Chapter 12 Token Bus

Introduction

- LANs have a direct application in factory automation and process control, where nodes are computers controlling the manufacturing process.
- In this type of application, processing must occur in real-time with minimum delay, and at the same speed as the objects moving along the assembly line.

Introduction

- Ethernet is not suitable for this purpose, because the number of collisions is not predictable and delay in sending data from the control center to the computers along the assembly line is not a fixed value.
- Token Ring is not suitable either, because an assembly line resembles a bus topology and not a ring topology.

Introduction

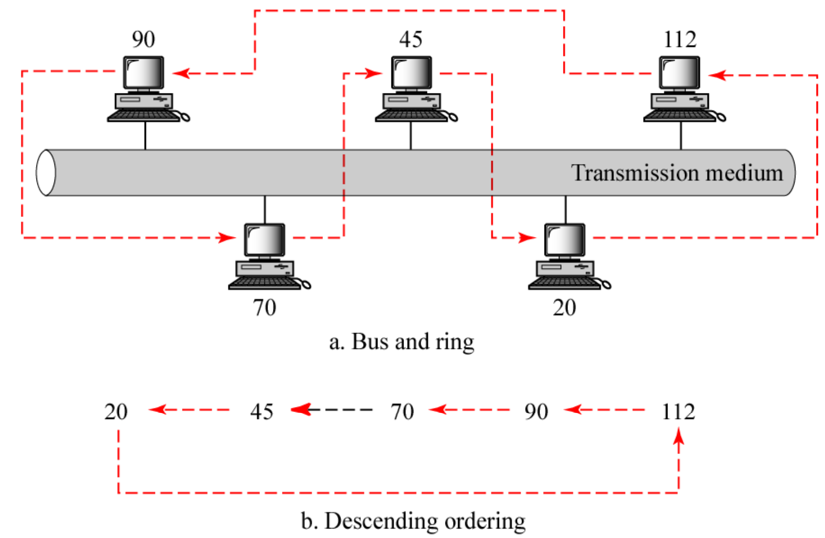
- **Token Bus** is the solution to this type of problem.
- It combines the physical configuration of Ethernet (a bus topology) and the collision-free (predictable delay) feature of Token Ring.
- It is a physical bus operating as a logical ring using tokens.

Physical Versus Logical Topology

- The logical ring is formed based on the physical address of the stations in **descending order**.
- Each station considers the station with the immediate lower address as the **next station** and the station with the immediate higher address as the **previous station**.
- The station with the lowest address considers the station with the highest address the next station, and the station with the highest address considers the station with the lowest address the previous station.

Figure 12-1

A Token Bus Network



Token Passing

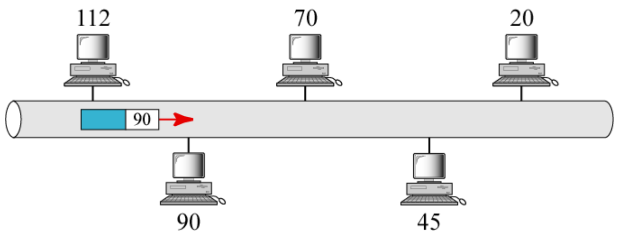
- To control access to the shared medium, a small token frame circulates from station to station in the logical ring.
- If a station has data frame to send, it keeps the token and sends its data frame.
- To prevent monopolization of the medium by any one station, each station can only keep the token for a specified period of time.

Token Passing

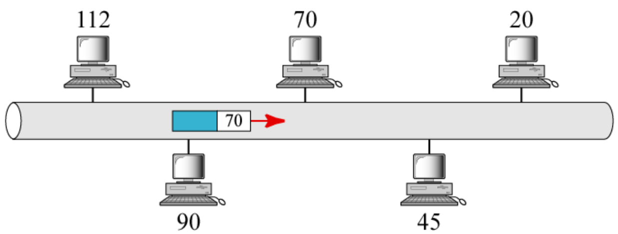
- After sending all its frames or after the time period expires (whichever comes first), the station releases the token to the next station.
- The token frame contains the address of the next station in the logical ring, because Token Bus is physically a bus and the token is received by all stations.
- Only the next station in the logical ring has the right to keep the token and sends its data frames.

Figure 12-2 a and b

Token Passing in a Token Bus Network



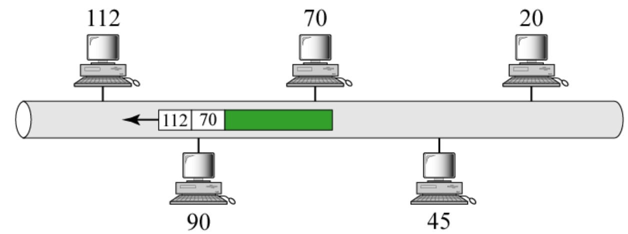
a. Station 112 does not have data; it sends the token to 90



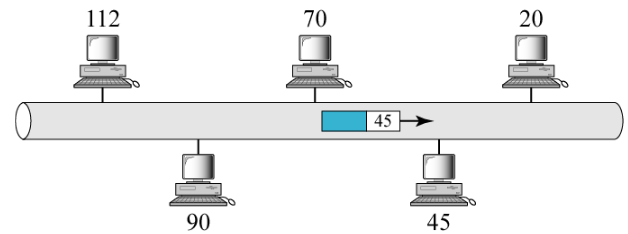
b. Station 90 does not have data; it sends the token to 70

Figure 12-2 c and d

Token Passing in a Token Bus Network



c. Station 70 sends a data frame to station 112



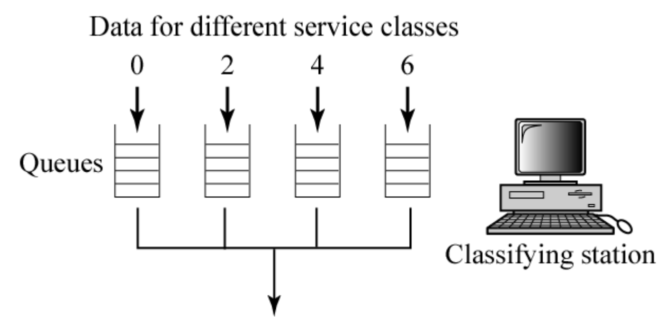
d. Station 70 sends the token to station 45

Service Classes

- In Token Bus all stations have the same priority.
- But each station can categorize its data into four classes: 0, 2, 4, and 6. Class 0 has the lowest priority and class 6 has the highest.
- If a station does not classify its data, all data belong to class 6.
- If a station wants to classify its data, it should have four queues, one for each class.

Figure 12-3

Queues for Service Classes



Timers

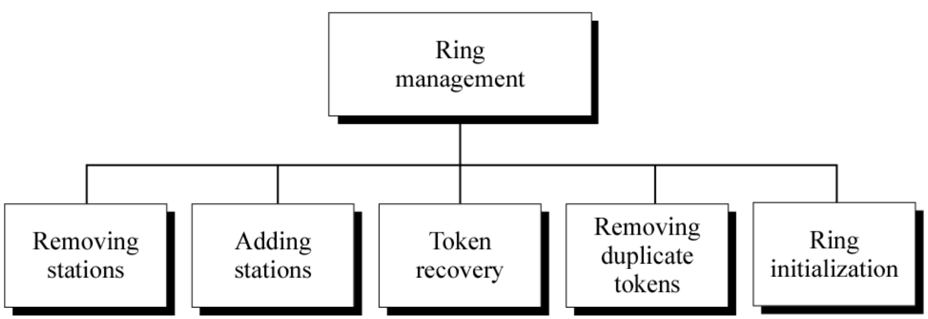
- To handle different data classes, Token Bus defines four timers: THT, TRT4, TRT2 and TRT0.
- The **token holding timer (THT)** is set to the maximum time that a station can send class 6 data. The setting of this timer is the same for all stations.
- The **token rotation timers (TRT4, TRT2,** and **TRT0)** are set to the maximum token circulation tome plus the time to send either class 4, 2, or 0 data. E.g. if TRT4 = 20 μs and it takes 18 μs for a token to come back, the station has 2 μs to send class-4 data if any.

Ring Management

- The whole Token Bus protocol depends on the maintenance of the logical ring and the controlling of the token.
- The logical ring must be maintained when
  - a station leaves the ring;
  - a station joins the ring;
  - a token is lost;
  - more than one token is created;
  - the ring is initialized at startup.

Figure 12-4

Ring Management



Different Types Of Procedures Used In Ring Management

Ring Management

- For management purpose, seven special frames are used:
  1. *Token*
  2. *claim-token*
  3. *set-successor*
  4. *solicit-successor-1*
  5. *solicit-successor-2*
  6. *resolve-contention*
  7. *who-follows*

Removing Stations

- When a station leaves the ring, its predecessor should be informed to bypass that station and reform the logical ring.
- Assume 5 stations (A $\rightarrow$ B $\rightarrow$ C $\rightarrow$ D $\rightarrow$ E $\rightarrow$ A) are in a Token Bus. If C leaves the ring, its predecessor (B) should know how to logically rebuild the ring (A $\rightarrow$ B $\rightarrow$ D $\rightarrow$ E $\rightarrow$ A).
- A station may leave the ring either voluntarily or unexpectedly (due to a malfunction).

Voluntary Leaving

- Assume 5 stations (A $\rightarrow$ B $\rightarrow$ C $\rightarrow$ D $\rightarrow$ E $\rightarrow$ A) are in a Token Bus. If station C leaves the ring **voluntarily**, the procedure is as follows:
  1. Station C waits until it receives a token from its predecessor (station B)
  2. Station C sends a *set-successor* frame to station B identifying station D as the successor to station B
  3. B updates its record to set its successor to D
  4. Station C sends the token to station D
  5. Station C leaves the ring

Unexpected Leaving

- Assume 5 stations (A $\rightarrow$ B $\rightarrow$ C $\rightarrow$ D $\rightarrow$ E $\rightarrow$ A) are in a Token Bus. If station C leaves the ring **unexpectedly** (due to a malfunction), station C cannot inform station B of the problem, and station B needs to find out by itself.
- When station B sends a token to station C, it always follows a procedure to monitor the bus activity and to make sure station C is functioning properly (see next slide).

Unexpected Leaving

Steps to follow after station B sends a token to station C:

1) Station B listens for activity on the bus. If station C is functioning, B will sense activity on the bus, and the following steps do not occur.
2) If B does not sense any activity on the bus, it waits a predetermined amount of time (response window time). If there is no news, B sends the token again to be sure that there is no transmission problem.

Unexpected Leaving

3) If B does not hear anything during the second response window time, it assumes C is not working. B sends a *who-follows* control frame containing the address of C, trying to find the next station after C.
4) Station D responds with a *set-successor* frame and specifies itself as the successor of station B.
5) Station B now changes its records and defines D to be its successor.
6) Station B sends a token to station D.

Adding Stations (Case 1)

- Each station in the ring must provide an opportunity periodically for other stations to join the ring.
- Case 1: If the joining station's address is within the range of the addresses of the stations already in the ring, the following procedure is followed:
  1) A token holding station sends a *solicit-successor-1* frame, containing the address of the sender and its successor, as an invitation to stations that want to join the ring.

Adding Stations (Case 1)

2) The token holding station waits a response window time. A station whose address is between the addresses of the sender and its successor can respond with a *set-successor* frame.
3) If the inviting station receives no response during the response window time, it closes the window and sends the token to its successor in the ring.
4) If the inviting station receives a response, it changes its successor to the responding station and sends the token to it.

Adding Stations (Case 1)

The added station also sets its predecessor (inviting station) and successor (previous successor of the inviting station) and becomes part of the ring.

5) If there is more than one response, the inviting station sends a *resolve-contention* frame and waits for four response window times.
6) Each station who has sent a *set-successor* frame earlier responds to the *resolve-contention* frame in one of the response windows according to the first two bits of its address (see Table 12-1).

Table 12-1

Response to Resolution-Contention Frame

First two bits	Responding window	Waiting time
00	1	None
01	2	One response window time
10	3	Two response window time
11	4	Three response window time

Adding Stations (Case 1)

7) If two or more stations respond in the same response window, collision will occur again. The inviting station then sends another *resolve-contention* frame. This time only stations involved in the second collision may respond based on their second two address bits.
8) Step 7 will be repeated if there are more collisions. Since no two stations have identical address bits, finally one station will be added to the ring.

Adding Stations (Case 2)

- Case 2: If a station has an address outside the range of the addresses of the stations already in the ring, it must wait for an invitation from the station with the lowest address. The following steps occur:
- When the token holding station has the lowest address in the ring, it sends *solicit-successor-2* frame, defining its own address and the largest address in the ring. A station with an address smaller than the inviting station or larger than the largest address in the ring can respond to the invitation.
- The remaining steps are the same as case 1.

Token Recovery

- A token may get lost due to:
  - The ring is set up for the first time;
  - The token holding station fails and takes the token down with itself;
  - The token is corrupted and discarded by the receiving station.
- Token-recovery procedure is used to generate a new token (see next slide).

Token Recovery

1) Every station has a token-recovery timer. When the timer expires and the station doesn't sense any activity on the line, it knows the token is lost.
2) The station sends a *claim-token* frame to indicate that it wants to generate a new token. Multiple stations may want to be the token claimant. To resolve the contention, each station pads the data field of the *claim-token* frame with an amount determined by the first two bits of its address (see Table 12-2).

Table 12-2

Claim-Token Frame Padding

First two bits	Length
00	$0 \times$ time slot
01	$2 \times$ time slot
10	$4 \times$ time slot
11	$6 \times$ time slot

Token Recovery

3) Each station that has sent a claim-token frame listens to the bus. If it senses a frame longer than the frame it has sent, it gives up the claim. The station that has sent the longest frame is responsible for generating a new token.
4) If two or more stations send frames of the same length, the contention continues this time with the second two bits.
5) Finally one station will emerge as the token claimant, since each address is unique in the LAN.

Removing Duplicate Tokens

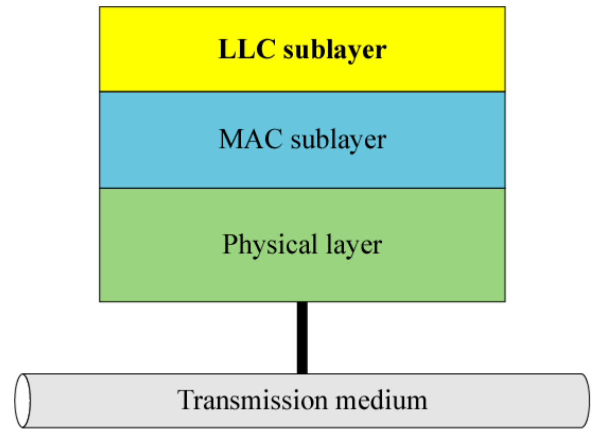
- If a token holding station detects some activity on the bus, it knows that some other station is also holding a token.
- The station that has detected the activity on the bus drops the token and moves to the listening state.

Ring Initialization

- At the initial startup, each station knows only its own address, and there is no token and no logical ring.
- The ring initialization procedure makes use of two previously discussed procedures to create the ring:
  - When the stations start up, each sends a *token-recovery* frame to contend for access to the token (token recovery).
  - The winning station uses *solict-successor-2* frame to start adding the next station to the ring (adding station/case 2).
  - After the successor is found, the token is passed to that station and the process continues until all stations are added to the ring.

Figure 12-5

Token Bus Layers Defined by IEEE 802.4



MAC Sublayer

- **Access Method** – uses **token passing** access method over a physical bus topology (see ring management discussion before).
- **Frame Format** – contains eight fields; slightly different from Ethernet frame which has seven fields.

SD Start delimiter (flag)
FC Frame control (frame type)
ED End delimiter (flag)
FCS Frame Check Sequence

Preamble	SD	FC	Destination address	Source address	Data	FCS	ED
1 or more bytes	1 byte	1 byte	2 or 6 bytes	2 or 6 bytes	0 to 8190 bytes	4 bytes	1 byte

Figure 12-6: Token Bus Frame

Frame Format

- **Preamble** – a predefined pattern to synchronize the sender and receiver; added by the physical layer and not part of MAC frame
- **Start delimiter** – used to alert the receiving station to the arrival of a frame; made of 0 or *N* bits where *N* is not the regular encoding for binary 1, such that an SD bit pattern will not be confused with any regular data pattern (Fig. 12-7); SD is added by physical layer.
- **Frame control** – defines the type of frame (Fig. 12-8).

Figure 12-7

SD Field

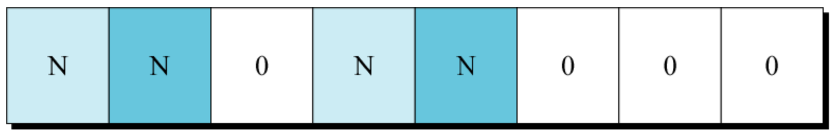
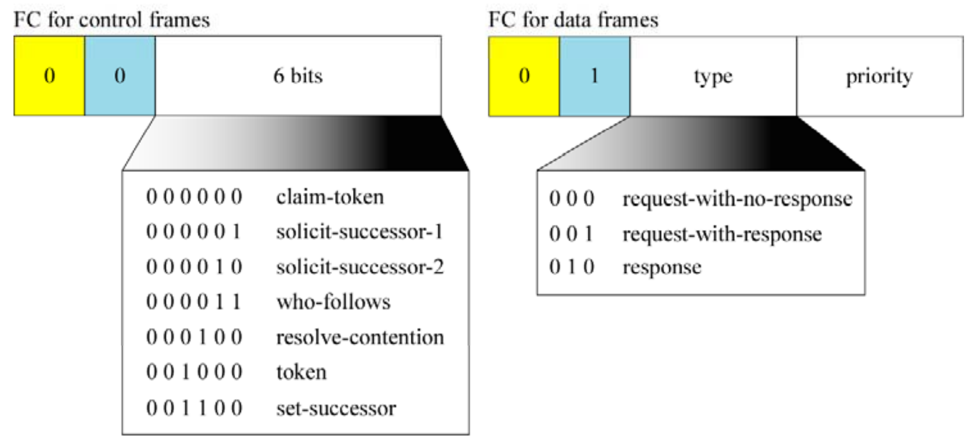


Figure 12-8

FC Field Format



Frame Format

- **Destination address** – contains the physical address of the frame's next destination; may or may not be valid base on the type of frame (see Table 12.3).
- **Source address** – contains the physical address of the sending station; address is present in all frame types.
- **Data** – contains data from the LLC layer for a data frame or extra information for some control frames based on the type of frame (see Table 12.4).
- **FCS** – CRC-32 error detection sequence.

Table 12-3

Destination Address for Each Frame Type

Frame	Destination Address
claim-token	ignored
solicit-successor-1	address of successor
solicit-successor-2	address of successor
who-follows	ignored
resolve-contention	ignored
token	address of successor
set-successor	address of station that has sent solicit-successor frame
data	**address of recipient of data**

Table 12-4

Data Field Contents

Frame	Data
claim-token	length of frame
solicit-successor-1	missing
solicit-successor-2	missing
who-follows	missing
resolve-contention	missing
token	missing
set-successor	new address of successor
data	**data from LLC level**

Frame Format

- **End delimiter** – used to alert the receiving station to the termination of a frame; made of 1 or *N* (nondata) bits where *N* is not the regular encoding for binary 0, such that an ED bit pattern will not be confused with any regular data pattern (see Fig. 12-9); ED is added by physical layer.

Figure 12-9

ED Field Format

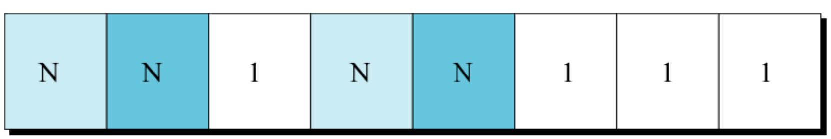
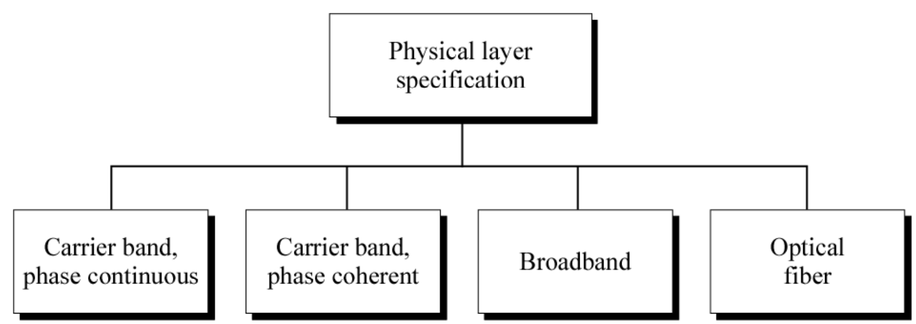


Figure 12-10

Physical Layer Specification



IEEE 802.4 defines four specifications for the physical layer of Token Bus.

Carrier Band, Phase Continuous

- Based on carrier band modulation – analog modulation using the whole capacity of the medium without multiplexing.
- Encoding and Signaling – data is first encoded using Manchester encoding and then modulated using FSK modulation (Fig. 12-11)
  - Bit 0: frequency range is from high (H) to low (L)
  - Bit 1: frequency range is from low (L) to high (H)
- The *N* bit is encoded in pair (*NN*) as LLHH.

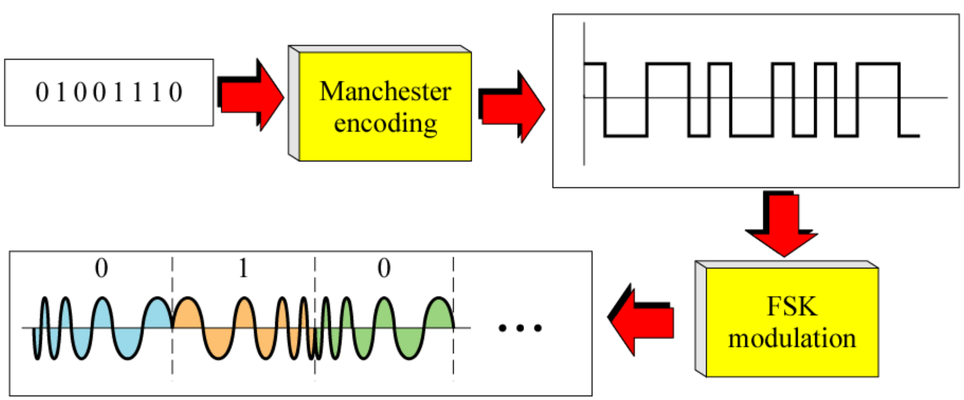
Table 12-5

Parameters for Carrier Band, Phase Continuous

Parameter	Value
Encoding	Manchester
Modulation	FSK (3.75 and 6.25 MHz)
Data rate	1 Mbps
Topology	Bidirectional bus (two buses)
Medium	Thin Coaxial Cable (75 ohm)
Medium Interface	T connector and drop cables up to 35 cm in length

Figure 12-11

Encoding and Signaling for Carrier Band, Phase Continuous



Carrier Band, Phase Coherent

- Also based on carrier band modulation – analog modulation using the whole capacity of the medium without multiplexing.
- Encoding and Signaling – data is first encoded using Manchester encoding and then modulated using FSK modulation (see Fig. 12-12)
  - Bit 1: modulated by low (L) frequency
  - Bit 0: modulated by high (H) frequency
- *NN* pair is encoded as:

  ½ period of H + 1 period of L + ½ period of H

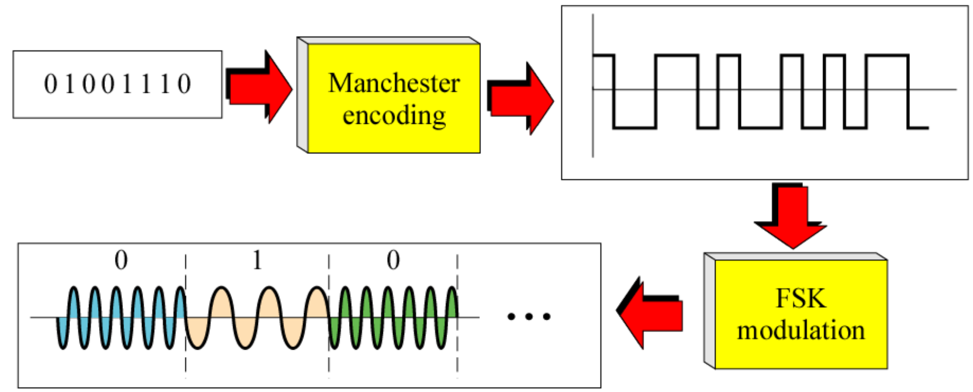
Table 12-6

Parameters for Carrier Band, Phase Coherent

Parameter	Value
Encoding	Manchester
Modulation	FSK (5/10 and 10/20 MHz)
Data rate	5 and 10 Mbps
Topology	Bidirectional bus (two buses)
Medium	Semi-rigid Coaxial Cable (75 ohm)
Medium Interface	Taps

Figure 12-12

Encoding and Signaling for Carrier Band, Phase Coherent



Broadband

- Defines an analog transmission with multiplexing
- Encoding and Signaling – data is first encoded using multilevel duobinary encoding and then modulated using a combination of ASK/PSK encoding (Fig. 12-13)
  - Three levels of encoding used: 0, 2, and 4
  - Bit 0 is encoded as signal value 0
  - Bit 1 is encoded as signal value 4
  - Signal value 2 is for nondata bits (such as the *N* bits)
- Uses one bus with a frequency converter or two buses.

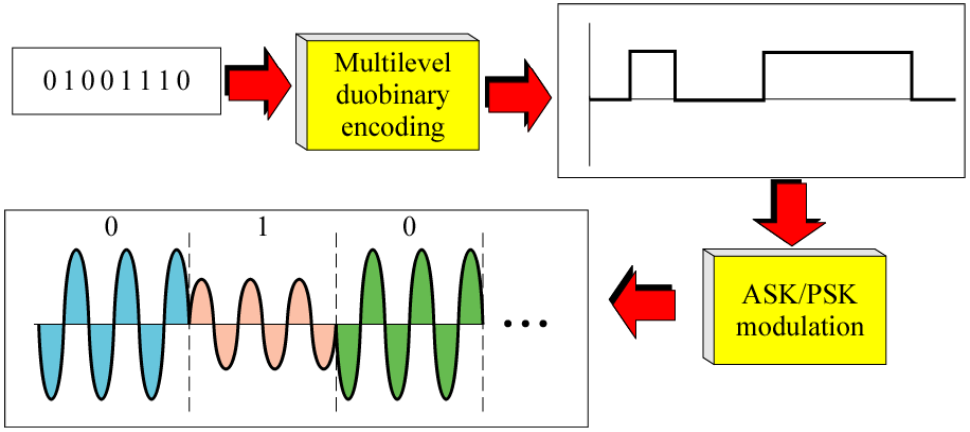
Table 12-7

Parameters for Broadband

Parameter	Value
Encoding	Multivalue duobinary
Modulation	ASK/PSK
Data rate	1, 5, and 10 Mbps
Bandwidth	1.5 MHz, 5 MHz, 12 MHz
Topology	Unidirectional bus (one bus with a frequency converter)
Medium	Semi-rigid Coaxial Cable (75 ohm)
Medium Interface	Taps

Figure 12-13

Encoding and Signaling for Broadband



Optical Fiber

- Defines a broadband transmission with multiplexing
- Encoding and Signaling – data is first encoded using Manchester encoding and then modulated using ASK.
  - Bit 1 is represented as high amplitude
  - Bit 0 is represented as zero amplitude
- The *N* bit is represented as a third amplitude level.

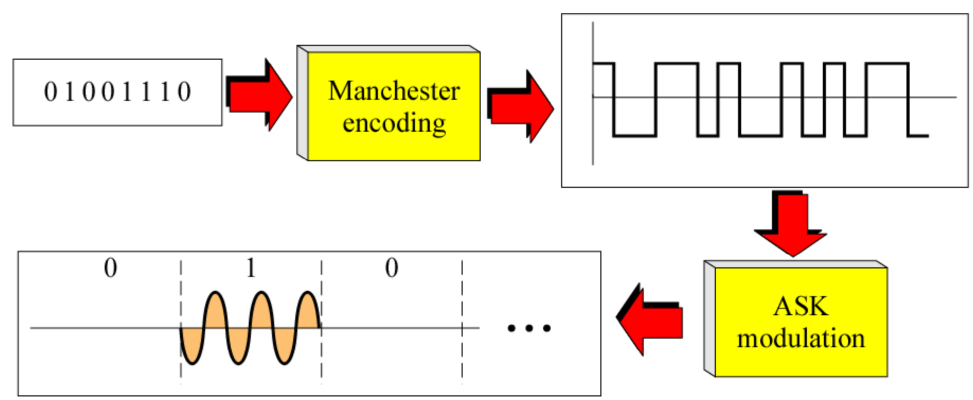
Table 12-7

Parameters for Broadband

Parameter	Value
Encoding	Manchester
Modulation	ASK (on/off)
Data rate	5, 10, 20 Mbps
Bandwidth	10 MHz, 20 MHz, 40 MHz
Topology	Star
Medium	Fiber-optic cable

Figure 12-14

Encoding and Signaling for Optical Fiber



Summary

- Token Bus combines the physical configuration of Ethernet (a bus topology) with the collision-free (predictable delay) feature of Token Ring.
- Manufacturing processes requiring automation and process control can use a Token Bus network.
- The token circulates from physical address to physical address around the ring in the descending order.
- Each station can prioritize its data into four classes: 0, 2, 4, and 6. Timers control the classes.

Summary

- The ring is managed through the use of seven special frames. These frames handle station removal, station addition, token recovery, duplicate token removal, and ring initialization.
- The MAC sublayer is responsible for the token passing operations.
- The IEEE 802.4 defines four specifications for the physical layer: carrier band with continuous phase, carrier band with coherent phase, broadband, and optic fiber.