0. PRECURSORS

– Earlier schemes from which CSMA/CD evolved

• ALOHANET
  – First packet radio network developed at U. of Hawaii to connect scattered terminals on several islands to communicate with the university computer.
  – Two independent channels used:
    • inbound terminals → central node; channel at 407.35 MHz
    • outbound central node → terminals;          at 413.48 MHz
    • Data rate of 9600 bps
  – Repeaters used to increase range
  – Fixed routing is used with repeaters
  – Inbound channel use ALOAH contention mechanism for channel access
    • Any terminal/repeater with a packet immediately transmits it.
    • Source of packet waits for timeout period for ACK for the packet from central node (0.2 sec).
    • If no ACK is received, packet assumed to have “collided” with some other; retransmitted after random interval (uniform distribution: 0.2 ~ 1.5 sec.)
  – Outbound channel is not contended since only central node originates transmission.
• **ALOHA**
  - Random access (or contention) techniques used for shared-channel access
  - Was developed for use with packet radio networks, but forms basis for most contention-based shared-medium access techniques.
  - Any node with a newly generated packet:
    - 1. Immediately transmits the packet
    - 2. Waits for a round-trip interval for ACK for packet
    - 3. If ACK is not received, waits for a random timeout interval and retries (step 1.)
  - Also called pure-ALOHA: “Talk when you please”

- **Packet ready?**
  - No
  - Yes
  - **Transmit**
  - **Wait round-trip propagation delay**
  - **Positive ACK?**
    - No
    - Yes
  - **Delay k packet transmission times**
  - **Compute random backoff integer(k)**
• Analysis
  • S: throughput of network (rate of successfully received packets; normalized to the network capacity) (carried load)
  • G: offered load (rate of data presented to the network for transmission; in case of collision, count both)
  • I: Input load (rate at which new data is being generated by all stations combined)
  • D: average packet delay (time from generation to successful receipt at destination)

  – Assumptions
    • All packets have constant length (normalized w.r.t. packet-transmission time: “t” = 1)
    • Channel is noise-free
    • Throughput = input load (S=I): packets do not pile up at stations
    • G is Poisson distributed:
      \[ \Pr[k \text{ packets in time } "t" ] = \frac{G^k e^{-G}}{k!} \]

  – For successful transmission:

\[ \Rightarrow \text{Vulnerable period } = 2t \]
• \( \Pr[\text{successful transmission of a packet}] = \Pr[0 \text{ other packets are generated in } t1 \text{ or } t2] = \Pr[\text{No packets in } t1] \cdot \Pr[\text{No packets in } t2] = \frac{G^0 e^{-G}}{0!} \cdot \frac{G^0 e^{-G}}{0!} = e^{-G} \cdot e^{-G} = e^{-2G} \)

• \( \Pr[\text{successful transmission}] = \frac{\text{Rate of successful packet transmissions}}{\text{Rate of attempted packet transmissions}} = \frac{S}{G} \)

• So \( S/G = e^{-2G} \Rightarrow S = G e^{-2G} \)

Since \( dS/dG = e^{-2G}[1 - 2G] = 0 \rightarrow G = 1/2 \)

Maximum throughput = \( \frac{1}{2e} = 18\% \)

– Average Delay

\( = (\text{Expected # of retransmissions}) \cdot (\text{Delay per retransmission}) + \text{Delay for last (successful) transmission} \)

• Expected # of transmissions per packet

\( = \frac{\text{Rate of attempted transmissions}}{\text{Rate of successful transmissions}} = \frac{G}{S} \)

• Expected # of retransmissions per packet = \( \frac{G}{S} - 1 \)

• Retransmission algorithm: wait for random time between 1 and \( k \) packet transmission times (uniformly distributed)

• Delay per retransmission = \( 1 + 2a + w + (k+1)/2 \)
  – \( 1 + 2a \) : transmission time + 2 propagation times
  – \( w \) : time at receiver to generate ACK
  – \( (k+1)/2 \) : average timeout delay
\[
\begin{align*}
\therefore D &= (e^{2G} - 1) \left(1 + 2a + w + \frac{k+1}{2}\right) + a + 1 \\
\text{Expected # of retrans.} & \quad \text{Expected delay per retrans.} & \quad \text{Time for final successful trans.}
\end{align*}
\]

- If packet propagation time is not negligible, modification needed; Now vulnerable period = \(2(1+a)t\)

- Pr[k packets in time \((1+a)t\)]

\[
\frac{[(1+a)G]^k e^{-(1+a)G}}{k!} \quad \text{for } k = 0 \rightarrow e^{-(1+a)G}
\]

\[
\therefore S = \frac{G}{e} e^{-(1+a)G} e^{-(1+a)G} = e^{-2(1+a)G}
\]

\[
D = (e^{2(1+a)G} - 1)\left(1 + 2a + w + \frac{k+1}{2}\right) + a + 1
\]

- **Slotted ALOHA**
  - Improvements in throughput possible by dividing time into fixed slots
  - Transmission is only allowed at the beginning of slot; if packet is generated in between, node waits till next slot

  Packet created

  \[\text{Packet transmission begins}\]

  Collisions may occur only with packets created in this interval

  \[\therefore \text{Vulnerable period} = 1 \cdot t = (or \ (1+a)t \text{ if } a \text{ is not negligible})\]
Pr[successful transmission]  
\[ S = \frac{e^{-G}G^0}{0!} = e^{-G} \]  
\[ \therefore S = G e^{-G} \]  
\[ D = (e^G - 1)(1 + 2a + w + \frac{k+1}{2}) + 1.5a + 1.5 \]  
- Expected # of retrans.  
- Expected delay per retrans.  
- Extra 1/2 slot avg. waiting time

CSMA (Carrier Sense Multiple Access)  
- When \( a \ll 1 \) (Propagation time \( \ll \) Transmission time), improvements possible by “Listen Before Talk” discipline  
- Now collisions can only occur if two nodes decide to transmit within “\( a \)” seconds of each other, rather than \( 2(1+a) \) with pure-ALOHA (\( 1+a \) with S-ALOHA).  
- Three CSMA schemes  
  - 1. Nonpersistent CSMA  
  - 2. \( p \)-persistent CSMA  
  - 3. 1-persistent CSMA
– Nonpersistent CSMA
  • 1. If the medium is idle, transmit.
  • 2. If the medium is busy, wait an amount of time drawn from a probability distribution and repeat step 1.

– 1-persistent CSMA
  • 1. If the medium is idle, transmit.
  • 2. If the medium is busy, continue to listen until the channel is sensed idle, then transmit immediately.
  • 3. If there is a collision (determined by a lack of ACK), wait a random amount of time and repeat step 1.

– p-persistent CSMA
  • 1. If the medium is idle, transmit with probability p, and delay with probability (1-p). (The time unit is typically equal to the maximum propagation delay.)
  • 2. If the medium is busy, continue to listen until the channel is idle and repeat step 1.
  • 3. If transmission is delayed one time unit, repeat step 1.
CSMA

Packet ready? No
Yes

Carrier sense strategy

Wait round-trip propagation delay

Positive ACK? No
Yes

Transmit

Delay k packet transmission times

Compute random backoff integer(k)

Nonpersistent CSMA

Channel busy? Yes
No

A
B
C

p-persistent CSMA

Channel busy? Yes
No

Select random number r from [0,1]

Delay one time unit

r ≤ p?

No connection

A
B
C
1. ETHERNET AND FAST EATHERNET (CSMA/CD) IEEE802.3

- **CSMA/CD**: CSMA with Collision Detection
  - “Listen While Talk” scheme:
    - Listen before transmission till channel is free.
    - Additionally continue to monitor channel during transmission.
    - If collision is detected, immediately abort transmission.
  - Reduces “bandwidth waste” when collisions occur.
  - For baseband CSMA/CD, worst-case “wasted-time” due to a collision = $2 \cdot T_{\text{prop}}$
    $\Rightarrow$ Minimum packet length $\geq 2 \cdot T_{\text{prop}}$
    $\Rightarrow$ Packet length should be at least twice the propagation delay ($a \leq 0.5$)
- For broadband CSMA/CD, the maximum time to detect a collision is four times the propagation delay from an end of the cable to the headend
- 1-persistent CSMA used; low delay at low loads
- To improve utilization at high loads, "binary exponential backoff" is used: doubles mean delay at each collision
CSMA/CD

Packet ready?  
Yes → A

Carrier sense strategy

Collision detected?  
Yes → B

Transmit

Delay k packet transmission times

Compute random backoff integer(k)

No → C

Abort transmission

Transmit jamming signal
• Performance analysis of CSMA/CD
  – Alternating periods of successful packet transmission and contention. Contention period can have “collisions” and “no-tries”.

  \[ S = \text{useful time fraction} = \frac{1}{1 + \text{Average contention time}} \]

  \[ \text{Average contention time} = (\text{Expected value of "\# of slots before successful transmission"}) \times 2a \]

  \[ \text{Pr}[i \text{ slots for successful transmission}] = \text{Pr}[i \text{ unsuccessful slots}] \times \text{Pr}[(i+1)\text{th slot is successful}] \]

  \[ \text{Pr}[\text{successful slot}] = \text{Pr}[\text{exactly one attempt in slots}] = \left( \binom{N}{i} \right) P^i (1 - P)^{N-1} \equiv \]

  \[ \text{Pr}[i \text{ slots for successful transmission}] = P_i = (1-A)^i \times A \]

  \[ \text{Average \# of contention slots} = \sum_{i=1}^{\infty} i P_i = \sum_{i=1}^{\infty} i(1-A)^i A = \frac{1-A}{A} \]

  \[ \therefore S = \frac{1}{1 + 2a(1 - A)/A} \]
\[ S = \frac{1}{1 + 2a(1 - A) / A} \]

- \( S \) is max when \( A \) is max
- \( A = NP(1 - P)^{N-1} \) is max when \( P = 1/N \)
- \( A_{\text{max}} = (1 - 1/N)^{N-1} \)
- \( \lim_{N \to \infty} A_{\text{max}} = \frac{1}{e} \Rightarrow \lim_{N \to \infty} S = \frac{1}{1 + 3.44a} \)
IEEE 802.3 MAC Frame Format

<table>
<thead>
<tr>
<th>Octets</th>
<th>Preamble</th>
<th>SFD</th>
<th>DA</th>
<th>SA</th>
<th>Length</th>
<th>LLC data</th>
<th>Pad</th>
<th>FCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>≥0</td>
<td>≥0</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

- **Preamble**: A 7-octet pattern of alternating 0s and 1s used by the receiver to establish bit synchronization (establishes the rate at which bit are sampled.)
- **Start frame delimiter (SFD)**: Special pattern 10101011 indicating the start of a frame.
- **Destination address (DA)**:
- **Source address (SA)**:
- **Length**: Length of the LLC data field
- **LLC data**:
- **Pad**: Octets added to ensure that the frame is long enough for proper CD operation.
- **FCS**: Error checking using 32-bit CRC.
IEEE 802.3 10-Mbps Specifications (Ethernet)

- Many alternative physical configurations
  - 10BASE5
  - 10BASE2
  - 10BASE-T (Twisted-pair)
  - 10BROAD36
  - 10BASE-F (optical Fiber)

<table>
<thead>
<tr>
<th>Transmission medium</th>
<th>Protocol</th>
<th>10BASE5</th>
<th>10BASE2</th>
<th>10BASE-T</th>
<th>10BROAD36</th>
<th>10BASE-FP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coaxial Cable (50 ohm)</td>
<td>Baseband (Manchester)</td>
<td>500</td>
<td>185</td>
<td>100</td>
<td>1800</td>
<td>500</td>
</tr>
<tr>
<td>Coaxial Cable (50 ohm)</td>
<td>Baseband (Manchester)</td>
<td>Bus</td>
<td>Bus</td>
<td>Star</td>
<td>Bus/Tree</td>
<td>Star</td>
</tr>
<tr>
<td>Unshielded twisted pair</td>
<td>Baseband (Manchester)</td>
<td>850 nm optical fiber pair</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coaxial Cable (75 ohm)</td>
<td>Broadband (DPSK)</td>
<td>Manchester/On-off</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Topology</td>
<td>Maximum segment length (m)</td>
<td>100</td>
<td>30</td>
<td>...</td>
<td>...</td>
<td>33</td>
</tr>
<tr>
<td>Nodes per segment</td>
<td>Cable diameter (mm)</td>
<td>10</td>
<td>5</td>
<td>0.4 - 0.6</td>
<td>10 - 25</td>
<td>62.5/125 µm</td>
</tr>
</tbody>
</table>
IEEE 802.3 100-Mbps Specifications (Fast Ethernet)

- Fast Ethernet: a set of specifications developed by IEEE 802.3 committee to provide a low-cost Ethernet-compatible LAN operating 100 Mbps.

---

**IEEE 802.3 (100-Mbps)**

- **100BASE-X**
  - 2 Category 5 UTP
  - 2 STP

- **100BASE-TX**
  - 2 Optical Fiber

- **100BASE-FX**
  - 4 Category 3 or Category 5 UTP

**IEEE 802.3 100BASE-T options**

---

**IEEE 802.3 100BASE-T physical layer medium alternatives**

<table>
<thead>
<tr>
<th></th>
<th>100BASE-TX</th>
<th>100BASE-FX</th>
<th>100BASE-T4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmission medium</td>
<td>2 pair, STP</td>
<td>2 pair, Category 5 UTP</td>
<td>4 pair, Category 3, 4, or 5 UTP</td>
</tr>
<tr>
<td>Signaling technique</td>
<td>4B5B, NRZI</td>
<td>MLT-3</td>
<td>4B5B, NRZI</td>
</tr>
<tr>
<td>Data rate</td>
<td>100 Mbps</td>
<td>100 Mbps</td>
<td>100 Mbps</td>
</tr>
<tr>
<td>Maximum segment length</td>
<td>100 m</td>
<td>100 m</td>
<td>100 m</td>
</tr>
<tr>
<td>Network span</td>
<td>200 m</td>
<td>200 m</td>
<td>400 m</td>
</tr>
</tbody>
</table>
2. TOKEN RING AND FDDI

- IEEE 802.5 Token Ring Medium Access Control
  - MAC Protocol
Token ring operation
- **MAC Frame Format**

<table>
<thead>
<tr>
<th>Octets</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>2 or 6</th>
<th>2 or 6</th>
<th>≥ 0</th>
<th>4</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD</td>
<td>AC</td>
<td>FC</td>
<td>DA</td>
<td>SA</td>
<td>Data</td>
<td>FCS</td>
<td>ED</td>
<td>FS</td>
<td></td>
</tr>
</tbody>
</table>

SD = Starting delimiter  
AC = Access control  
SA = Source address  
FC = Frame control  

**(a) General frame format**

<table>
<thead>
<tr>
<th>SD</th>
<th>AC</th>
<th>ED</th>
</tr>
</thead>
</table>

**(b) Token frame format**

| J | K | 1 | J | K | 1 | I | E |

J, K = Nondata bits  
I = Intermediate-frame bits  
E = Error-detected bits

| P | P | P | T | M | R | R | R |

PPP = Priority bits  
M = Monitor bit  
T = Token bit  
RRR = Reserv. bits

**(c) Access control field**

| A | C | x | x |

A = Addressed recognized bit  
C = Copied bit

**(d) Ending delimiter field**

| A | C | x | x |

**(e) Frame status field**

- **Starting delimiter (SD):** Indicates start of frame. JK0JK000, J and K are nondata symbols.

- **Access Control (AC):** PPPTMRRR, PPP and RRR are 3-bit priority and reservation variables, T is for indicating whether the frame is a token, M is for the monitor station. If T is 0, then the frame is a token, and the only remaining field is ED.

- **Frame control (FC):** FFFZZZZZZ, F: frame type bits and Z: control bits
• **Ending delimiter (ED):** JK1JK1IE, J and K are non data symbols, I is an intermediate frame bit. A communication between two stations may consists of many frames, and bit I is 0 in the last frame and 1 otherwise. E is an error bit, which is set to 1 whenever an error (such as an FCS) is detected.

• **Frame status (FS):** ACXXACXX, A: address recognized bit, C: Frame copied bit, and X: undefined bit.

– **General Operation:**

• If nothing to send, then continue regeneration and forwarding bits across the ring as they are passed through the repeater.

• If something to send, wait for Token to come across. Upon seeing the T bit in AC as 0, change to 1, and send the data.

• Stations between the sender and destination will pass the bits through their repeaters. The destination will detect its own DA and copy the frame in. Also change the A and C bits of the Frame status to 1.

• During or after transmission, the frame will have looped back. The sender can check the A and C bits for a form of ACK.
  
  AC: 00  Destination doesn’t exist  
  AC: 10  Destination exists, but is too busy to copy  
  AC: 11  Frame copied  

• Sender removes frames that it sent off the ring. In general, any bits it receives during transmission must be its own.

• After station is done sending, and after it starts receiving bits from its own transmission, it puts a new token on the ring.

• The P and R bits are used for a priority scheme.
Performance analysis of simple Token Ring

- N: # of stations on ring
- Assumption: Every station is always ready to transmit a packet
- Case 1: \( a < 1 \)
  \[
  T_1 = 1, \quad T_2 = \frac{a}{N} + 1, \quad \sum S = \frac{1}{1 + \frac{a}{N}}
  \]
- Case 2: \( a > 1 \)
  \[
  T_1 = 1, \quad T_2 = a + \frac{a}{N}, \quad \sum S = \frac{1}{a(1 + 1/N)}
  \]
Token ring priority scheme
- IEEE 802.5 Physical Layer Specification

<table>
<thead>
<tr>
<th>Transmission Medium</th>
<th>Shielded Twisted Pair</th>
<th>Unshielded Twisted Pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data rate (Mbps)</td>
<td>4 or 16</td>
<td>4</td>
</tr>
<tr>
<td>Signaling Technique</td>
<td>Differential Manchester</td>
<td>Differential Manchester</td>
</tr>
<tr>
<td>Maximum number of repeaters</td>
<td>250</td>
<td>72</td>
</tr>
<tr>
<td>Maximum length between repeaters</td>
<td>Not specified</td>
<td>Not specified</td>
</tr>
</tbody>
</table>

Table 13.4 IEEE 802.5 Physical Layer Medium Alternatives
FDDI (Fiber Distributed Data Interface) Medium
Access Control

- MAC Frame

<table>
<thead>
<tr>
<th>Bits</th>
<th>64</th>
<th>8</th>
<th>8</th>
<th>16 or 48</th>
<th>16 or 48</th>
<th>≥ 0</th>
<th>32</th>
<th>4</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preamble</td>
<td>SD</td>
<td>FC</td>
<td>DA</td>
<td>SA</td>
<td>Info</td>
<td>FCS</td>
<td>ED</td>
<td>FS</td>
<td></td>
</tr>
</tbody>
</table>

(a) General frame format

<table>
<thead>
<tr>
<th>Preamble</th>
<th>SD</th>
<th>FC</th>
<th>ED</th>
</tr>
</thead>
</table>

(a) Token frame format

SD = Start-frame delimiter
FC = Frame control
ED = Ending delimiter

SD = Start-frame delimiter
FC = Frame control
ED = Ending delimiter

- **Preamble:** For synchronization.
- **Starting delimiter (SD):** Indicates start of frame. JK, where J and K are nondata symbols (4 bits).
- **Frame control (FC):** Has the bit format CLFFZZZZ, where C indicates whether this is a synchronous or asynchronous frame; L indicates the use of 16- or 48-bit address; FF indicates whether this is an LLC, MAC control, or reserved frame. For a control frame, the remaining 4 bits indicate the type of control frame. For token frame, FC has the bit format 10000000 or 11000000 to indicate this is a token.
- **Ending delimiter (ED):** Contains a nondata symbol (T), and a pair of nondata symbols (T) for the token frame.
- **Frame status (FS):** Contains the error detected (E), address recognized (A), and frame copied (F) indicators. Each indicator is represented by a symbol, which is R for “reset” or “false” and S for “set” or “true.”
– FDDI MAC Protocol
  • Fundamentally similar to IEEE 802.5
  • Due to the high data rate (100 Mbps) and the longer distance segment than the 802.5, a frame on the FDDI ring may be significantly shorter than the bit-length of the ring.
  • In normal token ring, a station does not give up the token until the following:
    – Finished transmitting all its frame
    – Starts to receive leading edge of last frame transmitted
  • Waiting for the edge of the frame to come back waists potential capacity
  • In FDDI, token is sent immediately after last frame sent: “Fast (Early) token release”

  • In 802.5, station seizes the token by flipping the T bit of a passing token frame from 0 to 1, and then appends its own frame to it.
  • In FDDI, bits move too fast to be modified. Token seizure is done by aborting the rest of frame as soon as it is recognized as a token. Rest of token is read in. Next station will recognize the aborted frame and discard it.
Figure 13.9 Example of FDDI Token Ring Operation
Capacity Allocation

- The priority scheme used in 802.5 does not work in FDDI, as a station issues a token before its own transmitted frame returns.
- FDDI uses a capacity allocation scheme which seeks to accommodate a mixture of stream and bursty traffic.
- FDDI defines two types of traffic: synchronous and asynchronous.

- Assume fixed length frame sizes.
- Rather than defining “capacity” as bps, define it as the number of frames that can be transmitted in a given time period.
- Define the total amount of frames in a given time period as the “synchronous traffic”.
- Each station is allocated a certain percentage of the synchronous traffic. “Synchronous Allocation” or “SA”.
- Also define the rate at which the token needs to circulate around the ring as TTRT - “Target Token Rotation Time”.
- Thus, each station $i$ is allocated a specific $SA_i$ values such that
  \[ \sum SA_i \leq TTRT \]

- However, we have to account for the time to actually transmit the token, propagation time, and the time to get at least one frame around the ring:

  Total Synchronous Allocation = \( \sum SA_i \) + Time to transmit one token + Propagation time around the ring + Time to transmit a frame

  Total Synchronous Allocation \( \leq TTRT \)
• If this summation is less than TTRT, then all time left is considered “asynchronous allocation”.
  Asynchronous allocation = TTRT - Total Synchronous Allocation

• Operation:
  – Each station holds the following variables in a state machine.
    TTRT: Fixed constant, same for all stations
    \(SA_i\): pre-assigned allocation amount
    TRT: Token Rotation Timer - Amount of time before the TTRT
time expires
    THT: Token Holding Time - Amount of extra time left
    LC: Late Counter - Either 0,1, or 2, number of TTRT cycles
      that have elapsed since last token received.

  – TRT is a counter. It continually decrements, unless otherwise
    stopped, or reset.
  – Initialize: \(\text{TRT} \leftarrow TTRT; \ LC \leftarrow 0\)
  – While waiting for a token, the TRT continues to decrement. If it
    hits 0, then it increments the LC from 0 to 1, resets TRT, then
    continues waiting for token. If LC gets increment to 2, then the
    token is considered lost
  – If it receives a token, and the LC is zero, then TRT represents
    “extra time”. \(\text{THT} \leftarrow \text{TRT}, \ \text{TRT} \leftarrow TTRT\), enable TRT. Then,
    the station sends synchronous frames for a time \(SA_i\). After
    transmitting synchronous frames, or if there were no
    synchronous frames to transmit, THT is enabled. The station
    can transmit asynchronous frames as long as THT > 0.
  – If it receives a token and the LC is 1, then \(\text{LC} \leftarrow 0\), TRT
    continues to decrement. The station can only transmit
    synchronous frames for a time \(SA_i\).

• Example situation:
  – 4 stations. TTRT = 100 frame times. \(SA_i = 20\) frame times for
    each station. Each station is always prepared to send its full
    synchronous allocation and as many as asynchronous frames
    as possible. The total overhead during one complete token
    circulation is 4 frame times (one frame time per station).
### Operation of FDDI capacity allocation scheme

![Diagram showing the operation of the FDDI capacity allocation scheme](image)

<table>
<thead>
<tr>
<th>Arrival Time</th>
<th>TRT</th>
<th>Sync</th>
<th>Async</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>96</td>
<td>20</td>
<td>96</td>
</tr>
<tr>
<td>184</td>
<td>20*</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>263</td>
<td>36*</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>264</td>
<td>40*</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>452</td>
<td>52*</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>532</td>
<td>52*</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>644</td>
<td>60*</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>734</td>
<td>68*</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>836</td>
<td>68*</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>928</td>
<td>76*</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>1024</td>
<td>80*</td>
<td>20</td>
<td>0</td>
</tr>
</tbody>
</table>

*LC = 1; otherwise LC = 0
– FDDI Physical Layer Specification

Table 13.5 FDDI Physical Layer Medium Alternatives

<table>
<thead>
<tr>
<th>Transmission Medium</th>
<th>Optical Fiber</th>
<th>Twisted Pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data rate (Mbps)</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Signaling Technique</td>
<td>4B/5B/NRZI</td>
<td>MLT-3</td>
</tr>
<tr>
<td>Maximum number of repeaters</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Maximum length between repeaters</td>
<td>2 km</td>
<td>100 m</td>
</tr>
</tbody>
</table>

– FDDI digital signal encoding schemes
  
  • Differential Manchester used in Token ring is not used in FDDI, since 200 million baud rate would be needed for a 100 Mbps data rate.
  
  • To lower the baud rate and to maintain a synchronization ability, FDDI uses a 4B/5B code in conjunction with an NRZI (Nonreturn to zero inverted) technique.
  
  • For every 4 bits of data, a 4B/5B encoder creates a 5-bit code, which is then transmitted using NRZI.
  
  • Using this scheme, a signal will change at most 5 times for each 4 data bits. ⇒ 125M baud rate is enough.
  
  • The 4B/5B encoder never codes more than two consecutive binary 0s for data, ensuring that the signal is never constant for long periods.
  
  • This method preserves the self-synchronizing ability using a baud rate just 25% higher than the data rate.
# 4B/5B code groups

<table>
<thead>
<tr>
<th>Data Input (4 bits)</th>
<th>Code Group (5 bits)</th>
<th>NRZI pattern</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>11110</td>
<td></td>
<td>Data 0</td>
</tr>
<tr>
<td>0001</td>
<td>01001</td>
<td></td>
<td>Data 1</td>
</tr>
<tr>
<td>0010</td>
<td>10100</td>
<td></td>
<td>Data 2</td>
</tr>
<tr>
<td>0011</td>
<td>10101</td>
<td></td>
<td>Data 3</td>
</tr>
<tr>
<td>0100</td>
<td>01010</td>
<td></td>
<td>Data 4</td>
</tr>
<tr>
<td>0101</td>
<td>01011</td>
<td></td>
<td>Data 5</td>
</tr>
<tr>
<td>0110</td>
<td>01110</td>
<td></td>
<td>Data 6</td>
</tr>
<tr>
<td>0111</td>
<td>01111</td>
<td></td>
<td>Data 7</td>
</tr>
<tr>
<td>1000</td>
<td>10010</td>
<td></td>
<td>Data 8</td>
</tr>
<tr>
<td>1001</td>
<td>10011</td>
<td></td>
<td>Data 9</td>
</tr>
<tr>
<td>1010</td>
<td>10110</td>
<td></td>
<td>Data A</td>
</tr>
<tr>
<td>1011</td>
<td>10111</td>
<td></td>
<td>Data B</td>
</tr>
<tr>
<td>1100</td>
<td>11010</td>
<td></td>
<td>Data C</td>
</tr>
<tr>
<td>1101</td>
<td>11011</td>
<td></td>
<td>Data D</td>
</tr>
<tr>
<td>1110</td>
<td>11100</td>
<td></td>
<td>Data E</td>
</tr>
<tr>
<td>1111</td>
<td>11101</td>
<td></td>
<td>Data F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Control symbols</th>
<th>Encoded bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Halt</td>
<td>00100</td>
</tr>
<tr>
<td>Idle</td>
<td>11111</td>
</tr>
<tr>
<td>non-data-J</td>
<td>11000</td>
</tr>
<tr>
<td>non-data-K</td>
<td>10001</td>
</tr>
<tr>
<td>Quiet</td>
<td>00000</td>
</tr>
<tr>
<td>Reset</td>
<td>00111</td>
</tr>
<tr>
<td>Set</td>
<td>11001</td>
</tr>
<tr>
<td>Terminate</td>
<td>01101</td>
</tr>
</tbody>
</table>