

Multiple Access Media

- multiple senders on some media
 - buses (Ethernet, including links in switched form)
 - radio, satellite
 - token rings



- need method to moderate access
 - fair arbitration
 - good performance

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 1

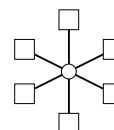
- carrier sense multiple access with collision detection (CSMA/CD)
 - used by Ethernet
 - Xerox and IEEE 802.3 (10Mbps standards)
 - IEEE 802.3u (Fast Ethernet, 100Mbps standard)
 - outline
 - historical development
 - topologies and components
 - MAC algorithm
 - collision detection limitations
 - lessons learned

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 4

- historical development of Ethernet
 - first widely used LAN
 - developed at Xerox PARC in the mid-1970's
 - employed ALOHA techniques
 - standardized by Xerox, DEC (Digital Equipment Corporation), and Intel in 1978
 - IEEE 802.3u defined Fast Ethernet (100 Mbps)
 - switched Ethernet now popular
 - scales to 10Gbps (1Gbps+ never on shared media)
 - frame compatibility
 - commodity pricing

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 5

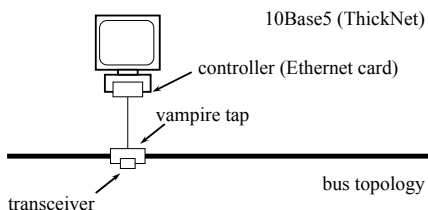
- Ethernet topologies and components
 - bus—all nodes connected to a wire
 - star—all nodes connected to a central repeater



- combinations thereof

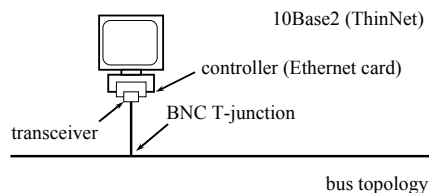
© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 6

- Ethernet topologies and components (continued)



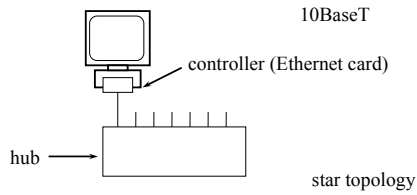
© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 7

- Ethernet topologies and components (continued)



© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 8

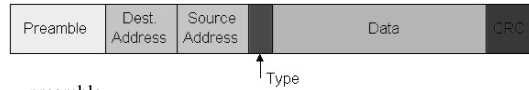
- Ethernet topologies and components (continued)



© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 9

- Ethernet frame structure

- data encapsulation
 - sending adapter encapsulates
 - data is IP datagram



- preamble
 - seven bytes with pattern 10101010
 - followed by one byte with pattern 10101011
 - Manchester coded, so result is 10MHz square wave
 - used for clock recovery

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 10

- Ethernet frame structure (continued)

- 6-byte addresses
 - if adapter receives frame with matching destination address, or with broadcast address (eg ARP packet), it passes data in frame to net-layer protocol
 - otherwise, adapter discards frame
- type: ID for higher layer protocol (e.g., IP, Novell IPX, AppleTalk)
- CRC-32: if error is detected, frame is dropped



© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 11

- Ethernet specifications

- coaxial cable
 - up to 500m for each segment
 - repeaters connect up to five segments
- taps must be > 2.5m apart
- total number of hosts < 1024
- transceiver
 - sense other transmissions (idle detection)
 - sends/receives signals

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 12

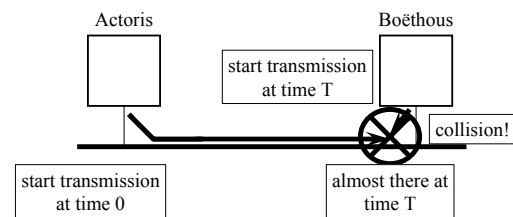
- Ethernet MAC algorithm

- MAC algorithm for senders/transmitters
 - if line is idle (no carrier sensed)
 - send immediately
 - send maximum of 1500B data (~1527B total)
 - wait 9.6 microseconds before sending again
 - if line is busy (carrier sensed)
 - wait until line becomes idle
 - send immediately (called 1-persistent)
 - if collision detected
 - stop sending and jam signal
 - try again later

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 13

- Ethernet MAC algorithm

- constraints for collision detection



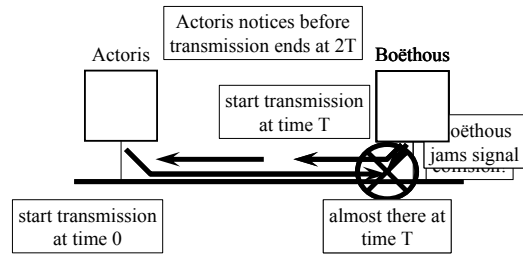
How can we ensure that Actoris knows about the collision?

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 14

- Ethernet MAC algorithm
 - constraints for collision detection (continued)
 - consider
 - Actoris' message reaches Boëthous at T
 - Boëthous' message reaches Actoris at 2T
 - thus, Actoris must still be transmitting at 2T
 - specifics of IEEE 802.3
 - bounds 2T to 51.2 microseconds
 - at 10Mbps, 512 bits or 64B
 - packet must be at least 64B long
 - jam after collision
 - ensures that all hosts notice collision

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 15

- Ethernet MAC algorithm
 - constraints for collision detection



© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 16

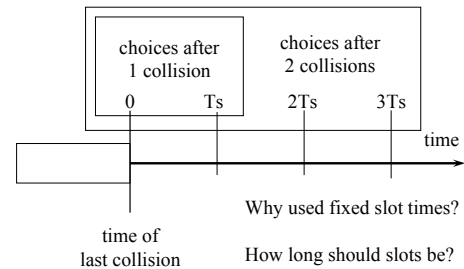
- Ethernet MAC algorithm

How long should a host wait to retry after a collision?

- binary exponential backoff
 - characteristics
 - maximum backoff doubles with each failure (exponential)
 - after N failures, pick an N-bit number
 - 2^N discrete possibilities from 0 to maximum

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 17

- Ethernet MAC algorithm
 - binary exponential backoff (continued)



© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 18

- Ethernet MAC algorithm/binary expon. backoff (cont'd)
 - T_s is 51.2 microseconds for IEEE 802.3
 - consider
 - k hosts collide
 - each picks random number from 0 to $2^{(N-1)}$
 - if minimum value is unique
 - all other hosts see busy line
 - remember that Ethernet RTT < 51.2 microseconds

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 19

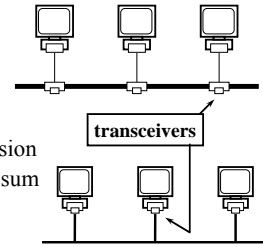
- Ethernet MAC algorithm/binary expon. backoff (cont'd)
 - consider
 - if minimum value is not unique
 - hosts in minimum value slot collide again
 - following slot idle
 - consider next smallest backoff value

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 20

- Ethernet MAC algorithm/frame reception
 - sender handles all access control
 - receiver simply pulls frames from network
 - Ethernet controller/card
 - sees all frames
 - selectively passes frames to host processor
 - acceptable frames
 - addressed to host
 - addressed as broadcast
 - addressed to multicast to which host belongs
 - anything if in promiscuous mode (packet sniffing and tcpdump requirement)

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 21

- collision detection techniques
 - bus topology Ethernets

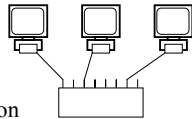


- transceiver handles
 - carrier detection
 - collision detection
 - jamming after a collision
- transceiver sees voltage sum
 - outgoing signal
 - incoming signal
- looks for voltages impossible for local alone
 - attenuation can prevent detection
 - limits segment length

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 22

- collision detection techniques (continued)
 - hub topology Ethernets

- controller/card handles carrier detection
- hub handles
 - collision detection
 - jamming after a collision
- detect transmission activity on each line
- if > 1 line active
 - assert collision to all lines
 - continue until no lines active
- attenuation is less critical



© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 23

- media for 10 Mbps Ethernet

Name	Cable	Cable Diam.	Max. Segment Length	Max. Nodes on Segment
10Base5	coaxial	10 mm	500 m	100
10Base2	coaxial	5 mm	200 m	30
10BaseT	twisted pair	0.5 mm	100 m	1 (to hub)
10BaseFP	fiber	0.1 mm	500 m	33

– extend segments with up to 4 repeaters (total of 2.5 km end to end)

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 24

- media for 100 Mbps Ethernet

Name	Cable	Max. Segment Length	Advantages
100BaseT4	twisted pair	100 m	category 3 UTP
100BaseTX	twisted pair	100 m	full duplex on category 5 UTP
100BaseFX	fiber	2000 m	full duplex, long runs

– all hub-based; other types not allowed
hubs can be shared or switched

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 25

- lessons learned from Ethernet

- number of hosts required
 - limit of 200 in practice, standard allows 1024
- range
 - typically much shorter than 2.5 km limit in standard
- round-trip time
 - typically 5 or 10 microseconds, not 50
- higher-level flow control (*i.e.*, TCP) limits load
- star topologies easier to administer than bus topologies
- (backwards) compatibility more important than elegance or raw performance ($1 > 0$)

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 26

CSMA/CD efficiency

- t_{prop} = max prop between 2 nodes in LAN
- t_{trans} = time to transmit max-size frame
 - efficiency = $1/(1+5 * t_{prop} / t_{trans})$
 - for 10 Mbit Ethernet, $t_{prop} = 51.2 \text{ us}$, $t_{trans} = 1.2 \text{ ms}$
 - efficiency is 82.6%!
- much better than ALOHA, but still decentralized, simple, and cheap
- efficiency goes to 1 as t_{prop} goes to 0
- goes to 1 as t_{trans} goes to infinity

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 27

- token ring
 - focus on Fiber Distributed Data Interface (FDDI)
 - 100 Mbps
 - was (not is) a candidate to replace Ethernet
 - used in some MAN backbones (LAN interconnects)

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 28

- token ring
 - outline
 - rationale
 - topologies and components
 - basic concepts
 - MAC algorithm (FDDI)
 - example
 - performance
 - feedback
 - token management

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 29

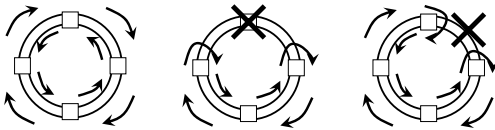
- token ring rationale

Why emulate a shared medium with point-to-point links?

- Why a shared medium?
 - convenient broadcast capabilities
 - switches costly
- Why emulation?
 - simpler MAC algorithm
 - fairer access arbitration
 - fully digital (802.3 collision detection requires analog)

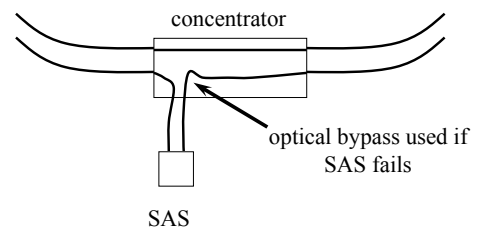
© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 30

- token ring topologies and components
 - self-healing ring
 - dual ring
 - normal flow in green direction
 - failures automatically detected and avoided



© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 31

- token ring topologies and components
 - single- and dual-attachment stations (SAS's and DAS's)



© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 32

- basic token ring concepts
 - frames flow in one direction: upstream to downstream
 - special bit pattern (token) rotates around ring
 - must capture token before transmitting
 - stations get round-robin service (for IEEE 802.5 rings, node can hold the token at most THT time units)
 - remove your frame when it comes back around

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved.

Sep-08, slide 33

- basic token ring concepts (continued)
 - two options after sending last frame
 - immediate release
 - used in FDDI
 - token follows last frame immediately
 - delayed release
 - used in IEEE 802.5
 - token sent after last frame returns to sender

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved.

Sep-08, slide 34

- FDDI MAC algorithm
 - synchronous (real-time) and asynchronous traffic
 - station i allocated S_i time units for synchronous traffic per token revolution
 - Target Token Rotation Time (TTRT) negotiated such that $S_1+S_2+\dots+S_N + \text{RingLatency} \leq \text{TTRT}$
 - algorithm goals
 - keep actual rotation time less than TTRT
 - allow station i to send S_i units of synchronous traffic per rotation (if such traffic is available)
 - fairly allocate remaining capacity to asynchronous traffic
 - if token is lost, regenerate it

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved.

Sep-08, slide 35

- FDDI MAC algorithm implementation
 - two counters at each station: TRT and THT (THT is not constant, as in IEEE 802.5)
 - assume for simplicity
 - stations stop early rather than transmit longer than allowed (in order to complete a packet)
 - packet length is known (or use worst-case)
 - upon initiating token or first time seeing token at i
 - initialize TRT to zero and commence incrementing TRT
 - transmit synchronous data (up to S_i time units)
 - release the token

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved.

Sep-08, slide 36

- FDDI MAC algorithm implementation (cont'd)
 - when station i receives the token again
 - if $\text{TRT} \geq \text{TTRT}$ (late arrival)
 - decrease TRT by TTRT (continue incrementing TRT)
 - send synchronous traffic for up to S_i seconds
 - release token

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved.

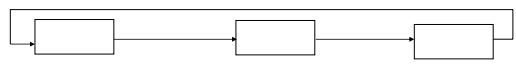
Sep-08, slide 37

- FDDI MAC algorithm implementation (cont'd)
 - if $\text{TRT} < \text{TTRT}$
 - set $\text{THT} = \text{TTRT} - \text{TRT}$ and hold
 - set TRT to zero (continue incrementing TRT)
 - send synchronous traffic for up to S_i seconds
 - resume decrementing THT
 - send asynchronous data as long as it exists and $\text{THT} > 0$

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved.

Sep-08, slide 38

FDDI example




Assume: RingLatency=12 μ s, three active stations, each with $S_i=20 \mu$ s
TTRT=100 μ s, stations have unlimited supply of asynchronous traffic.

Arrival time	TRT	s/a	Arrival time	TRT	s/a	Arrival time	TRT	s/a
0	0	0/0	4	0	0/0	8	0	0/0
12	12	20/88	124	120	20/0	148	140	20/0
172	160	20/0	196	92	20/8	228	120	20/0
252	140	20/0	276	80	20/20	320	112	20/0
344	132	20/0	368	92	20/8	400	92	20/8
432	120	20/0	456	88	20/12	492	92	20/8

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 39

FDDI example (continued)



Arrival time	TRT	A	Arrival time	TRT	A	Arrival time	TRT	A
0	0	0	8	0	0	16	0	0
12	12	88	124	120	0	148	140	0
172	160	0	196	92	8	228	120	0
252	140	0	276	80	20	320	112	0
344	132	0	368	92	8	400	92	8
432	120	0	456	88	12	492	92	8
524	112	0	548	92	8	580	88	12
616	104	0	640	92	8	672	92	8
704	92	8	736	96	4	764	92	8
796	92	8	828	92	8	860	96	4
888	92	8	920	92	8	952	92	8
984	96	4	1012	92	8	1044	92	8
1076	92	8	1108	96	4	1136	92	8
1168	92	8	1200	92	8	1232	96	4
1260	92	8	1292	92	8	1324	92	8
1356	96	4	1384	92	8	1416	92	8
1448	92	8	1480	96	4	1508	92	8
1540	92	8	1572	92	8	1604	96	4
1632	92	8	1664	92	8	1696	92	8
1728	96	4	1756	92	8	1788	92	8

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 40

- FDDI performance
 - actual token rotation time can be larger than TTRT (after some station begins using more of its synchronous allocation)
 - claim: TRT (at each station) stays in $[0, 2 * TTRT]$ (proof given on next two slides)
 - so actual token rotation time is less than or equal $2 * TTRT$
 - and, can't have actual rotation times equal $2 * TTRT$ in back-to-back rotations
 - capacity available for asynchronous traffic is fairly allocated to the stations (see problem set 3)

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 41

FDDI performance (continued)-proof of claim

A word on notation: $i-1$ and $i+1$ should be understood modulo N .
 $TRT[i]$ = value of TRT at station i
 $m[i]$ = the propagation time from station i to station $i+1$
 $S[i]$ = the synchronous allocation for node i
 Stronger claim: after the first token rotation the following holds:

Just before the token arrives at station i :

(a) monotonicity:
 $TRT[i] \geq TRT[i-1] \geq TRT[i-2] \geq \dots \geq TRT[i+1]$

(b) (largest TRT) - (smallest TRT) is less than or equal to TTRT:
 $TRT[i] \leq TRT[i+1] + TTRT$

(c) clearance for propagation time plus synchronous service time:
 $TRT[i] \leq 2 * TTRT$,
 $S[i] + m[i] + TRT[i+1] \leq 2 * TTRT$,
 and in general, for any j ,
 $S[i] + m[i] + S[i+1] + m[i+1] + \dots + S[j-1] + m[j-1] + TRT[j] \leq 2 * TTRT$

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 42

FDDI performance (continued)-proof of claim

The proof of the stronger claim is by induction. For the base case: it is easy to see that the properties are true just before the token visits the initiating station for the second time. For the induction step, suppose properties (a)-(c) are true for a particular token visit to station i . It must be shown that the properties are true just before the token visits the next station, station $i+1$. Consider two cases:

Case 1: just before the visit to station i : $TRT[i] > TTRT$. Then $TRT[i]$ is decreased by $TTRT$, and from that time until the token arrives at station $i+1$, all TRT variables increase by the same amount, which is at most $S[i] + m[i]$. So (a)-(c) are true when the token arrives at station $i+1$. This completes the proof for case 1.

Case 2: just before the visit to station i : $TRT[i] \leq TTRT$. Then TRT is computed and $TRT[i]$ is set to zero. From that time until the token arrives at station $i+1$, all TRT variables increase by the same amount. So (a) and (b) are true when the token arrives at station $i+1$. Also just after $TRT[i]$ is set to zero, it holds that $THT + TRT[j] \leq TTRT$ for all stations j , and station i can hold the token for at most $THT + S[i]$ time units. So when station i releases the token, $TRT[j] \leq TTRT + S[i]$ for all stations j , which implies that property (c) is true when the token arrives at station $i+1$. This completes the proof for case 2.

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 43

- token ring feedback
 - error detection
 - host attaches "error" marker to frame
 - sender detects error marker, resends later
 - flow control
 - host attaches "my address but did not copy"
 - sender detects problem, resends later (backs off)

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved. Sep-08, slide 44

- token management in a token ring
 - generating a token
 - at startup or when token lost
 - process
 - each host sends claim frame
 - claim includes bid for TTRT
 - lowest bid wins
 - hosts only pass lower bids on
 - receive own bid implies winning (and owning token)
 - everyone knows TTRT

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved.

Sep-08, slide 45

- token management in a token ring
 - losing a token
 - bit errors
 - node holding token crashes
 - monitoring for valid token
 - must periodically see either
 - valid frame
 - token
 - $\text{max gap} = \text{max ring latency} + \text{max frame} \leq 2.5\text{ms}$
 - set timer after each event
 - start negotiation if timer fires

© Copyright 2001 Steven S. Lumetta and Bruce Hajek. All rights reserved.

Sep-08, slide 46