Lecture 6: Datalink Layer and Local Area Networks

(10/07/2009)

Lecture Outline

1. Introduction
2. Selection issue
3. IEEE Standard 802.3: CSMA/CD bus
4. Fast and gigabit Ethernet
5. IEEE Standard 802.4: Token Bus
6. IEEE Standard 802.5: Token Ring
7. Comparisons of the Three LANs

1. Introduction

(1) Characteristics of LANs

a. Unique ownership – hence LANs are also called *private data networks*

b. A diameter of no more than several kilometers

c. High speed and robust data links

(2) Issues

a. Selection of appropriate LAN topologies

b. Selection of appropriate transmission media

c. Selection of appropriate LAN protocols (media access methods)

2. Selection issues

(1) Topology

a. Linear

b. Star

c. Ring

d. Hub-subhub

(2) Transmission media

a. Copper wire

b. Optical fiber

c. Radio

d. Hybrid

(3) LAN protocols

a. CSMA/CD

b. Control token based

c. Switch based LANs

d. Wireless LANs

(4) LAN standards

a. Three IEEE standards for classical LANs: CSMA/CD, token bus, token ring, called IEEE 802.

b. IEEE 802 was also adopted by ANSI and ISO (ISO 8802) as international standard

c. 802.1 gives introduction to the set of standards, and defines interface primitives. 802.2 describes the upper part of the data link layer using the LLC (Logical Link Control)

3. Ethernet/IEEE 802.3: **CSMA/CD** bus

(1) IEEE 802.3 protocol is a 1-persistent CSMA/CD.

a. A station will first check the channel before sending a frame.

b. If the channel is busy, it will wait. If the channel is idle, it will send its frame.

c. A collision can occur either during the sending period, or the uncertain period. The station will stop transmission immediately, if it detects the collision during the sending period. It then waits for an random period before trying retransmission.

d. After the first collision, the time is divided into discrete slots whose length is equal to the worst case round-trip propagation time 2τ.

e. The random interval is determined by the following formula called **binary exponential backoff**:

* After the first collision, the station waits for either 0, or 1 slot time before its attempt of first retransmission.
* If the first retransmission collides, the possible yielding range doubles. It will wait for either 0, 1, 2, or 3 time slots.

* In general, after the ith collision, a station will wait for a random number of slot times choosing from the range 0 to $2^i - 1$.
* After 10 collisions, the random interval is fixed between 0 and $2^{10} - 1 = 1023$ slots. The value 10 is called the *backoff limit*.
* After 16 collisions, the controller suspends retransmission and reports the failure to the data link layer. The value 16 is called the *attempt limit*.

f. Binary exponential backoff is used to dynamically adapt to the number of active stations.

* If the randomization interval is 1023 slots, the chances for two stations to collide is very slim.
* On the other hand, if the interval is 2 slots and 100 stations are trying to send, the chance for collision is very high.

(2) Data transmission and reception: Fig.3.4, p.173

(3) Frame structure (Fig. 3.3. p.171)

a. The total size of a frame is 1518 bytes. This size excludes the 7 byte preamble and 1 byte SFD. Some references call the first 8 bytes as preamble.

b. Two addresses: destination address and source address.

(a) The standard allows either a 2-byte or a 6-byte addresses. The the 10Mbps protocol uses only 6-byte addresses. Such an address consists of 12 hexadecimal numbers and is of the form:

08-00-2b-e5-fb-24

(b) High order bit is 0 for ordinary addresses, and 1 for group addresses.

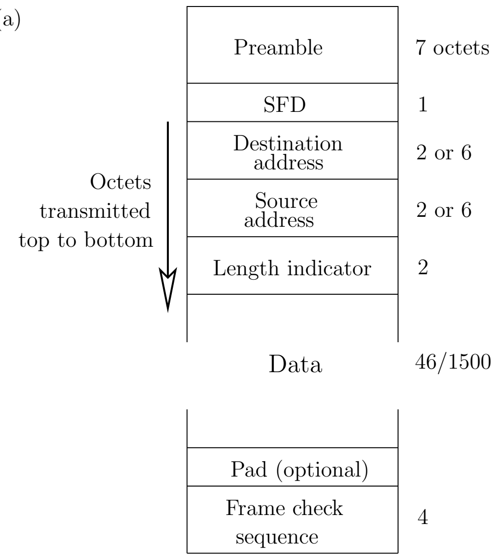
(c) Group addresses allow multiple stations to listen. When a frame is sent to a group address, all the stations in the group receive it.

(c) Sending to a group called **multicast**, while the case of address of all 1 bits is for **broadcast**. A frame containing all 1's in the destination field is delivered to all the stations, and is propagated by all bridges.

c. The bit 46 (adjacent to the high-order bit) is used to distinguish local from global addresses.

(a) Local addresses are assigned by local administrators and have no significance outside the local network.

(a)



(b)

Bit rate	10 Mbps (Manchester encoded)
Slot time	512 bit times
Interframe gap	$9.6\mu s$
Attempt limit	16
Backoff limit	10
Jam size	32 bits
Maximum frame size	1518 octets
Minimum frame size	512 bits

Figure 38: CSMA/CD bus network characteristics (a) frame format; (b) operational parameters.

(b) Global addresses are assigned by IEEE to ensure uniqueness. With $48-2=46$ bits, there are about 7×10^{13} global addresses. The network layer interprets the global address.

d. Length field tells the number of bytes in the data field, from 0 to 1500. The minimum whole frame length is 64.

e. Checksum field. it is a 32-bit hash code of the data. If some frame are received with error, the checksum will almost always incorrect.

(4) Wiring configuration: Fig.3.2, p.170

a. Cables:

(a) **10Base5**: thick-ware (0.5 inch diameter) coaxial cable with maximum segment length of 500 m.

(b) **10Base2**: thick-ware (0.25 inch diameter) coaxial cable with maximum segment length of 185 m.

(c) **10BaseT**: hub (star) topology with twisted-pair drop cables of up to 100 m.

(d) **10BaseF**: hub (star) topology with optical fiber drop cable of up to 2 km.

b. Comments:

(a) For all types of cables, the same CSMA/CD MAC method is used.

(b) 10Base5 has longer transmission range. However it is thick and difficult to install. 10Base2 has shorter transmission range, but easier to handle.

(c) 10BaseT becomes feasible for Ethernet after the inexpensive adaptive crosstalk canceler circuits were developed to eliminate NEXT (near-end crosstalk). The circuit makes it possible to use the normal UTP used on phones to transmit at bit rates of tens of Mbps over up to 100 m in range.

(d) To simulate the broadcast effect, the repeater electronics at the hub will broadcast each frame on different output pairs.

(5) *Slot* time and collision detection: Fig.3.1, p.166

a. T_P: maximum one-way propagation delay between any two stations

(a) The maximum length of cable is assumed to be 2.5 km

(b) Normally $v = 2 \times 10^8$ m/s

(c) Therefore, the worst case signal propagation delay time T_P going from one end to another end of the cable is:

$$T_P = l/v = 2.5 \times 10^3/2/ \times 10^8 = 12.5\mu s$$

b. How many bits can a station send before it eventually detects a collision?

(a) Just prior to the 1st bit of the frame (from A) is arrives at B's interface, station B senses and finds the medium is free, hence it starts its own transmission.

(b) This transmssion will collide with A's frame. However, this collision will take another T_P time to be propagated back to A. That is a total 25μs.

(c) Normally a 2.5km cable is segmented into 500 m each. These segments are connected through *repeaters*. Each repeater introduces another several μs delay. The total worst case delay is hence set to 50μs.

(d) For that total 50μs, with a link bit rate 10 Mbps, station A has transmitted

$$10 \times 10^6 \times 50 \times 10^{-6} = 500\text{bits}$$

(e) A 12 bits of safety margin is added, given a total 512 bits.

c. **Slot time**

(a) The time to send 512 bits is called a slot time. The slot time is the time that ensures that A will have detected a collision before it has transmitted its smallest frame. With 10 Mbps data rate, the 512 bit time is equal to 51.2μs.

(b) To ensure that A will be able to detect a collision, B will continue to send a special frame, called *jam sequence* that consists of 32 bits, for a small period of time.

This is the reason why Ethernet requires each frame to have at least 512 bits.

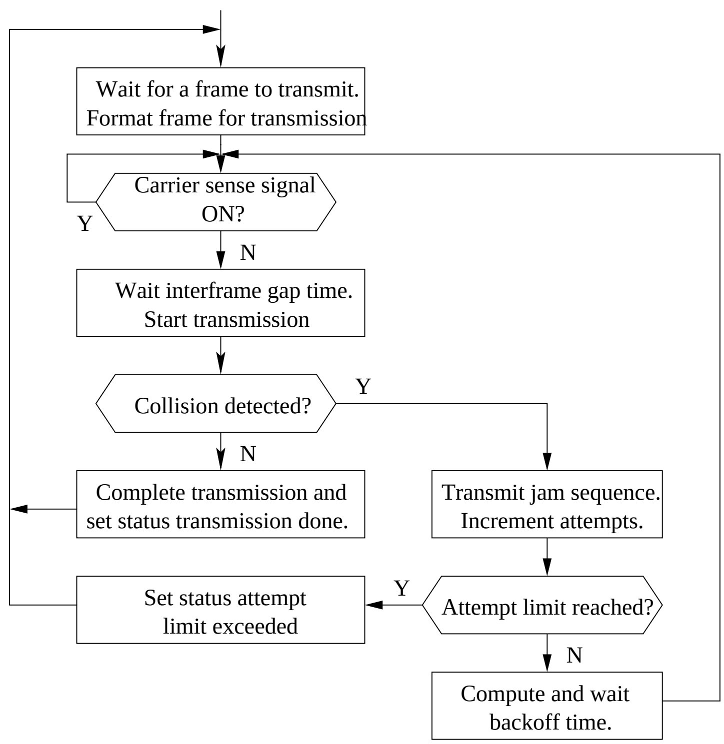


Figure 39: CSMA/CD MAC sublayer operation: (a) transmit. (Fig.3.4 (a), p.173, 1st book)

(6) 802.3 performance (under the conditions of heavy and constant load)
Assume that k stations are always ready to send, and each station transmits during a contention slot with probability p.

a. The probability A that a station acquires the channel during that slot is:

$$A = kp(1-p)^{k-1}$$

A is maximized when $p = 1/k$, with $A \rightarrow 1/e$ as $k \rightarrow \infty$.

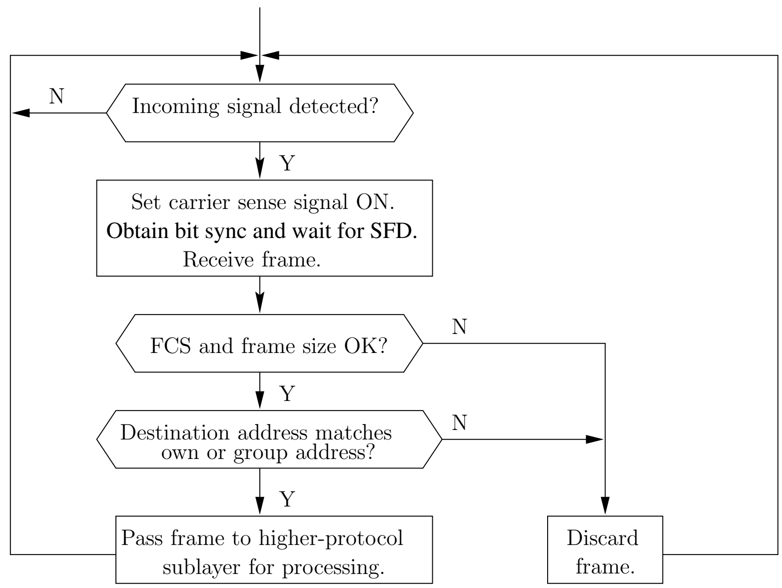


Figure 40: CSMA/CD MAC sublayer operation: (b) receive. (Fig.3.4(b), p.173, 1st book)

b. The probability that the contention interval has exactly j slots is

$$A(1-A)^{j-1}$$

The mean number of slots per contention is given by

$$\Sigma_{j=0}^{\infty} jA(1-A)^{j-1} = \frac{1}{A}$$

c. Since each slot has a duration 2τ, the mean contention interval $w = 2\tau/A$. Under any condition $w < 2\tau e \approx 5.4\tau$.

d. If the mean frame time is P, then the channel efficiency is

$$\text{Channel Efficiency} = \frac{P}{P + 2\tau/A}$$

The longer the cable, the longer the contention interval, hence the lower the efficiency.
By requiring the maximum cable length be 2.5 km, at most 4 repeaters between any two transceivers, the round-trip time 2τ can be bound to 51.2 μsec, which at 10 Mbps corresponds to 512 bits or 64 bytes, the minimum frame size.

e. Let F be the frame size, B the network bandwidth, L be the cable length, and C the propagation speed. In the optimal case, $A = 1/e$. Since $P = F/B$, we have

$$\text{Channel Efficiency} = \frac{1}{1 + 2BLe/CF}$$

The longer the cable, and the longer the bandwidth, the lower the efficiency. So 802.3 may not be suitable for long range networks

4. Fast and gigabit Ethernet

(1) Motivations

a. 10Mbps performance by the classical Ethernet is not adequate for many new Internet applications.

b. Due to the nature of the classical Ethernet, increasing the transmission rate will decrease its utilization.

c. Requirements:

(a) Such Ethernets have to be backward compatible with the classical Ethernet.

(b) Such Ethernets should not sacrifice utilization.

d. Solution: fast and gigabit Ethernets

(2) Switched Ethernets: Fig.41

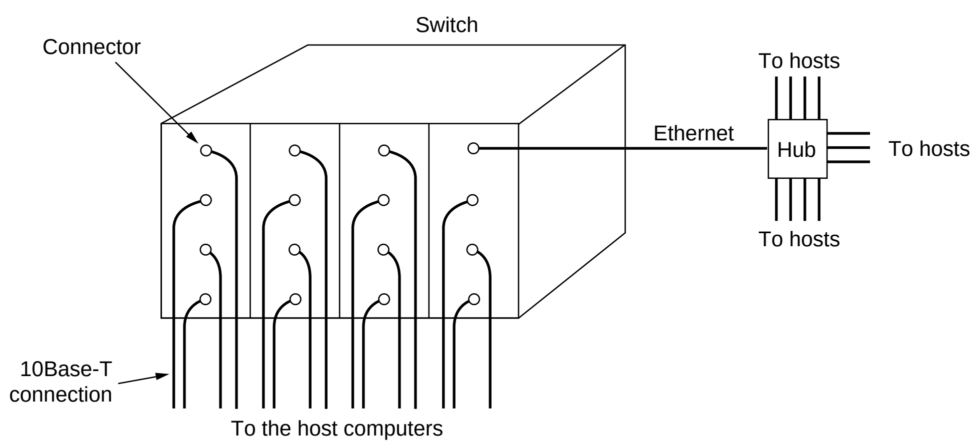


Figure 41: A simple example of switched Ethernet.

a. All fast Ethernet and gigabit Ethernets are in fact switched Ethernets.

b. Gigabit Ethernets also use hubs.

(3) Fast Ethernets

a. Motivations:

(a) Utilization issue

(b) Compete/replace **FDDI** (fiber distributed data interface) and **Fiber Channel**. Both can provide bandwidth upto 100Mbps. But they are not compatible with standard PC networking interfaces.

(c) KISS (Keep it simple, stupid) philosophy

b. 1992 debate: two choices

(a) Make the classical 802.3 faster. This group lost and eventually drafted standard 802.12

(b) Reinvent it altogether with many additional features. This choice won and the standard 802.3u was drafted. It is called fast Ethernet.

c. Original fast Ethernet cabling standards: Fig.42

(a) **100Base-T4** is a scheme based on category 3 UTP and provides a signaling speed of 25MHz. To achieve 100Mbps bandwidth, four such wires are needed. Fits well for offices with four twisted pair telephone lines.

(b) **100Base-TX** is category 5 UTP and can handle clock rates of 125MHz. Two such wires are used, one to hub and one from hub.

(c) **100Base-FX** uses two multimode fibers.

The first two are often called **100Base-T**.

d. One more cabling, **100Base-T2** was added in 1997. 100Base-T2 allows fast Ethernet to run on two category 3 wiring. Complex signaling techniques are needed to support the transmission rate.

e. 100Base-T fast Ethernets allow two kinds of interconnection devices (Fig.4.20): hubs and switches.

f. Backward compatible with 802.3

Name	Cable	Max. segment	Advantages
100Base-T4	Twisted pair	100 m	Uses category 3 UTP
100Base-TX	Twisted pair	100 m	Full duplex at 100 Mbps (Cat 5 UTP)
100Base-FX	Fiber optics	2000 m	Full duplex at 100 Mbps; long runs

Figure 42: The original fast Ethernet cabling.

(4) Gigabit Ethernets

a. Drafted in 1995 as 802.3z.

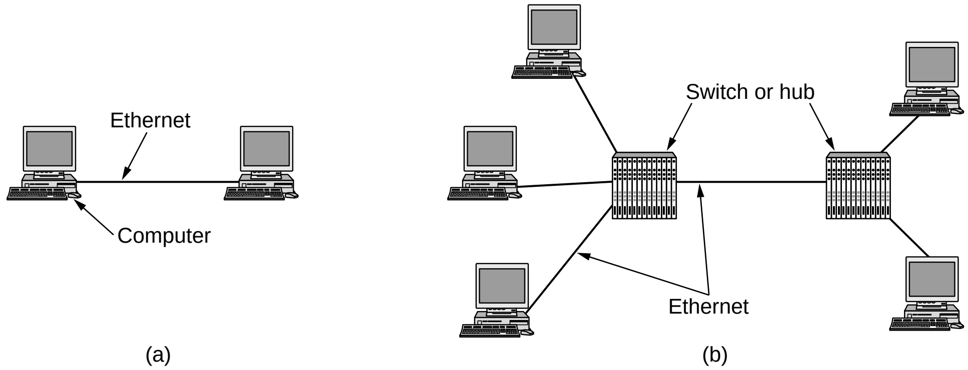


Figure 43: (a) A two-station Ethernet. (b) A multistation Ethernet.

b. Features

(a) All configurations are point-to-point: Fig.43.

(b) Support of unacknowledged datagram services with both unicast and multicast.

c. Specifications: Fig.44

d. Backward compatible with 802.3 and 803.3u

Name	Cable	Max. segment	Advantages
1000Base-SX	Fiber optics	550 m	Multimode fiber (50, 62.5 microns)
1000Base-LX	Fiber optics	5000 m	Single (10 μ) or multimode (50, 62.5 μ)
1000Base-CX	2 Pairs of STP	25 m	Shielded twisted pair
1000Base-T	4 Pairs of UTP	100 m	Standard category 5 UTP

Figure 44: Gigabit Ethernet cabling.

5. IEEE Standard 802.4: Token Bus

(1) Multi-access with single access

a. Problems with 802.3 for industrial applications. IEEE 802.3 is suitable for office and academic environments. However it has severe problems with regard to industrial applications.

(a) The probabilistic features (binary exponential backoff) may cause a station to wait for an arbitrary long time. In other words, no worst case waiting time is known for any frame.

(b) All stations are equal to compete for the use of the cable. As a result frames are given priorities. This makes it unsuitable for real-time applications.

b. Single access protocols: each station takes turn to send frames. Access to the medium is controlled by possession of a special control frame called *token*.

  (a) The medium can be a bus or a physical ring.

  (b) As access is the medium is regulated, there are no collisions during normal operations.

  (c) The worst case waiting time in a single-access protocol is known. Suppose each station uses the cable for a time T in its turn, then in worst case, a station has to wait for nT time to send its next frame.

  (d) Priority can be easily introduced.

(2) 802.4 is a collision free protocol.

a. The allowed bandwidths are 1, 5, or 10 Mbps.

b. Token bus uses the 75-ohm broadband coaxial cable used for cable television.

c. The physical layer is much more complicated than in 802.3

(3) The protocol

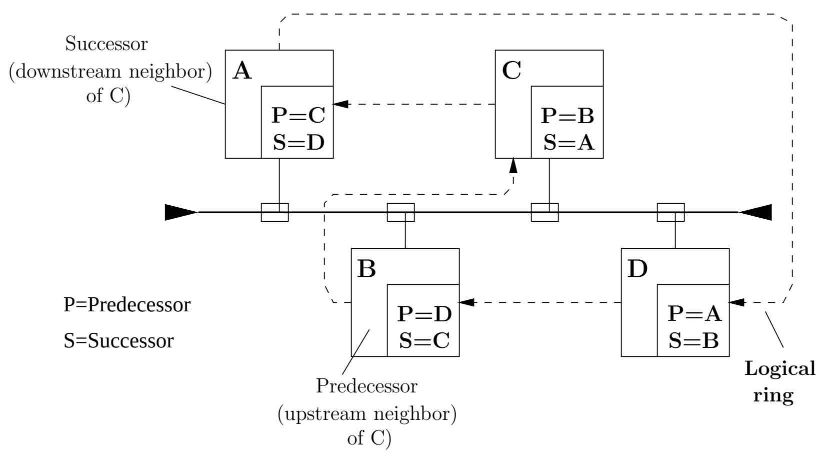


Figure 45: Token bus network principle of operations (Fig.16.20,p.309, 1st book)

a. Physically, the token bus is a linear or tree-shaped cable attached with stations (Fig.6.20,p.309).

b. The whole network is logically organized as a ring. A station knows its left and right neighbors

c. A special control frame is circulating around the logical ring. Only the token holder is allowed to transmit frames. When a station receives the token, it may transmit its frames for a certain amount of time, and then it must pass the token to one of its neighbor stations. If a station does not have any data to send when it receives the token, it passes the token immediately.

d. Frames are divided into four priority classes: 0, 2, 4, and 6, with 0 being the lowest, and 6 the highest.

  * Each station maintains four queues, one for each priority. A timer is associated with each queue. Each frame coming from the Data Link Layer is put into one of the four queues according to its priority.
  * When the token arrives at a station, the queue with priority 6 is checked first. If that queue has frames in it, the station will send those frames first until the timer expires.
    Then the queue with priority 4 is checked, and so on.
    The queue with priority 0 is checked the last.
  * When the timers are set appropriately, a guaranteed portion of the time can be devoted to each queue.
    For instance, assume that the network has 50 stations and the bandwidth is 10 Mbps. If the priority 6 queue is given 1/3 of bandwidth, then each station has a guaranteed 67 kbps bandwidth for priority 6 queue.

(4) Frame structure (Fig.6.19(c), p.308)

Preamble	SD	FC	Dest. Addr.	S. Addr.	Data	FCS	ED
($>$ 1 octet)	(1)	(1)	(2 or 6)	(2 or 6)	$<$ (8191)	(4)	(1)

a. The *preamble* is used to synchronize the receiver's clock. Notice that it may be as shorter as 1 byte.

b. The *start delimiter* and *end delimiter* fields mark the frame boundaries. They contain analog encoding of symbols other than 0s and 1s, so that they can not accidentally appear in the user data.

c. The *frame control* field is used to data frames from control frames. For data frames, it carries the frame's priority.
It also contains indication whether the receiver station should send acknowledgment.

d. Two addresses: destination address and source address: Same as in 802.3.

e. Data can be 0 to 8190 bytes. When the 2-byte addresses are used, data can be up to 8190 bytes, and when 6-byte addresses are used, data can be up to 8182 bytes. (In 802.3 the frame size is smaller to prevent a station from using the cable too long. In 802.4, frames can be longer, since timers are used to enforce cable holding time)

f. Checksum field. It is the same as in 802.3.

(5) Logical ring maintenance: from time to time, some stations might want to leave the network (shut down power), while some other stations might want to join the network (power on).

a. Token bus control frames:

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 the ring
00001000	Token	Pass the token
00001100	Set_successor	Allow station to leave the ring

b. Soliciting stations who intend to enter the ring:

(a) Each station maintains the addresses of its predecessor and successor's stations.

(b) Periodically the token holder solicits bits from stations not currently in the ring that wish to join by sending the control frame *Solicit_successor_1*. This frame contains the sender's address and its successor's address. Time is divided into slots (2τ each) immediately after the control frame was sent. Only stations whose addresses are between these two addresses are allowed to bid for joining the ring.

(c) The soliciting station waits for a periods of 2τ to see if any stations are interested in entering the ring. If no stations are interested, the token holder resumes its normal business. If exactly one station (say station i) bids to enter the ring, this station is allowed to do so and become the token holder's successor, while the original successor of the token holder becomes station i's predecessor.

(d) If two or more stations want to join the ring, they will all send their intention frames, and these intention frames will collide.

(e) The token holder resolves the contention by running an *arbitration algorithm*, starting by sending the control frame *Resolve_contention*.

(f) To further reduce the collision possibility, each stations maintains two random bits. A station who wants to join the ring uses these two random bits to decide if it should wait for 0, 1, 2, or 3 time slots before it sends its intention frame.

(g) The activity of soliciting new stations is not allowed to interfere the guaranteed worst case token rotation time. To achieve this, each station maintains a timer which was set each time it passed the token to its successor. Every time the token arrives, the token holder inspects this timer. If the time left on the timer is less than certain threshold value, then it knows that the network is heavily loaded. Therefore the token holder will not initiate the soliciting activity.

(h) When a station X (with predecessor P and successor S) wants to leave the ring, it sends the control frame *Set_successor* to station P. Station P will then set station S as its new successor.

(i) Ring initialization is a special case of adding new stations. When a station is powered on, it first waits for certain amount of time. If no traffic is observed during this period, this station would assume that all other stations are down, and send the control frame *Claim_token*.

   If not hearing response after sending this frame, it creates a token and sets up a ring only containing itself.

   Periodically this station solicits bids from other stations who want to join the ring.

   If two or more stations are powered on at the same time, an algorithm will be used to let them bit for the token.

(j) Failure stations or transmission errors that cause the missing of the token are handled as follows. After the token holder passes the token to its successor S, it listens to the cable for certain period of time. If frame transmission or token passing is detected, the old token holder knows that its successor is active and token has been correctly delivered. If no traffic is observed, the old token holder pass the token the second time. If this failed again, the old token holder assumes that its successor fails, and sends the control frame *Who_follows* which contains its successor's address. When the station who is S's successor receives this frame, it would know station S failed, and respond by sending the control frame *Set_successor*.

If S's successor also failed, the old token holder would send the control frame *Solicit_successor_2*. In this case, all stations must bid to join the ring.

(k) If the token holder fails, the token would be disappear. This is handled in the same way as ring initialization. Each station maintains a timer to monitor if there is no any traffic for a threshold amount of time. If no any traffic is detected, the timer will go off, and at least one station will come out to send the control frame *Claim_token*. Contention is solved again by an arbitration algorithm.

(l) Multiple tokens are resolved as follows. A station who is holding a token will discard its token when it sees another token. If all tokens are discarded, the initialization procedure is employed.

(6) Priority scheme in details

a. Frames are divided into four priority classes: 0, 2, 4, and 6, with 0 being the lowest, and 6 the highest.

   (a) Class 6: urgent messages (mission critical frames)
   (b) Class 4: ring maintenance and control frames
   (c) Class 2: messages relating to routine data logging and gathering
   (d) Class 0: messages relating to program downloading and file transfers

b. Each DTE maintains two timers that are used to control transmissions of frames:

   (a) *THT (token hold time)*: records the time elapsed since token was received.
   (b) *HP-THT (high-priority token hold time)*: records the time it has spent on high priority frames since token was received.

c. Each DTE also maintains two variables *TRT (token rotation time)* and *TTRT (target token rotation time)*:

   (a) TRT: it records the time that has elapsed since it last received the token.
   (b) TTRT: it records a constant value that represents the maximum allowed token rotation time for transmitting lower-priority frames.

d. When a DTE receives the token, it performs the following operations:

   (a) Transferring the value of TRT into THT.
   (b) Resetting TRT to 0.
   (c) Transmitting any queued high-priority frames. The timer TRT is ticking.
   (d) Computing the difference $\alpha = TTRT - THT$
      · If $\alpha > 0$, the DTE can spend time to send any lower-priority frames.

- If $\alpha \leq 0$, the DTE cannot send any lower-priority frames on this pass of token and it has to release the token.

e. Example (Fig.6.22, p.314)

(a) Assumptions:

i. Four stations DTE 9, DTE 7, DTE 5 and DTE 1 are logically connected in a ring in that order, with DTE 1 connecting back to DTE 9.

ii. The token passing time and propagation delay are negligible.

iii. There are two priority classes used. DTE 9 and 1 only have high-priority frames to send and DTE 7 and 5 only have lower-priority frames to send.

iv. TTRT for lower-priority frames is fixed at 8 units of time.

v. All data frames are of fixed length and transmitting each frame takes 1 unit of time.

vi. DTE 9 and 1 only can send up to three high-priority frames each time they hold the token.

vii. The XMIT values in the diagram are the number frames transmitted by a DTE each time it receives the token.

(b) Rotations in the diagram

Rotation 1 At the beginning all DTEs are inactive. The TRT at every DTE starts with value 0.

i. DTE 9 receives the token starts to hold token and transmit 3 high-priority frames. It resets TRT to 0 and releases the token.

ii. AT DTE 7, its TRT is 3 when the token arrives. It takes the token and transmit TTRT-TRT = 5 frames. It releases the token and resets TRT = 0.

iii. At DTE 5, its TRT=8. It has to release the token and resets TRT=0.

iv. At DTE 1, its TRT=8. But DTE 1 is not restricted by the value TTRT-TRT as it has high-priority frames. It sends three frames and releases token, resetting TRT=0.

Rotation 2 At DTE 9, TRT=11. Again, it is not restricted by TTRT-TRT. It transmits 3 frames and releases the token.

i. At DTE 7, TRT=11. TTRT-TRT = -3 < 0. It cannot use the token.

ii. At DTE 5, TRT=6. It grabs the token and sends TTRT-TRT = 2 frames.

iii. At DTE 1, TRT=8. It grabs the token and sends a maximum allowed 3 frames.

Rotation 3 At DTE 9, TRT=8. Again, it takes the token and sends 3 frames.

i. At DTE 7, TRT=8. TTRT-TRT = 0. It again cannot take the token.

ii. At DTE 7, TRT=8. TTRT-TRT = 0. It cannot take the token.

iii. At DTE 1, TRT=6. It grabs the token and sends a maximum allowed 3 frames.

Rotation 4 At DTE 9, TRT=6. Again, it takes the token and sends 3 frames.

i. At DTE 7, TRT=6. TTRT-TRT = 2. It takes the token and transmits 2 frames.

ii. At DTE 7, again TRT=8. TTRT-TRT = 0. It again cannot take the token.

iii. At DTE 1, TRT=8. It grabs the token and sends a maximum allowed 3 frames.

Rotation 5-12 Similar to above.

(c) Discussions: in this particular example, the two lower-priority DTEs get shares of TTRT-TRT quite equally.

5. IEEE Standard 802.5: Token Ring

(1) Introduction

a. Ring structures can be used for both LANs and WANs. Ring technique are well developed.

b. IEEE 802.5 was based on the IBM's ring networks. IEEE 802.5 is called *token ring*.

(2) Physical ring structure.

a. A ring consists of a collection of ring interfaces connected by point-to-point lines.

b. Each bit arrives at an interface is copied into a 1-bit buffer and recopied out onto the ring again. The bit in the buffer can be inspected and possibly modified before being copied out. This copying introduces 1-bit delay at each interface.

(3) Basic operations (Fig.6.12,p.293)

a. Like in token bus, a special control frame pattern, called **token** is circulating around the ring.

b. Every frame or token will circulate around the ring and thus can be received by every station.

c. Each station is attached to an interface on the ring. Each interface operates in two modes: listen and transmit. The interface must copy every bit passing by to buffers so that the bit sequence can be inspected. The interface must be able to switch from listen to transmit mode in one bit time.

d. When a station wants to transmit its frames, it first must obtain the token, and remove it from the ring. It then starts its transmission. After transmission, it must return the token to the ring. Therefore token rings have no collisions.

e. The sender of a frame is responsible for removing it from the ring.

f. In theory there is not limit on the frame size, because the entire frame never appear on the ring at one time. All bits propagated back to the sender station must be removed from the ring. Error checking can then be done by the sender.

g. Acknowledgment is simple. The sender attaches a bit 0 at the end of the frame. The receiver converts the bit to 1 if the frame passes checksum exam at the receiver. When the sender sees the converted bit 1, it knows that the frame is received.

(4) *Wire center*. Wire center is used to prevent the shut-down of the whole network caused by the shut-down of one station. By-pass relay is energized by the current from the station. If the station goes down, the loss of power will release the relay and bypass the station. The use of wire center results in **Star-shaped ring.**

(5) Ring length

a. The ring itself must have sufficient delay to allow a complete token to circulate.

b. **Physical length of ring of a bit**. If the data rate is R Mbps, a bit is emitted every $1/R$ μsec. Normally the signal propagation delay is 200 m/μsec. Each bit will occupy $200/R$ meters on the ring.
This implies that if $R = 1$ Mbps, and the ring circumference is 1000 meters, the ring can contain only 5 bits on it at any time.

c. Two components of delays: The one bit delay at each interface, and the signal propagation delay. If the interface is powered from the station and the station is shut down during night, the one bit delay is removed. Artificially delay may be necessary.

(6) Frame Structure

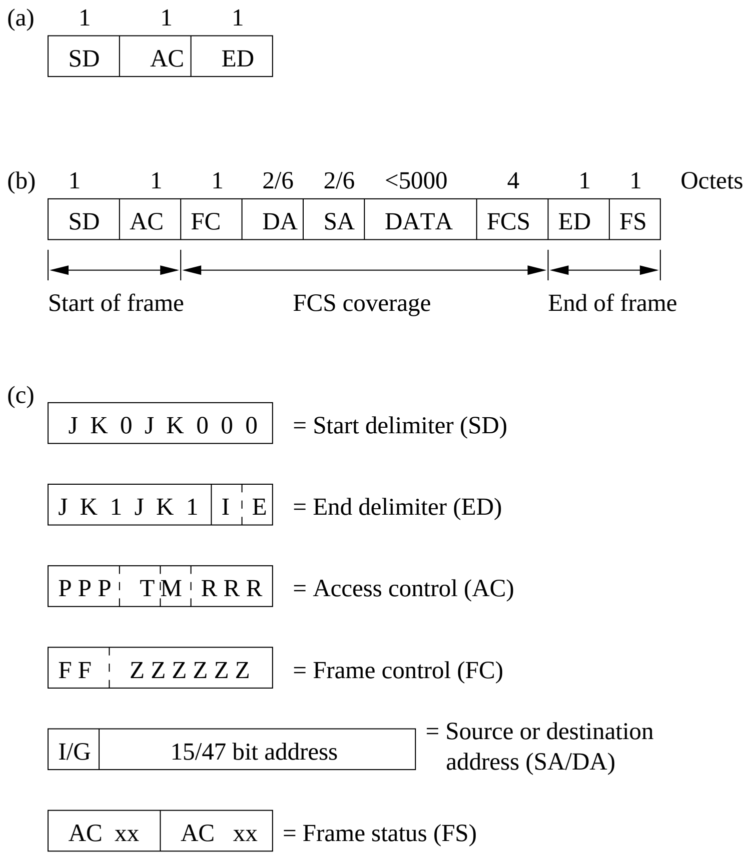


Figure 46: Token ring network frame formats and field descriptions: (a) Token format; (b) frame format; (c) field descriptions (Fig.16.14,p.296, 1st book)

a. SD: starting delimiter

b. AC: access control bits. Has four components: token bit, monitor bit, ring priority bits, and reservation bits

c. FC: frame control byte

d. ED: ending delimiter

e. FS: frame status byte

(7) The ring protocol (Fig.6.15, p.299)

a. A 3-byte token circulates around the ring, waiting for some station to seize it. When a station seize the token, it first converts a specific 0 bit in the second byte of the token to 1. It then puts a frame control (FC) byte and the rest of a normal frame.

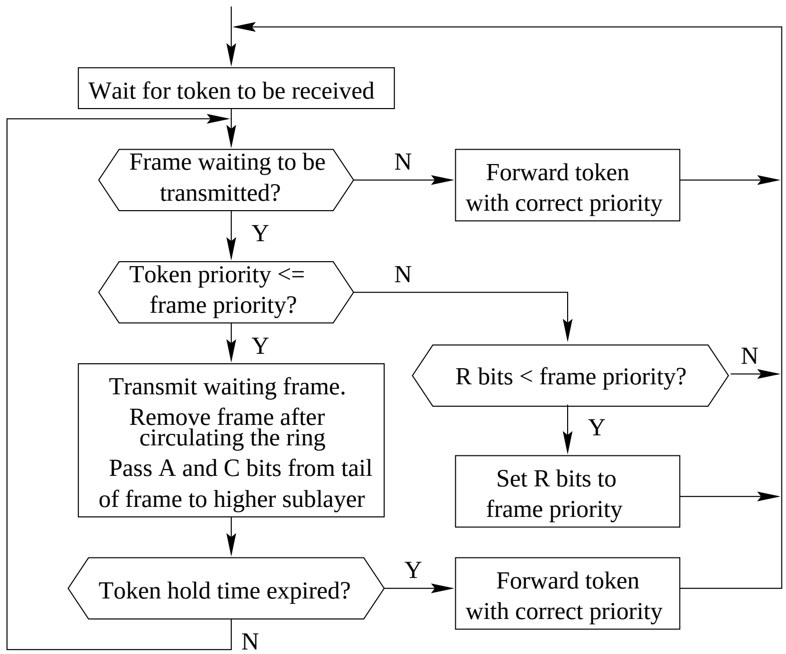


Figure 47: Token ring MAC sublayer operation: (a) transmit (Fig.16.15(a),p.299, 1st book)

b. The first bit of a data frame will normally return the sender before the full frame has been completely sent out. Only a very long ring can hold a even very short frame. The sender should drain the returned bits.

c. **Token holding time.** Each station is allowed to hold the token for a period of token holding time (10 msec). The station can transmit frames during this interval, and release the token when the time is up.

d. The *starting delimiter* and *ending delimiter* fields mark the start and the end of a frame. They contain invalid differential Manchester patterns (HH and LL) to distinguish them from the normal data bytes. The *ending delimiter* also contains an E bit that can be set if any interface detects an error (non Mach ester pattern), and a bit that marks the end of a sequence of logical frames.

e. The **Access control** byte contains four special components:

   * The token bit
   * The *Monitor bit*
   * *Priority bits* (2 bits)
   * *Reservation bits* (2 bits)

f. Priorities are handled differently here. The *access control* byte contains priority bits. A station who wants to transmit a frame with priority n must wait until a

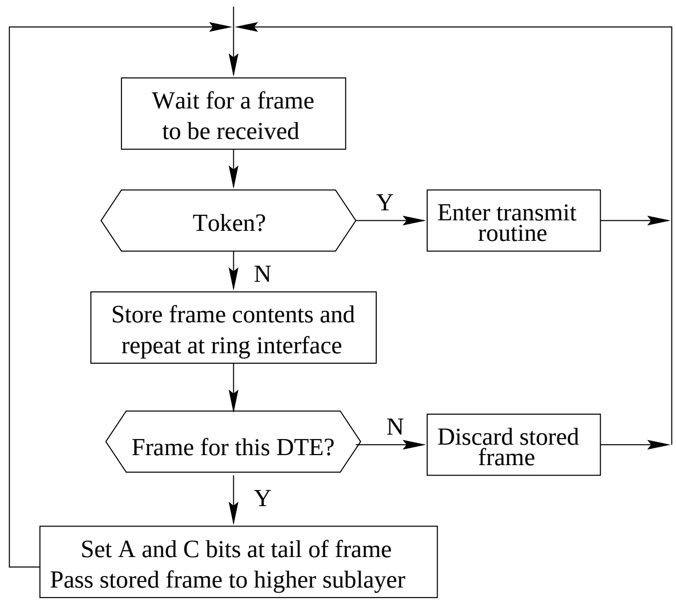


Figure 48: Token ring MAC sublayer operation: (b) receive (Fig.16.15(b),p.299, 1st book)

token with priority less than or equal to n arrives. Further, a station who wants to transmit a priority n frame may try to reserve the token by writing the priority n into the reservation bits. If a higher priority has been reserved, the station can not make reservation. The station who raise the reservation priority must lower the priority once it finishes its transmission.

g. The **Destination address**, **Source address**, **Checksum** fields are the same as in 802.3 and 802.4.

h. The **Frame status** byte is unique. This byte contains two bits: A and C bits, which are set to 0 during transmission. They can be used for acknowledgment. The three possibilities are:

(a) $A = 0$ and $C = 0$, destination not present or not powered on

(b) $A = 1$ and $C = 0$, destination present but frame not accepted. Can be thought as negative ack.

(c) $A = 1$ and $C = 1$, destination present and frame copied.

(8) Ring Maintenance.

a. Token bus handles ring maintenance in a distributed fashion, in the sense no central controller is present. Token ring handles the ring maintenance in a centralized fashion. Each token ring has a **monitor station** which oversees the ring. If the monitor station goes down, the contention algorithm will elect another monitor station. Each station has the ability to become the monitor station.

b. Monitor's responsibilities:
   (a) Seeing if the token is lost;
   (b) Taking actions if the ring breaks;
   (c) Cleaning up the ring when garbled frames appear;
   (d) Watching for and cleaning out orphan frames.

c. To check for if the token is lost, the monitor has a timer that is set to sum of the longest token holding time of each station. If the timer goes off, the monitor drains the ring and issues a new token.

d. If a frame is damaged, the monitor detects it by its invalid format or checksum. The monitor then drains the ring and issues a new token when the ring is clean.

e. The monitor detects orphan frame by using the monitor bit in the access control byte.

f. Monitor introduce extra delay if necessary.

Monitor can not be challenged.

(9) Control frames

Frame control field	Name	Meaning
00000000	Duplicate address test	Test if two stations have same addresses
00000010	Beacon	Used to locate breaks in the ring
00000011	Claim token	Attempt to become monitor
00000100	Purge	Reinitialize the ring
00000101	Active monitor present	Issued periodically by the monitor
00000110	Stand by monitor present	Announces the presence of potential monitor

a. Claim token control frame. Issued when stations suspects the monitor is down.

b. Beacon control frame. Issued if a station suspects that its neighbor is down.

c. Active monitor present control frame. Issued by the monitor periodically.

(10) Priority scheme in details

a. Basic ideas

(a) The MAC unit at each interface will assign a priority value to each frame from its up layer.

(b) There are four priority levels, 0, 2, 4, 6, with 0 the lowest and 6 the highest (same as in Token bus).

b. *Current ring service priority*. This priority is contained in the priority component of the AC field in a token or data frame.

c. Each MAC unit maintains three variables and two stacks:

(a) Three variables

i. Variable P_m, whose value is equal to the highest priority value of any frames waiting to be transmitted.

ii. Variables P_r and R_r, together called *priority registers*. They contain, respectively, the priority and reservation values held within the AC field of the most recently repeated token or frame.

(b) Two stacks S_r and S_x. They are used when the MAC unit becomes a *stacking station*. S_r is the *old priority stack* and S_x is the *new priority stack*. A stack station is one which has to raise the ring service priority due to reservations or its need of transmitting frames with priorities higher than the current service priority.

d. Obtaining a token and transmitting frames

(a) For a station with frames waiting to be transmitted, if a token is passing by with priority P (inside the AC field) less than or equal to P_m, the station can take the token and begin transmitting data frames with priority larger than or equal to P_r. Each such transmitted data frame will carry a priority P_r and reservation bits 0 inside its AC field.

(b) A token holding station can transmit queued frames with priority larger than or equal to P_r until it either has transmitted all such frames or until its token holding time expires. It releases the token by generating a new token according to the following rules:

i. If (i) the station does not have any queued frames with priority larger than or equal to P_r; or (ii) the current value R_r is not greater than P_r, it will generate a new token with

$$P = P_r \quad \text{and} \quad R = max(R_r, P_m)$$

Keep in mind the values R_r can be changed by other stations while the current station was transmitting frames.

ii. If (i) the station still has queued frames with priority P_m larger than or equal to P_r; or (ii) the current content of variable R_r is greater than P_r, it will generate a new token with

$$P = max(R_r, P_m) \quad \text{and} \quad R = 0$$

Keep in mind the values R_r can be changed by other stations while the current station was transmitting frames.

iii. In Case ii above, the station becomes a *stacking station* as it raises the ring service priority. The station has to keep the previous ring service priority in stack S_r and the new ring service priority in stack S_x. The intention is that this station has to lower the ring service priority later on when the station who made higher priority reservations (or itself because it has higher priority frames to transmit) finished transmission.

The operation details of a token releasing station is summarized in Fig.6.16(a), p.301.

(c) Operations of a stacking station.

i. A stacking station examines every token with a priority equal to the value S_x. It will examine the reservation bits in that token to determine whether the ring service priority should be raised again, lowered, or just maintained.

ii. If the value in the R bits of the AC field of the token is greater than that of S_r, it releases a token with

$$P = R_r \quad \text{and} \quad R = 0$$

The new ring service priority P will be stacked again onto S_x. The station continues to be a stacking station.

iii. If the value in the R bits of the AC field of the token is less than or equal to that of S_r, it releases a token with

$$P = S_r \quad \text{and} \quad R = R_r$$

(the R bits unchanged). The station popped stacks S_r and S_x. If both stacks become empty, it ceases to be a stacking station.

The operation details of a token releasing station is summarized in Fig.6.16(b), p.301.

e. Example: p.302,303, Table 6.1

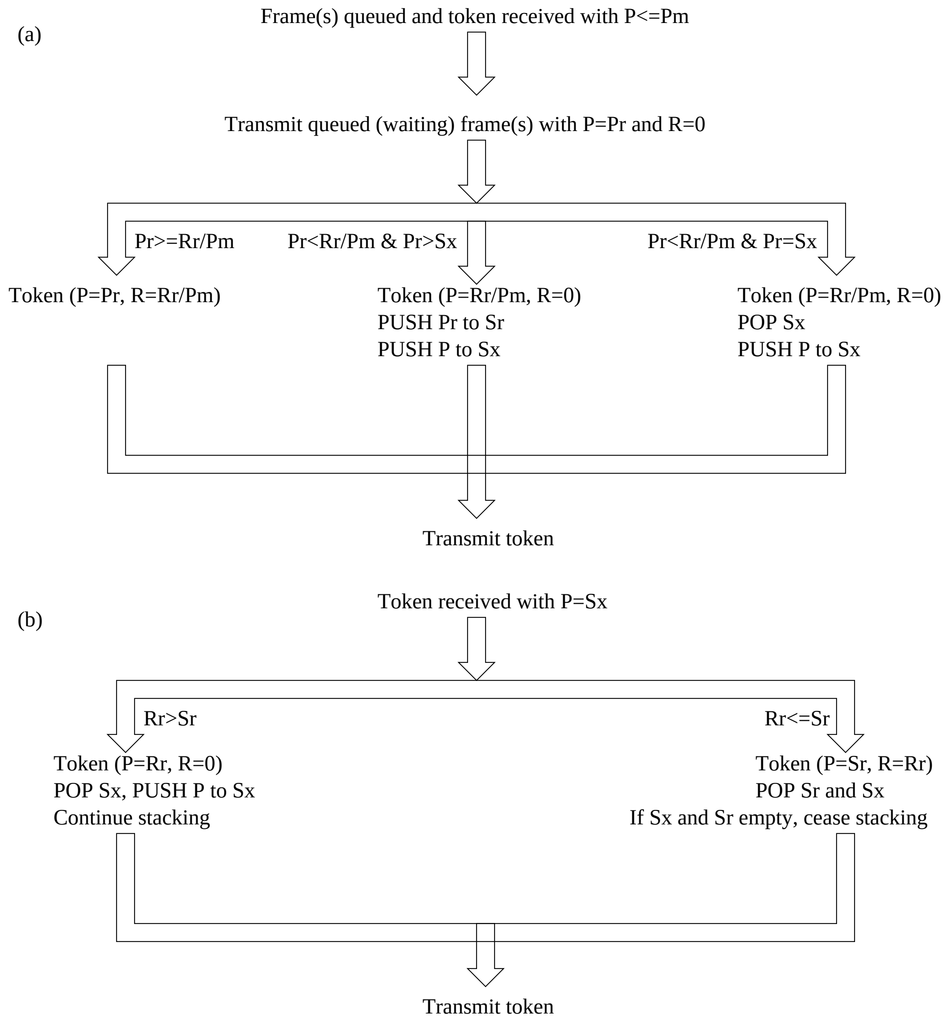


Figure 49: Token generation and stack modifications: (a) token generation [Note: $S_x = 0$ if stack empty]; (b) stack modification (Fig.6.16, p.301, 1st book)

(a) Four priority classes: 0, 2, 4, 6 (the book said 8, not 6).

(b) Four stations, 1, 7, 15, 17.

- Station 1: 1 frame of priority 2
- Station 7: 1 frame of priority 2
- Station 15: 1 frame of priority 4
- Station 17: 1 frame of priority 4

The stations sit on the ring in the increasing order of their station numbers.

(c) Summary of the example:

Rotation 0 No station has frames to transmit. The token is being circulated with both R bits and P bits equal to 0.

Rotation 1 Station 1 obtains the token since $P_r = 0, P_m = 2,\ P \leq P_m$.

i. It transmits a frame with $F(0,0)$. After that, its has $P = P_r = 0; R = 0$.

ii. Station 7 sees the data frame $F(0,0)$ from Station 1. As it has a frame with $P_m = 2 > R_r = 0$, it alters the R bits of the frame from 0 to 2 and passes the altered data frame $F(0,2)$. This indicates that Station 7 made a reservation to use the token to transmit frames with priority 2. For Station 7, it now has: $P = P_r = 0, R = P_m = 2$.

iii. Station 15 sees the altered data frame from Station 1. As it has a frame with $P_m = 4 > R_r = 2$, it *overrides* the R bits of the frame from 2 to 4 and passes the changed data frame $F(0,4)$. This indicates that Station 15 made a reservation to use the token to transmit frames with priority 4. For Station 15, it now has: $P = P_r = 0, R = P_m = 4$.

iv. Station 17 sees the altered data frame $F(0,4)$ from Station 1. It also has a frame with $P_m = 4$. However, since the value of the R bits in the frame is equal to 4, it *cannot* make another reservation. It has to just pass the frame. However, it updates its register values. For Station 17, it now has: $P = P_r = 0, R = R_r = 4, P_m = 4$.

Rotation 2 Station 1 sees its data frame coming back. The P bits still are unchanged at 0, but the R bits have been changed from 0 to 4. As Station 1 finishes transmitting frames, it releases a new token with $T(4,0)$. This is the middle case in Fig.6.16(a). Station 1 becomes a stacking station. It pushes old $P_r = 0$ onto S_r and new $P_r = 4$ onto S_x.

i. Station 7 sees the new token. Because $P = 4$ in the token, Station 7 cannot grab the token. However, since $R = 0$, it can make a reservation. It passes the token by revising the R bits to 2 from 0. It now has $P = P_r = 4, R_r = R = 2, P_m = 2$.

ii. Station 15 sees the revised new token $T(4,2)$. As its $P_m = 4$ and $P = 4$, it takes the token and transmits a data frame $F(4,0)$. It revises its registers as $P = P_r = 4, R = 0$.

iii. Station 17 sees the data frame. As $P_r = 4, R_r = 0, P_m = 4$, it makes a reservation by changing the R bits from 0 to 4 and passes the changed frame $F(4,4)$. It now has $P = P_r = 4, R = P_m = 4$.

Rotation 3 Station 1 sees the data frame from Station 15. It has no frame to transmit. As $P = 4 > S_r = 0$, it cannot do anything on its stacks. It simply passes the frame $F(4,4)$ and modifies its registers: R_r changes from 0 to 4.

i. Station 7 sees the frame $F(4,4)$ from Station 15. As $P_m = 2 < R_r = 4$, it cannot make a reservation. It simply passes the frame and modifies its registers: R_r changes from 2 to 4.

ii. Station 15 sees its data frame $F(4,4)$ coming back. It releases token $T(4,4)$, as $P_r = 4, R_r = 4, P_m = 0$ (it has no frames to send), this is the left case in Fig.6.16(a). It modifies its registers: R_r changes from 0 to 4.

ii. Station 17 sees the new token $T(4,4)$. As it has $P_m = 4$ and $P = 4$, it takes the token. It sends a frame $F(4,0)$ and modifies its registers: R_r changes from 4 to 0.

Rotation 4 Station 1 sees the new data frame $F(4,0)$. Because $P_r = 4 > S_r = 0$, it cannot change its stacks. It passes the token and revises its registers: R_r changes to 0 from 4.

i. Station 7 sees the frame $F(4,0)$. It makes another (second) reservation. It passes the revised frame $F(4,2)$ and changes its registers: $R_r = 4 \Rightarrow R_r = 2$.

ii. Station 15 simply passes the frame $F(4,2)$. It also modifies its registers: $R_r = 4 \Rightarrow R_r = 2$.

iii. Station 17 sees its data frame coming back. It now has no frames to transmit. However, as $P_r = 4, R_r = 2$, it simply releases a token $T(4,2)$. Notice that it cannot release a token $T(2,0)$. This is the left case in Fig.6.16(a). It then revises it registers: $R_r = 0 \Rightarrow R_r = 2$.

Rotation 5 Station 1 sees the new token $T(4,2)$. As $P_r = S_x$ ($P_r = 4, S_x = 4$) and $R_r > S_r$, ($R_r = 2, S_r = 0$), this meets the left case of Fig.6.16(b). It generates a new token $T(2,0)$ and pushes $P = 2$ onto S_x. Its registers are modified: $P_r = 4 \Rightarrow P_r = 2$ (R_r unchanged at 0). Intuitively, Station 1 now finds out that the ring service priority 4 it previously raised has to be lowed to 2. But it cannot low it to the previous ring service priority 0 yet.

i. Station 7 sees the token $T(2,0)$. As $P_m = 2$ and $P = 2$, it finally can take the token. It transmits a data frame $F(2,0)$ and modifies its registers: $P_r = 4 \Rightarrow P_r = 2$, $R_r = 2 \Rightarrow R_r = 0$.

ii. Both Station 15 and Station 17 simply pass the frame as they do not have any data frames to send. They also modify their register values accordingly.

Rotation 6 Station 1 simply passes the data frame from Station 7. Its

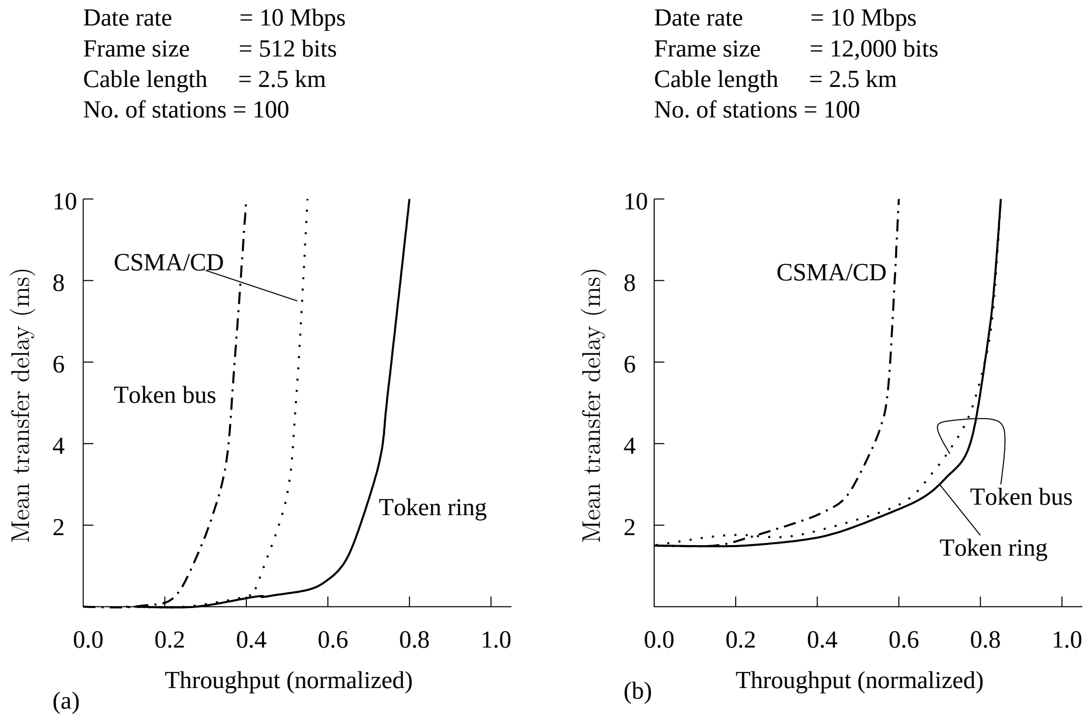


Figure 50: LAN performance comparisons: (a) 512 bit frames; (b) 12,000 bit frames (b) stack modification (Fig.6.16, p.301, 1st book)

register values stay the same.

i. Station 7 sees its data frame $F(2,0)$ coming back. It then releases the token $T(2,0)$. Its registers values stay the same.

ii. Again both Station 15 and Station 17 simply pass the frame as they do not have any data to send.

Rotation 7 Station 1 sees the token $T(2,0)$. As $P_r = S_x$ ($P_r = 2, S_x = 2$) and $R_r \leq S_r$, ($R_r = 0, S_r = 0$), this meets the right case of Fig.6.16(b). It generates a new token $T(0,0)$ and pops both S_r and S_x. This empties both stacks. The station ceases to be a stacking station. Its registers are modified: $P_r = 2 \Rightarrow P_r = 0$ (R_r unchanged at 0).

6. Comparisons of the Three LANs (Fig.6.23, p.316)

(1) CSMA/CD

a. Advantages
- Widely used
- Stations can be freely added without shutting down the network
- The algorithm is simple

- Only a cable, no modem, is required
- Delay at low load is zero

b. Weak points

- Each station must be able to detect the signal of the weakest station for the purpose of CD
- Minimum frame size is 64 bytes, resulting in overhead when only a one byte character is transmitted from one station to another
- As the data rate increases, the efficiency drops (recall channel efficiency = $1/(1 + 2B\tau/AF)$). So it is not a good candidate for high speed LANs.
- No priority scheme, hence not suitable for real-time control.
- Collisions at high load can seriously affect the performance.

(2) Token Bus

a. Advantages

- More robust if the token is present
- Short frames are allowed
- Priority schemes provides more freedom to stations
- At high load, the efficiency is excellent (becoming TDM)
- The broadband cable can be used to support multiple channels, such as TV.

b. Weak points

- Broadband systems use modem and amplifiers
- The protocol is complex
- The loss of the token at a critical moment may cause severe damage
- Substantial delay at low load

(3) Token Ring

a. Advantages

- Point-to-point connection can use full digital engineering
- Rings can use any transmission media
- Use of wire centers allows it to detect and eliminate cable failure automatically
- Priority schemes are provided, although not so fair
- No theoretic limitation on frame size
- At high load, the efficiency is excellent (becoming TDM)

b. Weak points

- Centralized monitor control. Monitor can not impeached
- The loss of the token at a critical moment may cause severe damage
- At low load, substantial delay