P}{P + 2\tau/A}$
- The longer the cable, the longer the contention period
  - Longest path is 2.5km + 4 repeaters, 51.2 μsecs
  - At 10 Mbps is 512 or 64 bytes, shortest frame
  - 1 Gbps Ethernet is even longer! (or shorter cable)

## Ethernet Performance (cont.)

- Convert previous formula to:
  - Frame length *F*
  - Network bandwidth B
  - Cable len *L*
  - Cable propagation speed *c*
  - (complicated stuff skipped)
- Channel Efficiency = $\frac{1}{1 + 2BLe/cF}$
- But everyone wants high-bandwidth, WAN!
  - Then they better not use Ethernet

## Ethernet Performance and Frame Size



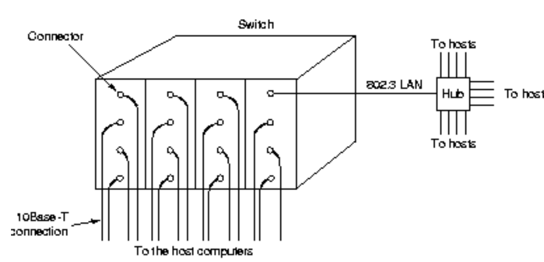

## Ethernet Perf Final Thoughts ...

- Lots of theoretical work on Ethernet perf
  - all assumes traffic is Poisson
- Turns out, traffic is self-similar
  - averaging over long-periods of time does not smooth out traffic (same variance each time interval)
  - bi-modal (packets are either big or small)
- Take models with grain of salt

## Saturated LAN

- Net saturated? Add bandwidth … good idea?
  - Expensive to replace cards
  - Efficiency
  - Instead *Switched LANs*
- Switch with high-speed *backplane* with connected *cards* (typically, 1 Gbps)
- When receives frame, sees if destined for another on same line, forwards as needed
  - different than hub or repeater
- Can reduce or eliminate contention

## Switched LANs

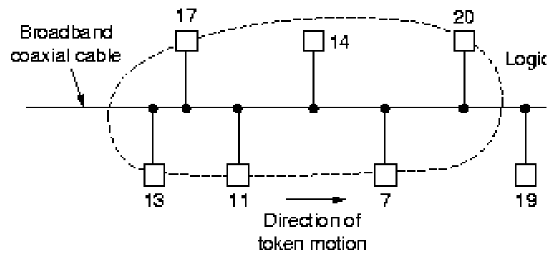



- If all input ports connected to *Hubs*, then have 802.3 to 802.3 bridge (later)

## Industry Complaints with 802.3

- Worst case transmission is unbounded
  - for automated systems, sending control signals to machines requires real-time response
- All traffic of equal importance
  - emergency shutoff better make it through
- Physical ring has constant delay
  - if *n* stations and takes *T* sec to send a frame, max is *nT* sec to wait
  - but breaks in ring will bring whole net down
  - ring is poor fit for linear assembly line
- Solution? *Token Bus*

## 802.4 - Token Bus




Physical line or tree, but logical ring. Stations know "left" and "right" stations. One token "passed" from station to station. Only station with token can transmit.

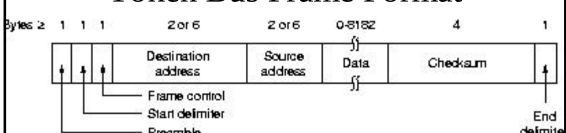
## Token Bus

- Physical order of stations does not matter
  - line is broadcast medium
- "Send" *token* by addressing neighbor
- Provisions for adding, deleting stations
- Physical layer is not at all compatible with 802.3
- A *very* complicated standard

## Token Bus Sub-Layer Protocol

- Send for some time, then pass token
- If no data, then pass token right away
- Traffic classes: 0, 2, 4 and 6 (highest)
  - internal substations for each station
- Set timer for how long to transmit
  - ex: 50 stations and 10 Mbps
  - want priority 6 to have 1/3 bandwidth
  - then 67 Kbps each, enough for voice + control

## Token Bus Frame Format



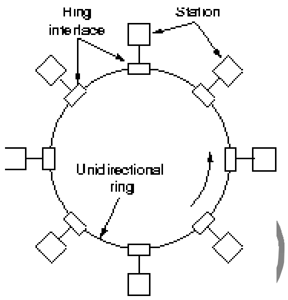

- No length field
- Data can be much larger (timers prevent hogs)
- Frame control
  - ack required?
  - Data vs. Control frame - how is ring managed?

## Token Bus Control Frame Summary

| Frame control field | Name | Meaning |
|---|---|---|
| 00000000 | Claim_token | Claim token during ring initialization |
| 00000001 | Solicit_successor_1 | Allow stations to enter the ring |
| 00000010 | Solicit_successor_2 | Allow stations to enter the ring |
| 00000011 | Who_follows | Recover from lost token |
| 00000100 | Resolve_contention | Used when multiple stations want to enter |
| 00001000 | Token | Pass the token |
| 00001100 | Set_successor | Allow station to leave the ring |

## 802.5 - Token Ring

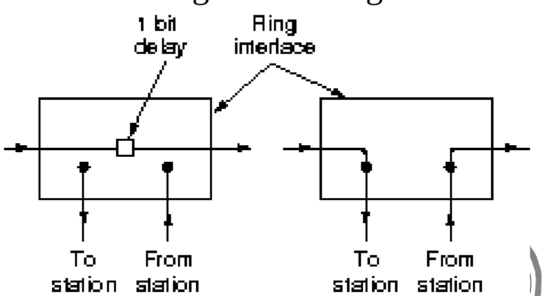
- Around for years
- Physical point-to-point connections
- Bounded delay




## Dealing with Bit “Length”

- Data rate of *R* Mbps
- Bit emitted every 1/*R* µsec
- Travels 200 m/µsec
  - each bit 200/*R* meters
- Ex: 1 Mbps ring, with 1000 meter ring can have only 5 bits on it at once!

## Reading and Writing Bits




Listen Mode

Transmit Mode

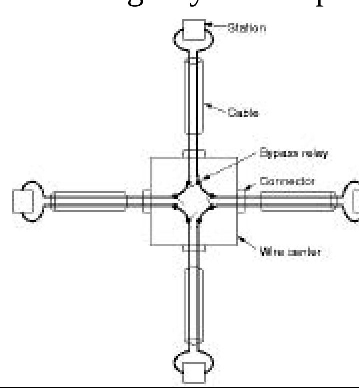
## “Token” Part of Token Ring

- Token circles around the ring
  - note, token needs to “fit” on the ring
  - if too big, then stations have to buffer, always
- When station wants to transmit, “seizes” token
  - looks like a data frame but for 1 bit
- Puts its data bits onto ring
  - no physical frame limit
- Once bits go around, removed by sender
- Regenerates token
- Acknowledgement by adding bit

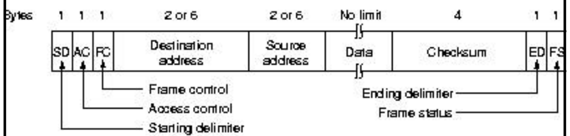
## Brief Note on Performance

- Light load
  - token circles
  - station grabs, transmits, regenerates token
- Heavy load
  - each station sends, regenerates
  - next station grabs token
  - round-robin
  - nearly 100% efficiency

## Token Ring Physical Topology



## Token Ring Sublayer Protocol




- Delimiters use invalid Manchester codes
  - End delimiter has bit for error
- *Access control* has token bit
- *Frame control* has Arrive and Check bits
  - A=0, C=0 destination not present
  - A=1, C=0 destination up, not accept frame
  - A=1, C=1 destination up, frame copied

## Ring Maintenance

- Monitor station (unlike decentralize token bus)
  - does a `claim_token` upon initial ring power-up
  - handles lost token, broken ring, cleaning ring (in case of garbage frame), orphan frame
- Timer to handle lost token
  - longest possible token cycle
  - drain ring and re-generate
- Sets monitor bit to catch orphan frame
  - if returns and is set, frame was not drained
- Extra buffer in case ring is too "short"

## Maintenance of Token Bus vs. Ring

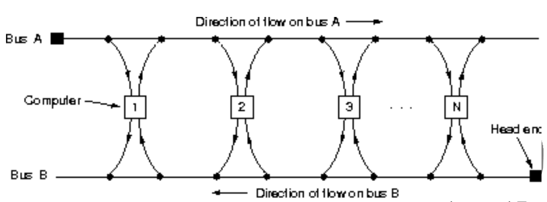
- Token bus had nothing centralized
  - all stations "peers"
  - scared that master station would go down
- Token ring felt centralized was more efficient
  - normal systems, stations hardly ever crash

## Comparison: 802.3, 802.4, 802.5

- 802.3 (Ethernet)
  - pros: popular, simple, reliable
  - cons: non-deterministic, no priorities, min frame size
- 802.4 (Token Bus)
  - pros: reliable equipment, more deterministic, priorities
  - cons: complex protocols, hard to implement in fiber, not popular
- 802.5 (Token Ring)
  - pros: fully digital, cheap to install, priorities
  - cons: delay at low load, monitor is critical component
- Usually, all perform roughly the same

## 802.6 - Distributed Queue Dual Bus

- 802.3, 802.4, 802.5 not good for MAN
  - cable length limitations
  - thousands of stations degrade performance

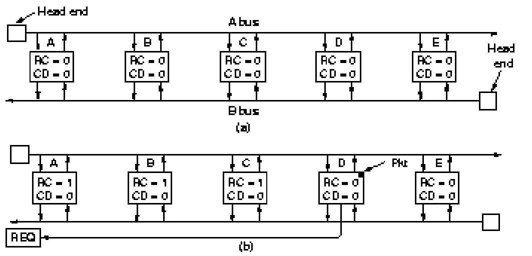



## DQDB Overview

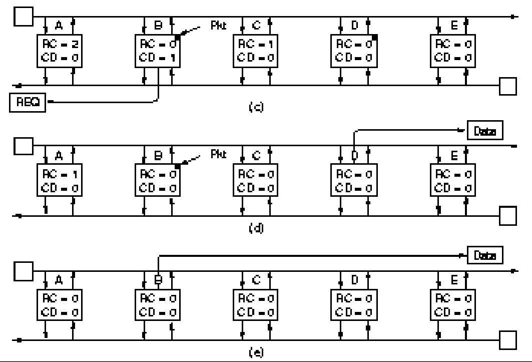
- Head End generates 53-byte cells, 44-byte data
- Cell has two bits for queue control information
  - *busy* - cell is occupied
  - *request* - station wants to transmit
- To send, station must know if destination is to left or right and use appropriate bus
- Not a "greedy" algorithm … defers to those downstream

## De-Centralized Queue

- CD = number of empties needing to go by
  - space in queue when data to send
- RC = request counter




## De-Centralized Queue




## Review

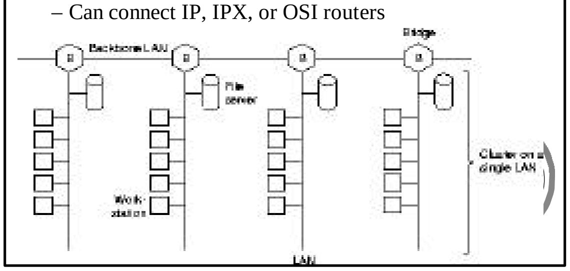
- What are:
  - 802.3?
  - 802.4?
  - 802.5?
- When does temporary token handoff occur in 802.4?
- What is the min and max data payload in 802.3?

## Where Are We?

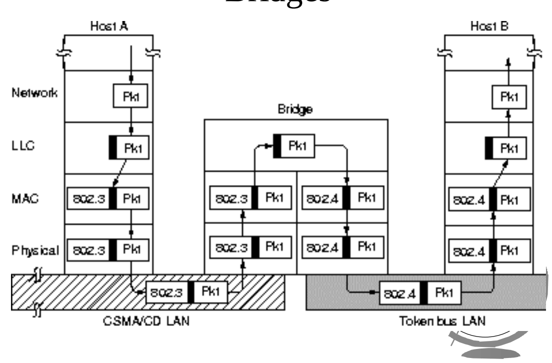
- Introduction ✓
- Multiple Access Protocols ✓
- IEEE 802 Standard ✓
- Bridges
  - issues (4.4 - 4.4.1) ¬
  - standards (4.4.2 - 4.4.5) ✗
- High-Speed LANs (4.5)
  - FDDI, Fast Ethernet
  - Fibre Channel, HIPPI

## Bridges

- Connect different LANs at the *Data Link Layer*
  - Transparently, so LANs can stay the same
  - Network layer not looked at
  - Can connect IP, IPX, or OSI routers

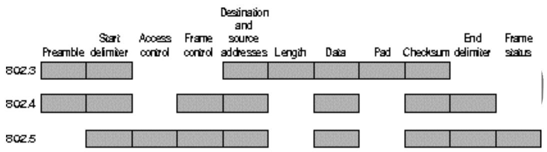



## Bridges



## What's the Big Deal?

- 802.*x* to 802.*y* give 9 combos (not 802.6, since it is not a LAN)
- Frame formats different
  - nobody (IBM, GM, Xerox) wanted to change




## What else is the Big Deal?

- Data rate
  - Fast to slow
- Frame length
  - 802.3 has limit, 802.4 bigger, 802.5 none
- Priority bits
  - 802.4 and 802.5 have them, 802.3 not
- Token handoff in 802.4
- (See 4.4.1)

## Resolving 802.x to 802.y Problems

- "Make some LAN standards!"
  - 3 incompatible LAN standards
- "Make some Bridge standards!"
  - 2 incompatible bridge standards
- "Make some Router standards!"
  - Not yet, but the trend is sorta right.
- (Not going to do bridge specifics, see 4.4.2-4.4.5)

## High-Speed LANs

- 802 LANS (and MANS) based on copper
- Fiber (mostly) for high bandwidth
- FDDI
- Fast Ethernet
- HIPPI
- Fibre Channel

## Fiber Distributed Data Interface (FDDI)

- *Token Ring* LAN, modeled after 802.5
- 100 Mbps, up to 200 km, 1000 stations
- Used primarily as backbone

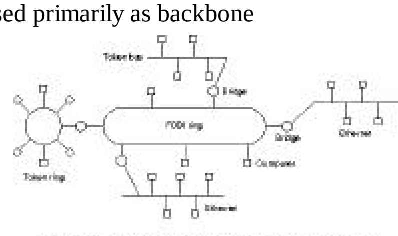



Fig. 4-44. An FDDI ring being used as a backbone to connect LANs and computers.

## FDDI

- Two fiber rings, one in each direction

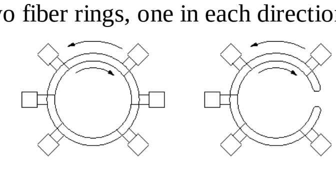



Fig. 4-45. (a) FDDI consists of two counterrotating rings. (b) In the event of failure of both rings at one point, the two rings can be joined together to form a single long ring.

- May have more than one frame in ring
  - unlike 802.5
  - more bits on wire
- Priority tokens based on timers

## Fast Ethernet

- FDDI too complicated, didn't become LAN
- Made 802.3 committee think tank
  - make Ethernet faster (winner, 802.3u)
  - make new LAN, call Ethernet (802.12)
- Change bit time from 100 nsec to 10 nsec
  - all must use hubs
  - shorter "wire-length" to hub
  - Wiring changes
  - not fiber, rather a *lot* of copper

## HIgh Performance Parallel Interface (HIPPI)

- Los Alamos National Laboratory
- Standards of 800 Mbps, 1600 Mbps
  - "Bomb" movies, 1024x1024 pixels with 24 bits/pixel at 30 frames/second needs 750 Mbps
- Not originally a LAN, but "point-to-point"
  - added switch
- Simplex
  - two wire, duplex
- Supercomputer connect

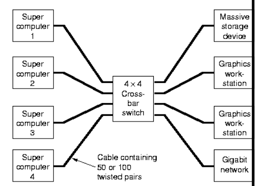



Fig. 4-48. HIPPI using a crossbar switch.

## Fibre Channel

- Designed to replace HIPPI over fiber
  - but *much* more complex
- Crossbar switch
- 200, 400 and 800 Mbps

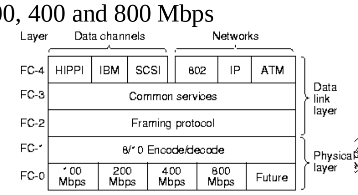



Fig. 4-49. The fibre channel protocol layers.

- Designed in U.S., name by British editor

## Review

- Describe each of the following in terms of network layers
  - Repeater
  - Hub
  - Bridge
  - Router

## Where Are We Going?

- Physical Layer ✓
- Data Link Layer ✓
  - Medium Access Sublayer ✓
- Network Layer ¬
- Transport Layer
- Katmandu