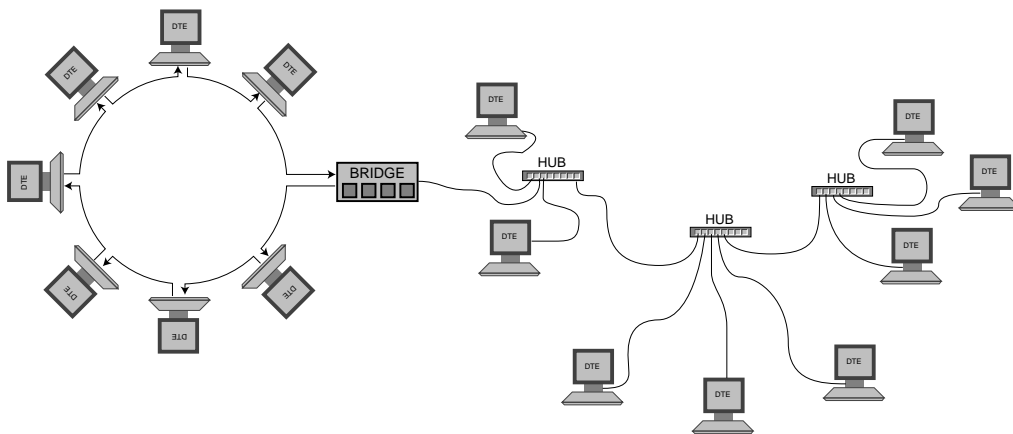


# Network Architecture

COSC 5110

Mark Zieg



Prof. David Metcalf

October 13, 1997

---

## Chapter 2: The Electrical Interface

### 2.1 Give a brief description of the application and limitations of the following types of transmission media:

- a. **Two-wire open lines**
- b. **Twisted-pair lines**
- c. **Coaxial cable**
- d. **Optical fiber**
- e. **Microwaves**

#### a. Open lines

These lines are cheap and easy to build and represent the wide variety of “straight-through” cables used in connecting peripherals to computing equipment. Such cables include classic serial (8, 9, 15, or 25-pin DB or DIN connectors), parallel (50-pin Centronics, ribbon cords), and “silver satin” phone cable. All these cables are customarily very short due to the ease with which they can pick up interference either from outside sources or from neighboring wires (crosstalk).

#### b. Twisted-pair

It turns out that by twisting two wires together you can greatly cut down on the amount of interference the wires pick up. This is because any interference inherited by one wire is almost guaranteed to affect the other wire when they are so closely intertwined. Since the main problem of signal interference is how it changes the voltage differential between the signals carried by two wires, it can be easier to try to equitably distribute interference universally than to try to block it altogether (via shielded cables). Standard unshielded twisted-pair lines are identified by category (Cat-3, Cat-5, etc), where the category refers to how many times each wire pair is twisted every inch: if you carefully stripped back the insulation on a 2-foot length of Cat-5 cable, you should expect to find at least 120 twists in each pair.

The main problem with this kind of cable is the *skin effect*. This refers to a physical phenomenon in which attenuation and media resistance increase at high signal frequencies, limiting the useful bandwidth of UTP over long distances. However, UTP is becoming extremely popular in LAN connections due to its low cost and the ease with which it may be wrapped, stripped, and crimped.

#### c. Co-axial

Co-axial was very popular for many years, and remains entrenched in the walls of many networked facilities. Unlike open cable, co-ax has natural shielding which minimizes interference, and unlike UTP co-ax can be used for extremely long runs at high bandwidth. For this reason, many early networks adopted co-ax—either thicknet or thinnet—for their backbone or ring network topologies. Co-ax is also the *de facto* standard for cable TV companies, who need its high bandwidth and reliable signal strength.

The main problem with co-ax is that it's a little hard to work with maintenance-wise. It's stiff, heavy, expensive, and awkward to crimp. Therefore many networks are gravitating to UTP for workstation connections and leaving co-ax (or more often fiber) for their building-to-building links and backbones.

#### d. Fiber

Fiber optical cable is a dream come true for bandwidth-hungry technology visionaries. It carries scads of data, quickly, reliably, and securely, with none of the interference or power problems evinced by electrical conductors. For instance, a single pair of fiber has been used to transmit data at over 3Gbps--try that with copper! Fiber is also much harder to tap than copper, especially in systems employing a carrier signal. Nor can electrical discharges, such as lightning bolts, propagate along fiber strands, since glass makes a fairly poor electrical conduit. For the same reason, you can run a fiber pair right past massive electrical generators, turbines, nuclear power plants, you name it, with only a thin layer of cladding to protect the integrity of the signal. Neat stuff.

The only real problems with fiber are cost. Termination gear is very expensive compared to copper variants, and F-type network cards still cost at least five times their copper equivalents. Finally, the lines themselves will remain expensive until volume picks up.

### e. Microwave

Microwaves represent the high-bandwidth end of the wireless spectrum. As such, their principle advantage is that they can be used to form data links between locations where physical cable connections would be prohibitively expensive, inefficient, or unsightly. There are two main kinds of microwave communication: satellite relay, in which ground signals are sent skyward to geosynchronous satellites for retransmission to one or more earth-based receiving stations, and terrestrial, in which microwave towers communicate directly over line-of-sight.

Besides freedom from cable connections, microwave communications are popular due to their relatively high bandwidth, option of coarse or narrow transmission, and broadcast capability. On the other hand, they are not suitable for every application. Transmitters/receivers can be both expensive and bulky, and of course satellites don't grow on trees. Signals can be subject to interference from weather, and microwave towers are subject to control by local zoning authorities.

## 2.2 With the aid of sketches, explain the differences between the following transmission modes used with optical fiber:

- a. **Multimode stepped index**
- b. **Multimode graded index**
- c. **Monomode**

### a. Multimode stepped index

Multimode stepped index suffers from a problem involving internal reflection. As light enters an optical core, it disperses in several directions. Any light not traveling straight down the fiber core strikes and is reflected by the inner surface of the cladding. This light keeps bouncing around, from wall to wall, until it reaches the end of the core. However, light bouncing at high angles of incidence will take somewhat longer to reach the cable end than light bouncing at low angles of incidence, due to the additional trips it must take across the core diameter. This causes some light pulses to arrive at their destination faster than others, with the result that data bits can overlap or arrive out of sequence at high pulse rates.

This is diagrammed in figure 2.2.1, which shows a stream of bits being transmitted through a section of fiber core, each color-block representing a frame of 1000 bits. The original light pulse is dispersed at three different angles, ranging from about 30° (triangles) to 60° (diamonds) from true. As you can see, although the frames remain mostly aligned at the beginning of the cable, by the end they are out of phase by as much as three full frames, creating a significant data jumble at the receiving end.

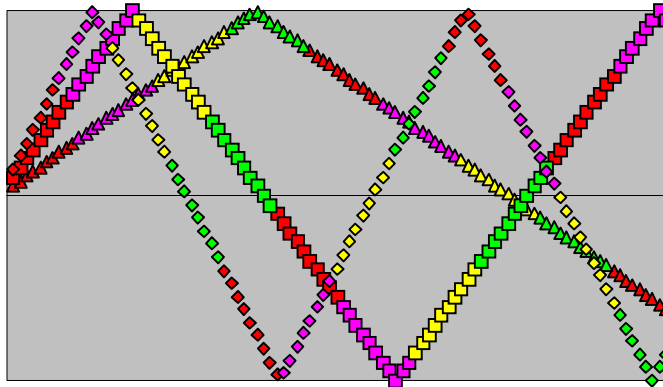


Figure 2.2.1

The simplest solution to this problem is to slow down the signal transmission rate so that all of the optical receiver can be certain that it has received all of the reflected impulses of one bit before the next is scheduled to arrive, preventing overlap. However, this decreases available bandwidth.

### b. Multimode graded index

Multimode graded index cable improves over stepped index by using a graduated core material with a variable refractive index. Light traveling near the axis of the core moves more slowly than light traveling toward the rim. Therefore, although light may be dispersed toward the cladding and thus have a longer distance to travel than light sent directly down the center of the core, the refracted light is allowed to move more quickly near the surface and thus arrives at the destination at almost the same time as the inner signal. The refractive traits of the core may be visualized as shown in Figures 2.2.2 and 2.2.3.

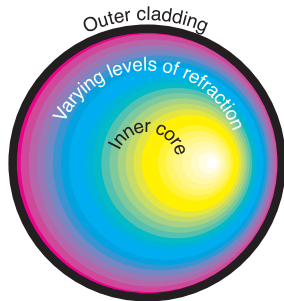


Figure 2.2.2

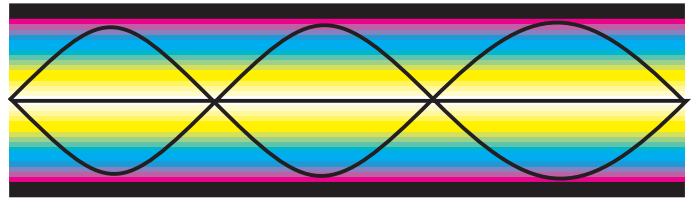


Figure 2.2.3

c. Monomode

Monomode cable is a second solution to the dispersion/refraction problem. By narrowing the core down to a diameter equal to a single wavelength of light, there is literally nowhere for pulses to go except for straight forward. By solving the bouncing problem without slowing either the pulse rate or the transmission medium, very high bit-rates can be attained. I assume, however, that monomode cable is significantly more expensive than other kinds, and possibly more fragile and difficult to work with (although cladding could probably solve that).

**2.3 The maximum distance between two terrestrial microwave dishes,  $d$ , is given by the expression:**

$$d = 7.14\sqrt{Kh}$$

**where  $h$  is the height of the dishes above ground and  $K$  is a factor that allows for the curvature of the earth. Assuming  $K = 4/3$ , determine  $d$  for selected values of  $h$ .**

As you can see in Figure 2.3.1, the advantage of adding height to a microwave relay tower tapers off rapidly, with a 600' tower providing only 17' more range than a 500' tower.

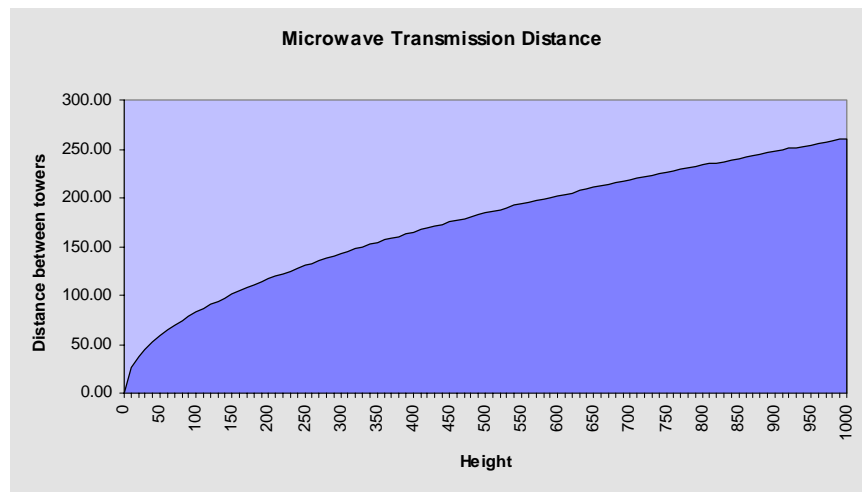


Figure 2.3.1

**2.4 With reference to Figure 2.5(b), determine the frequency assignments for a cellular system assuming a 7-cell repeat pattern. Explain the advantages of this over the 3-cell repeat pattern shown in the figure.**

Actually, I thought that the 3-frequency system was pretty clever and inexpensive. However, by expanding to a 7-frequency system you can decrease the likelihood of cross-cell errors and provide an easy way to monitor mobility between cells.

As shown in Figure 2.4.1, non-adjacent cells should never overlap; that is, no cell should share a common reception area with any cells except the six directly adjoining. Another way to say it is that no two cell coverage areas should overlap when the transmitters are separated by more than double the nominal maximum transmission distance.

However, sometimes geography and terrain, freak weather conditions, abnormal electrical behavior by the transmission or receiving units, or other unforeseen factors can unexpectedly enhance message transmission. In that case, otherwise non-adjacent cells could temporarily overlap, if only faintly. In the case of a 3-frequency system, such inappropriate overlaps would almost always cause signal errors. By expanding to a 7-frequency system, transmitters sharing the same frequencies would be separated by at least five nominal transmission radii, which vastly reduces the likelihood of crossover.

Also, and I don't know if anyone exploits this capability, 7-frequency plans could allow base stations to more easily track mobile receiving units. In Figure 2.4.1, for instance, any transition from a "Frequency 2" cell to a "Frequency 7" cell would indicate movement to the west, southwest, or south. On 3-cell systems, in contrast, transition from one frequency to another could indicate physical movement in literally any compass direction. Naturally, a hard-coded directional mapping between unique base stations could be used to derive the same data in a 3-freq schema, but the 7-cell variant is algorithmically driven rather than database-driven, and hence more elegant ☺

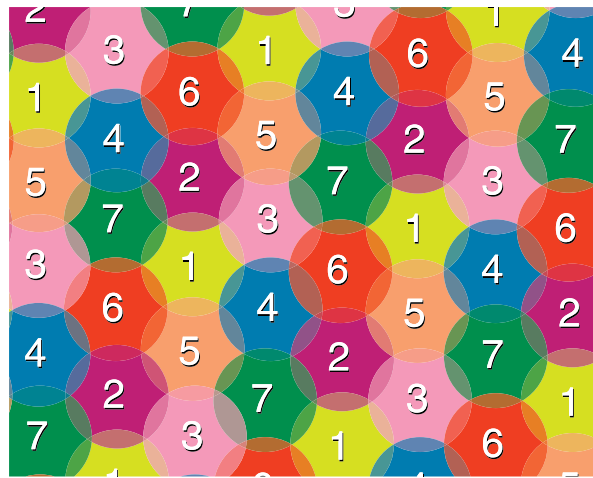


Figure 2.4.1

**2.9 A modem to be used with the PSTN uses QPSK modulation with four levels (phases) per signaling element. Assuming a noiseless channel and a bandwidth of 3000Hz, deduce:**

- a. The Nyquist maximum information transfer rate in bps
- b. The bandwidth efficiency of the modulation scheme.

$$C = 2W \log_2 M$$

$$C = 2 \times 3000 \times \log_2 4$$

$$C = 2 \times 3000 \times 2$$

$$C = 12000$$

$$B = \frac{1}{WT_b}$$

$$B = \frac{12000}{3000}$$

$$B = 4$$

a. Nyquist rate = 12000 bps

b. Efficiency = 4 bps/Hz

**2.15 Explain the terms 'signal propagation delay' and 'transmission delay'. Assuming the velocity of propagation of an electrical signal is equal to the speed of light, determine the ratio of the signal propagation delay to the transmission delay,  $a$ , for the following types of data link and 1000 bits of data:**

- a. 100m of UTP wire and a transmission rate of 1Mbps
- b. 2.5m of coaxial cable and a transmission rate of 10Mbps
- c. A satellite link and a transmission rate of 512kbps

Signal propagation delay ( $T_p$ ) refers to the time it takes a signal to propagate (move or transfer) from the transmitter to the receiver. Short of comparing the spin of separated quarks, this is traditionally limited by Einstein's self-imposed limit of  $c$ , or the speed of light. Other transmission materials such as copper, open sea, or

tupperware may impose significantly longer delays. The signal propagation delay is expressed as the ratio of the distance between nodes and the velocity of the signal passed between them.

Transmission delay ( $T_x$ ) refers to the related ratio of the quantity of data to be transmitted and the rate of data transfer. This ratio describes the speed with which a single frame (packet) of binary-encoded data may be passed over a specific link.

$$T_p = \frac{S}{V}$$

$$T_x = \frac{N}{R}$$

$$a = \frac{T_p}{T_x} = \frac{\frac{S}{V}}{\frac{N}{R}} = \frac{SR}{VN}$$

- a. S = 100m  
R = 1Mbps (1,000,000 bps)  
N = 1000 bits  
V = UTP ( $2 \times 10^8$  m/sec)

**RATIO =  $5 \times 10^{-4}$**

$$a = \frac{T_p}{T_x} = \frac{\frac{100}{2 \times 10^8}}{\frac{1000}{10^6}} = 5 \times 10^{-4}$$

- b. S = 2.5km (2500m)  
R = 10Mbps (10,000,000 bps)  
N = 1000 bits  
V = co-ax ( $2 \times 10^8$  m/sec)

**RATIO =  $1.25 \times 10^{-1}$**

$$a = \frac{T_p}{T_x} = \frac{\frac{2500}{2 \times 10^8}}{\frac{1000}{10^7}} = 1.25 \times 10^{-1}$$

- c. S = 50,000km (50,000,000m)  
R = 512kbps (512,000 bps)  
N = 1000 bits  
V = satellite ( $3 \times 10^8$  m/sec)

**RATIO = 8.53**

$$a = \frac{T_p}{T_x} = \frac{\frac{5 \times 10^7}{3 \times 10^8}}{\frac{1000}{5.12 \times 10^5}} = 8.53$$

- 2.17 a. **Why must a modem be used to transmit binary data through a PSTN? Use sketches and additional text to describe the following modulation methods:**
- i. **Amplitude shift keying (ASK)**
  - ii. **Frequency shift keying (FSK)**
  - iii. **Phase-coherent PSK**
  - iv. **Differential PSK**
- b. **Discuss the factors that influence the choice of carrier frequency and the bandwidth used by the demodulator section of a modem.**

The Public Switched Telephone Network, as its name implies, was designed to carry voice-quality signals falling within the limited frequency range of human speech. Since it only reliably transfers analog tones (waves) from 400Hz to 3000Hz, computers need a device to translate high-low binary digits to tenor-bass audible warbles. That device is a modem (modulator/demodulator), which "interprets" between the computer and the PSTN.

An important question is how to encode the binary values into the output analog signal. Analog signals are defined by several characteristics, especially amplitude (volume), frequency (pitch), and phase (time). Each of these characteristics can be used as a toggle to represent differing binary values, and each has advantages and disadvantages.

Amplitude Shift Keying (ASK)

Amplitude shift keying (ASK modulation) modifies the amplitude (volume) of a signal to convey data. For instance, a standard carrier wave  $v_c(t)$  can be transmitted to represent a binary 1, while no signal at all (or a very low amplitude signal) is sent to represent binary 0 (see figure 2.17.1). This method is extremely simple to design and understand, but didn't find much use due to its susceptibility to attenuation problems (which naturally reduce and distort amplitude).

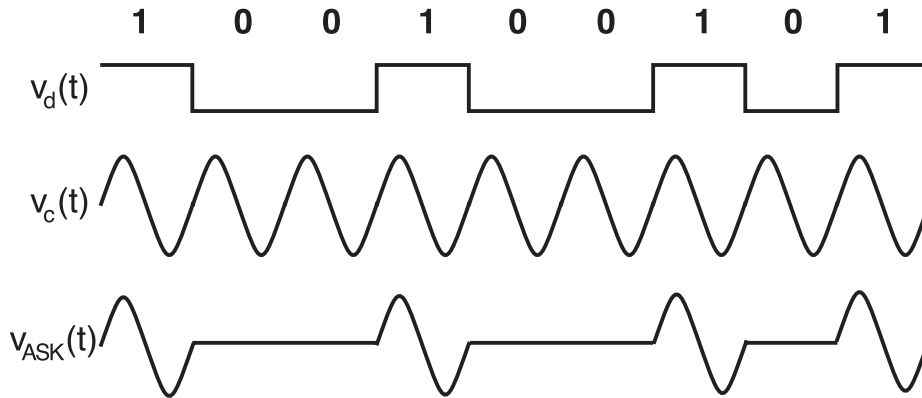


Figure 2.17.1

### Frequency-Shift Keying (FSK)

Frequency shift keying follows the same basic premise as ASK modulation except that it uses different *frequencies* (itches) rather than amplitudes to identify different data values. Basic FSK modulation employs two frequencies,  $v_1(t)$  and  $v_2(t)$ . In the example provided below,  $v_1(t)$  is used to represent the binary digit 1 and  $v_2(t)$  represents the binary digit 0. Since the amplitude is not significant in receiving a signal, attenuation problems are greatly reduced.

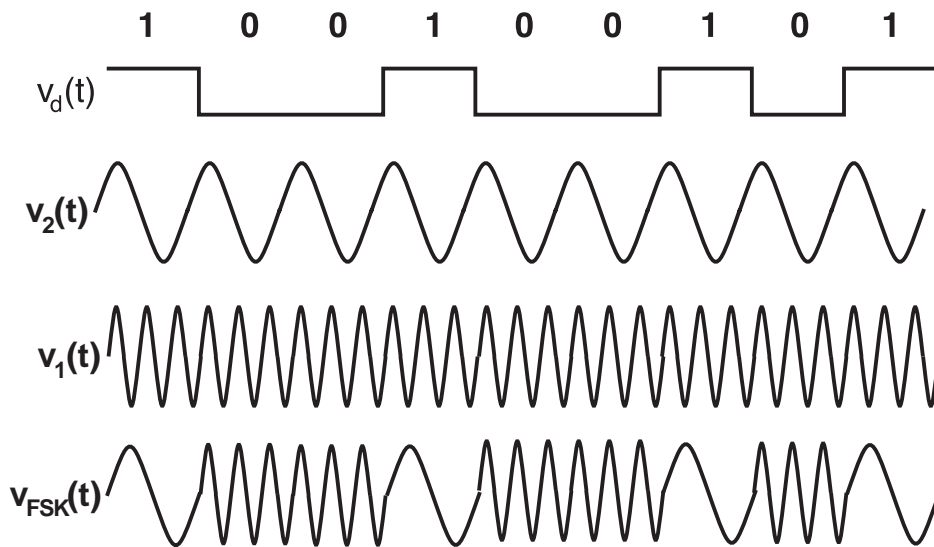


Figure 2.17.2

### Phase Shift Keying (PSK)

Phase Shift Keying (PSK) modulation is a little more complex than ASK or FSK algorithms, because actual phase-shifts are introduced into a signal. There are two breeds of PSK modulation: *phase-coherent* and *differential*. In Phase-Coherent PSK (shown in figure 2.17.3 as  $v_{PSK}(t)$ ), the carrier wave used to represent binary 1 is the inverse ( $180^\circ$  shift) of the signal for binary 0. In contrast, Differential PSK (shown in figure 2.17.3 as  $v'_{PSK}(t)$ ) adds a set shift amount to each data value:  $90^\circ$  for binary 1's and  $270^\circ$  for binary 0's. Besides eliminating the need for a reference carrier signal, this also forces a shift between every value, which assists in synchronization efforts.

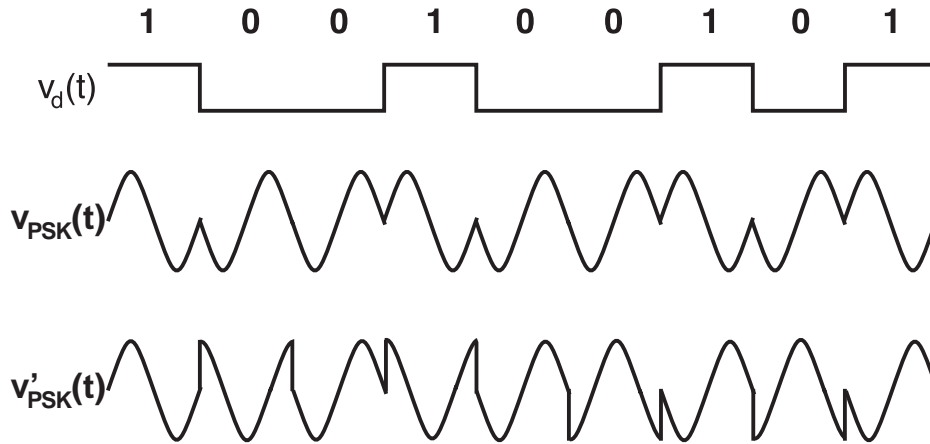


FIGURE 2.17.3

Carrier Frequency and Demodulator Bandwidth

All of these modulation methods produce their final output signals by adding or multiplying together several component signals. The combined waves approximate the hard-edged signal accuracy of raw digital signals, but ultimately retain their fundamental undulation. This intrinsic harmonic waver within a waveform causes potential problems when receiving a signal, because a signal can be expected to vary within a single sampling interval. To minimize these problems, modem protocols include limits on signal bandwidth, frequency, and sampling periods.

One of the results of combining waves in this manner is that "subwaves," or *harmonic components*, appear on the final signal. These are predictable, minor spikes symmetrically surrounding the main critical point within the curve. Modem circuitry can

**2.24 Explain the digital hierarchy used in North America and the ITU-T recommendation relating to leased digital circuits. Use sketches of the frame structure used in each of these schemes to show how the basic multiplexing groups are derived and hence derive the usable data rate with each circuit.**

Shown below are the frame formats currently in use in North America and Japan (Fig. 2.24.1) and the ITU-T recommendation (Fig. 2.24.2). Both formats are derived from the need to sample an analog voice signal 8,000 times each second, which translates to an eight-bit byte of data every 125 microseconds. However, the standards diverge on how many channels they bond together into a standard group. The Americans and Japanese use a 24-channel group, which provides 1.544 Mbps of useable bandwidth. This frame includes a special framing bit preceding each frame, provides 22 8-bit data slots operating at 64Kbps, and two 7-bit 56Kbps slots followed by signal bits on slots 6 and 12.

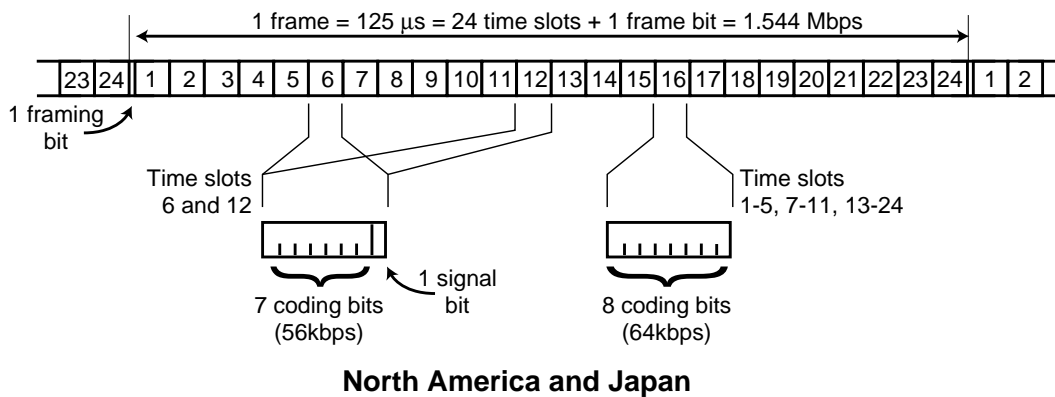
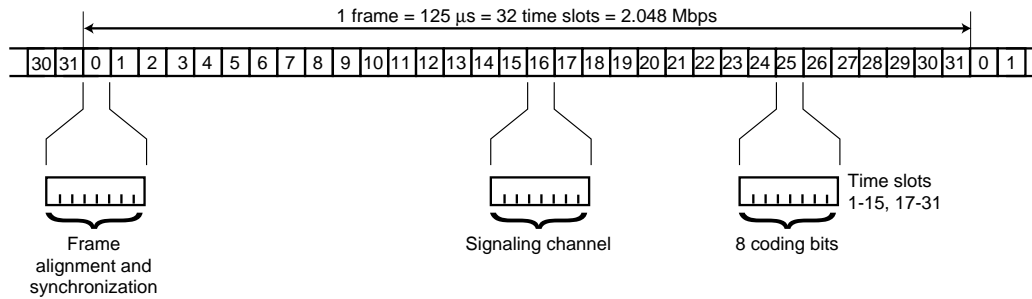


FIGURE 2.24.1



The ITU-T recommendation, on the other hand, suggests a 32-channel group, which offers 2.048 Mbps of bandwidth. The first frame slot is a full 8-bit frame alignment/synchronization byte, and the 17<sup>th</sup> slot is an 8-bit signaling channel. Otherwise, the remaining 30 slots are free for data.



**International Telecommunications Union — Telecommunications (Sector)**  
**ITU-T**

FIGURE 2.24.2

## Chapter 3: Data Transmission

- 3.1 a. Explain the difference between asynchronous and synchronous transmission.  
 b. Assuming asynchronous transmission, one start bit, two stop bits, one parity bit, and two bits per signaling element, derive the useful information transfer rate in bps for each of the following signalling (baud) rates:
- i. 300
  - ii. 600
  - iii. 1200
  - iv. 4800

The main difference between synch and asynch transmission is the clock in the receiver. Synchronized receivers operate in synchronization with the incoming signal, whereas asynchronous receivers use more of a store-and-forward inbox system. The differences in efficiency and complexity suggest different applications for each transmission mode.

Asynchronous transmission is used in cases when data is sent at random or quasi-random intervals. This is an easy, low-overhead system suitable for low-bandwidth transmissions. Asynch bytes are sandwiched between a start bit and one or more stop bits. The start and stop bits are of opposite polarity, which aids in retaining synchronization and finding a proper sampling instant. However, these additional bits constitute an overhead of 25-38%, which significantly slows down large data transfers. Therefore, an alternate transfer mode is provided, *synchronous transmission*.

Synch transfer is conducted in large frames, or blocks, in which special frame-begin and -end characters provide synchronization initialization. However, the transmitted data must often be processed in order to create sufficient transitions to maintain synchronization throughout the transmission.

$$\begin{aligned} \text{Character size} &= 8 \text{ data bits} + 1 \text{ start bit} + 2 \text{ stop bits} + 1 \text{ parity bit} \\ &= 12 \text{ bits} \end{aligned}$$

Signalling Rate (baud)	Data Transfer Rate (bps)
300	25
600	50
1200	100
4800	400

- 3.7 Assuming a synchronous transmission control scheme, explain how character and frame synchronization are achieved:
- a. With character-oriented transmission
  - b. With bit-oriented transmission

There are two key methods applied in both character- and bit-oriented synchronization: *clock encoding* and *digital phase-lock-loops*. In turn, there are at least three standard ways to implement clock encoding: bipolar, Manchester, and differential Manchester. Bipolar encoding uses three signal levels (positive, zero, and negative) in a return-to-zero scheme. This ensures that even constant streams of zeros or ones will contain regular clock pulses to establish synchronization. Manchester encoding only uses two signal levels, but still guarantees a transition in the center of each bit cell. Binary zeros are represented by high-low transitions and binary ones by low-high transitions. Transitions will also occur between bit cells in the case of repeated bit values. With differential Manchester encoding, in contrast, data is encoded via a second transition at the beginning of 'zero' bit cells. This allows the recipient to both decode the transmitted data and stay in synch.

Digital phase-lock-loop schemes don't attempt to provide synchronization data with every transmitted bit. Instead, they assume that, once synchronized with a header transmission, the receiver is capable of maintaining a basic rhythm. Only "reminder" transitions are inserted every byte or so to make sure that the receiver hasn't slipped too far from true, and basic zero-insertion techniques are sufficient to force zero-tagged transitions into the transmitted frame.

Once one or more synchronization methods have been chosen for a particular application, designers must also decide whether it would make more sense to utilize character-oriented or bit-oriented transmission. Character-oriented transmission, as its name implies, is commonly applied to text transfer. It uses special control-code bytes to establish the start and end of frames (STX and ETX) and provide synchronization between frames (SYN). This system is very simple and works for many common applications, but is basically limited to the 8-bit unit and has trouble working outside of the ASCII framework. Bit-oriented transmission, on the other hand, uses arbitrary bit strings for delimiters. This makes it a lot more flexible with regards to binary data and obscure formats, but requires a bit more processing and design-side work.

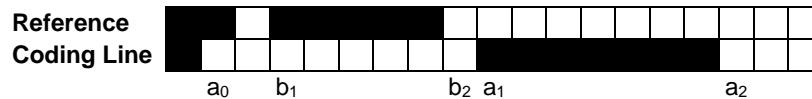
Both systems deal with trivial cases, in which transmission control codes and delimiters may naturally occur in binary data streams, by "escaping" special characters, or using a bit-stuffing technique.

**3.21 With the aid of example pel patterns, explain the meaning of the terms (as used with Group 4 fax machines):**

- a. Pass mode
- b. Vertical mode
- c. Horizontal mode

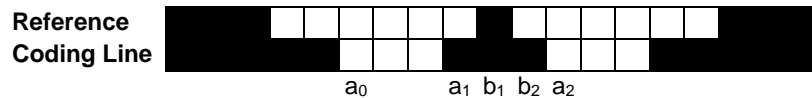
**Use your examples to deduce an algorithm to perform the encoding operation.**

- a. Pass Mode



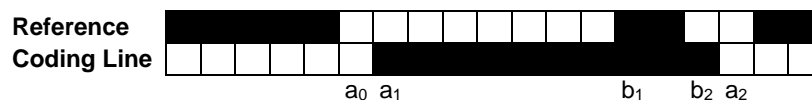
Pass mode occurs when the next block of the coding line-- $a_1a_2$ --is completely past the next block of the reference line. In this case,  $b_1a_2$  falls within  $a_0a_1$  and so no overlap at all takes place. Therefore  $b_1b_2$  is encoded as a change (XOR) from the reference line.

- b. Vertical Mode



Vertical mode refers to the common condition in which the next block of the coding line overlaps the next block of the reference line with three pels or so; in other words,  $|a_1 - b_1| < 4$ . In the example shown,  $a_1b_1$  is encoded to reflect the change.

- c. Horizontal Mode



Horizontal mode is the final case in which  $|a_1 - b_1| > 3$ . In this case, two entirely new runs are encoded (not XOR of the reference line):  $a_0a_1$  and  $a_1a_2$ .

The following could constitute pseudocode for a basic implementation:

```
void output( int mode, int code[], int ref[], int a0, int a1, int a2, int b1, int b2 )
{ switch mode
  { PASS:
    printf( "0001%s", tablelookup( len( b1, b2 ) ) );
    break;
  HORIZONTAL:
    printf( "001%s%s", tablelookup( len( a0, a1 ) ), tablelookup( len( a1, a2 ) ) );
    break;
  VERTICAL:
    diff = a1 - b1;
    switch diff
      { 0: printf( "1" ); break;
        -1: printf( "011" ); break;
        -2: printf( "000011" ); break;
        -3: printf( "0000011" ); break;
        1: printf( "010" ); break;
        2: printf( "000010" ); break;
        3: printf( "0000010" ); break;
      }
    }
}

void Group4Encoding
{ bit refLine[ MAXLINE ], codeLine[ MAXLINE ];
  int a0, a1, b1, b2;
  int color;

  fill( refLine, WHITE, MAXLINE );
  codeLine[ 0 ] = !EOP;
  while( codeLine[ a0 = 0 ] ) != EOP )
    { while( codeLine[ a0 ] != EOL )
      { color = codeLine[ a0 ]; // determine a1, a2, b1, b2
        a1 = nextindex( codeLine, a0, !color );
        a2 = nextindex( codeLine, a1, color );
        b1 = nextindex( refLine, a0, !color );
        b2 = nextindex( refLine, b0, color );
        if( b2 < a1 )
          { output( PASS, refLine, codeLine, a0, a1, a2, b1, b2 );
            a0 = b2;
          }
        elseif( abs( a1 - b1 ) < 4 )
          { output( HORIZONTAL, refLine, codeLine, a0, a1, a2, b1, b2 );
            a0 = a2;
          }
        else
          { output( VERTICAL, refLine, codeLine, a0, a1, a2, b1, b2 );
            a0 = a1;
          }
        }
      }
    copy( refLine, codeLine );
  }
}
```

**3.22 Explain the difference between a time-division multiplexer and a statistical multiplexer. Produce a sketch showing the internal architecture of a time-division multiplexer and explain its operation. Describe the organization of the shared data link and how the controlling device determines the destination of each received character.**

The main differences between a time-division multiplexer and a statistical multiplexer is that time-division MUXes *guarantee* a *slow but fixed* bit rate using an expensive high-speed connection, while statistical MUXes deliver the *probability* of a *reasonable* bit rate using a cheap slow connection. This is how they work:

Time-division MUXes divvy up a high-speed shared line (say 56K) and provide equal-sized slices to individual terminals in what resembles a slotted bus frame-packing technique. Each terminal retains the rights to its "slot" whether or not it happens to be transmitting at that instant. This means that, on the one hand, a user can use the system whenever they want and expect a constant responsiveness; on the other hand, a majority of slots are likely

to be unused at any given second, meaning the company is paying for unused bandwidth. Personally, I'd rather use a time-division MUX myself, because I'm a sworn propeller-head who likes everything quantifiable and predictable, and I don't like my response time to vary based on other people's behavior. But that's just me.

Noteworthy cost savings can be realized by switching to a statistical MUX, which is designed with that wasted bandwidth in mind—it depends on it, in fact. Statistical MUXes proponents scrimp by buying only enough bandwidth to support a handful of terminals—and then hook a whole department to it. They plan to get away with their tightfistedness by betting that only a handful of users will actually hit a key at any given second, and therefore they only need enough bandwidth to support those few keyclicks (9.6k is often enough for several users). Of course, all of this falls apart when it's a production crunch on Friday afternoon and suddenly everyone is banging on their XMT keys and demanding to know what the problem is...

Following is a sample diagram of a time-division MUX. The asynch UART's on the right are hooked to individual terminals, which exchange data with the MUX at a low bit rate. Each UART is buffered, and the CPU pulls the contents of each buffer, and serially transfers them through the synchronous USRT at high speed. In the sample given, if the USRT can communicate four times as fast as the individual UART/terminals, then the entire process will appear transparent and each terminal will feel that it has a direct and unobstructed, if somewhat slow, connection to the remote host (or whatever).

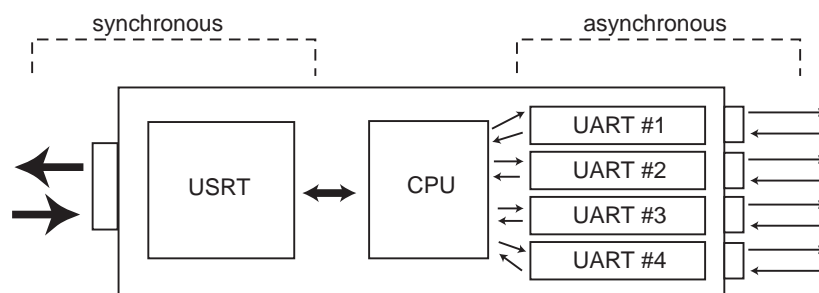


FIGURE 3.22.1

## Chapter 4: Protocol Basics

### 4.1 Assume a terminal is connected to a computer. Explain the two techniques that are used to achieve error control and flow control. Clearly outline the effect of each mechanism on the user of the terminal.

There are two fundamental types of error control: automatic and user-provided. Ideally, the underlying network protocols will detect and correct any transmission errors, but in the case that they miss, the user can visually verify the characters echoed back to his or her screen. Of the automated techniques, there are two major categories: idle RQ and continuous RQ. Idle RQ isn't particularly efficient, but it works well enough for most character/terminal applications (good enough for government work, as they say). Continuous RQ is more complex and requires more intelligent network interfaces, especially on the transmitting side, but it produces a much faster transmission stream.

Idle RQ is based upon the popular and easily understood ACK/NAK system. When the sender (either the terminal or the computer) wants to send a message to the recipient, they break the message into a series of packets, or frames (TCP, IPX, etc). After the sender issues each frame, they wait for an ACKnowledgement message to be sent back to indicate successful completion. However, the sender won't wait forever—after a while the sender gets bored (times out) and re-sends the last packet. On the off-chance that it was actually the ACK which got lost or corrupted, the recipient disregards repeated frames based on their sequential serial numbers. A faster version of this system employs an explicit NAK (negative acknowledgement) from the recipient when a corrupted frame was received, which bypasses the otherwise expensive timeout period.

Continuous RQ works on a similar principle to idle RQ with NAK messages, in that it's still basically the recipient's job to figure out when an error has occurred. However, continuous RQ doesn't bother with ACK messages—it blithely assumes that every packet is received correctly. As a result, it may have transmitted a half dozen new packets before it receives the NAK message indicating that an old frame (N-6) was lost or corrupted. It must therefore retransmit at least the missing frame, in a technique called selective repeat; another scheme called go-back-N retransmits every subsequent packet as well. In order to retransmit any packets, the sender must maintain a robust cache of recently transmitted frames. This buffer must hold each packet which has been sent but has not yet received an ACK. As ACKs are received, buffered packets get bumped off the queue in FIFO order.

Since the buffer is of finite size, transmission will eventually come to a halt if ACKs are particularly slow in coming. If a particular ACK never arrives before the established timeout, then a retransmit is triggered exactly as if a NAK had been received.

**4.2 Explain the meaning of the following terms relating to a data link control protocol:**

- a. **Connectionless**
- b. **Connection-oriented**

When two processes are communicating over a connectionless protocol like UDP, there is no convenient method to request retransmission of individual characters within a block, nor is there a way to temporarily pause transmission of a block halfway to correct a corrupt byte. As a result, entire blocks have to be retransmitted in a relatively expensive operation that can significantly impact total system response. As a result, connectionless protocols are usually used for applications where corrupt or lost packets can simply be dumped without retransmission. Data sent from the Mars Pathfinder probably falls into this category, since retransmission of packets across millions of miles of space would be inconvenient at best. For this reason, connectionless protocols are also referred to as “best try.”

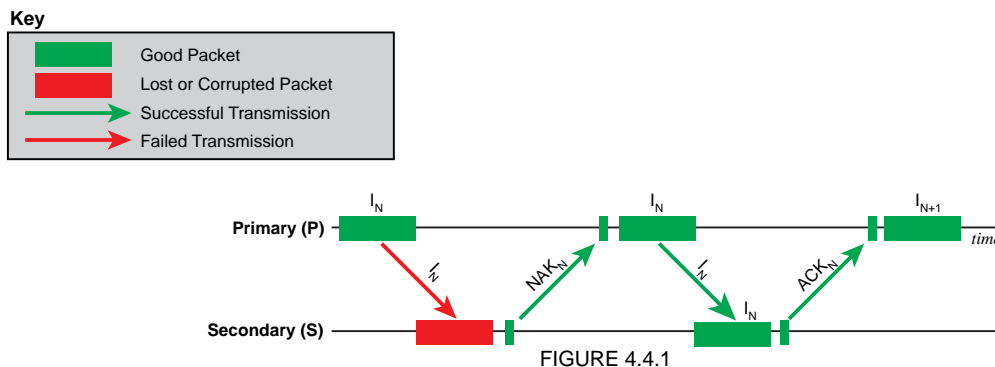
On the other hand, connection-oriented protocols like TCP allow duplex communication and mid-stream error correction. Connection-oriented protocols are considered “reliable” because the primary and secondary processes are able to verify the correct transfer and receipt of exchanged data.

**4.4 With the aid of frame sequence diagrams and assuming an idle RQ error control procedure with explicit retransmission, describe the following:**

- a. **The factors influencing the minimum time delay between the transmission of two consecutive information frames**
- b. **How the loss of a corrupted information frame is overcome**
- c. **How the loss of a corrupted acknowledgement frame is overcome**

The minimum time delay between the transmission of two frames ( $I_N$  and  $I_{N+1}$ ) is determined by several factors. First, there is the underlying network latency time. When using an idle RQ scheme, network latency is doubled since an acknowledgement packet (either positive or negative) must be sent in receipt. Added to that is the time it takes to transmit the frame itself. Added to that is the time it takes for the recipient to decide that the frame has been either properly received, lost, or corrupted, and hence which acknowledgement to send. The maximum time cannot be determined, because although a timeout value prevents lost frames or acknowledgements from hanging systems, a very dirty or physically damaged network can cause numerous transmission faults for a particular frame.

If a particular frame is lost or corrupted in transmission from the primary system to the secondary (ie, the frame fails error-checking on receipt, or is never received at all within the timeout limit), the secondary system sends a Negative Acknowledgement (NAK) token back to the primary. The primary then re-sends the faulty frame, and life goes on from there. This process is shown in figure 4.4.1.



On the other hand, if it's the positive acknowledgement (ACK) token which gets fumbled, then the primary system eventually times out and assumes that a transmission fault has occurred. Therefore it re-sends the last packet, which presumably is again received successfully by the secondary. To prevent duplicate information from being logged, the secondary system uses the serialized frame identifiers to recognize duplicate packets and ignore them. This process is diagrammed in figure 4.4.2.

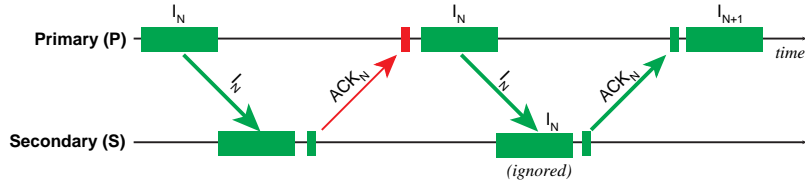


FIGURE 4.4.2

- 4.13 Using a continuous RQ error control scheme as an example, describe how the operation of the primary and secondary side of a link may be defined in the form of a finite-state machine and:
- A state transition diagram
  - A state transition table
  - A high-level language program segment written in pseudocode

The primary side of continuous RQ transmission utilizing Go-Back-N error control can be defined as a finite state machine. Such a machine could be visualized with a state transition diagram as shown in figure 4.13.1. Another way to describe the machine specification would be with a state transition table as shown in figure 4.13.2.

*Note: I haven't done NFA's in a while, but I think that some of the compound condition/action states I've defined should properly be broken down into multiple states. For simplicity, I left it as-is.*

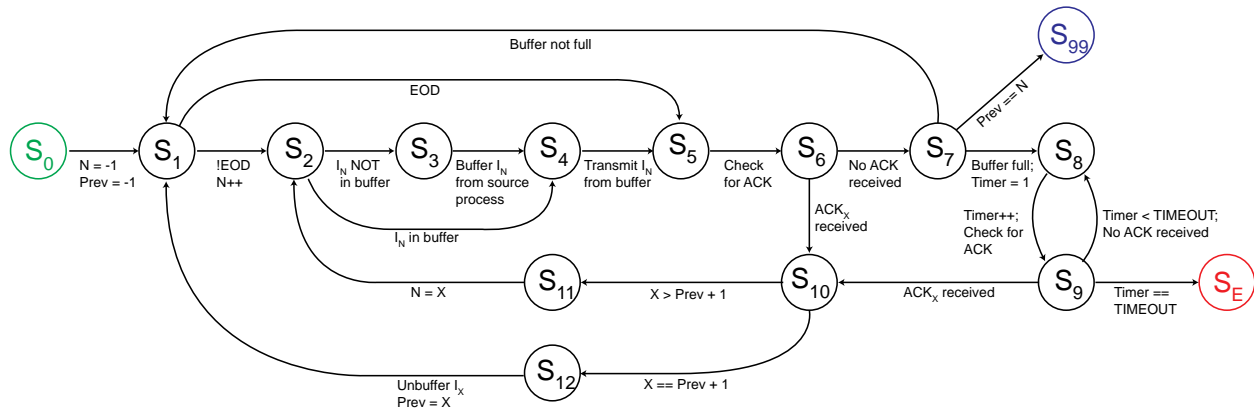


FIGURE 4.13.1

State ID	Condition	Actions	Next State
0	Opening State	Open Link Set N = -1 Set Prev = -1	1
1	Not End-of-Data	N++	2
1	End-of-Data		5
2	$I_N$ in buffer		4
2	$I_N$ not in buffer		3
3		Copy $I_N$ from source process to buffer	4
4		Transmit $I_N$ from buffer	5
5		Check for ACK	6
6	No ACK received		7
6	$ACK_X$ received		10
7	Prev == N		99
7	Buffer not full		1
7	Buffer full	Set timer = 1	8
8		Timer++	9
8		Check for ACK	9
9	$ACK_X$ received		10
9	Timer < TIMEOUT		8
9	Timer == TIMEOUT		Err
10	$X > Prev + 1$		11
10	$X == Prev + 1$		12
11		Set N = X	2
12		Unbuffer $I_X$ Set Prev = X	1
99		Close link	DONE

FIGURE 4.13.2

Finally, the machine could be implemented in a high-level programming language using an algorithm similar to the following pseudocode:

```

void ContrQ_GoBackN( process *primary, process *secondary )
{ int n, prev;
  queue buffer[ BUFFER_SIZE ];
  boolean found;
  ack_type ack;

  initlink( &n, &prev, &done ); // S(0)
  while( 1 )
  { if( !endofdata() ) // S(1)
    { if( !inbuffer( n ) // S(2)
      { copytobuffer( primary, n ); // S(3)
      }
      transmit( secondary, buffer[ n ] ); // S(4)
    }
    found = checkforACK( secondary, &ack ); // S(5), S(6)
    if( found )
    { state_10( &x, ack, &prev, &n, buffer ); // S(10)
    }
    else
    { if( prev == n )
      { return( 1 ); // S(99)
      }
      elseif( isfull( buffer ) ) // S(7)
      { timer = 1;
        while( ( found = checkforACK( secondary, &ack ) && ( timer++ < TIMEOUT ) )
          ; // S(8)
        if( timer == TIMEOUT ) // S(9)
        { return( ERROR_TIMEOUT ); // S(Err)
        }
        else
        { state_10( &x, ack, &prev, &n, buffer ); // S(10)
        }
      }
    }
  }
}

void state_10( int *x, ack_type *ack, int *prev, int *n, queue buffer[] )
{ *x = getACKid( ack );
  if( *x > *prev + 1 )
  { *n = *x; // S(11)
  }
  else
  { dequeue( buffer );
    *prev = *x; // S(12)
  }
}

```

**4.19 Explain what is meant by the term ‘link management.’ Use an example set of user primitives and a time sequence diagram to show how a logical communication path is established (set-up) between two systems and subsequently cleared (disconnected).**

Link management deals with establishing a communication path between two processes so that data transfer can occur. Data transmission error and flow control procedures take place within and across this link. However, the link routines are themselves responsible for initializing, maintaining, and shutting down a link before and after transmission. This is visualized in figure 4.19.1 below.

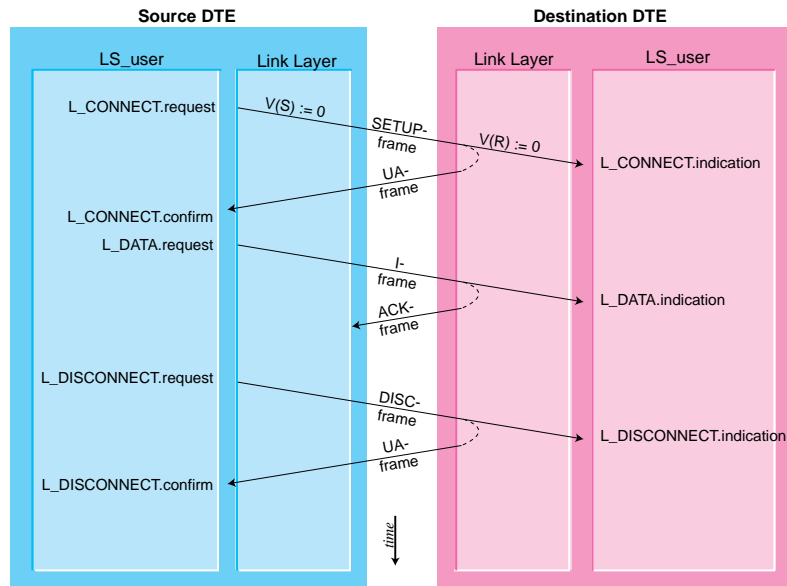


FIGURE 4.19.1

## Chapter 5: Data Link Control Protocols

### 5.1 Explain the meaning of the following terms relating to data link protocols:

- a. **Character-oriented** Character-oriented protocols are designed to facilitate the exchange of textual data. These protocols are simpler than bit-oriented because they're able to make certain assumptions about the nature and format of the data which they're transmitting. For instance, characters data is typically stored as an 8-bit byte, so a block is already available for synchronization and sub-framing. Moreover, the majority of characters exchanged fall within less than half of the available bit combinations, leaving plenty of rarely-used sequences available for control tokens and metadata.
- b. **Bit-oriented** Bit-oriented protocols aren't able to make the 8-bit byte assumption of character data, nor can they assume that any particular bit sequence can be safely reserved for control information. This makes bit-oriented protocols more powerful and flexible, because they're designed to be able to handle virtually any conceivable stream of digital data, but they're also more difficult to implement and sometimes have higher overhead requirements.
- c. **Framing and data transparency** Framing is the technique of blocking data into a predefined packet template which can hold not only a series of data bits but also header and footer information such as recipient and sender addresses, data length, checksums, serial packet identification, etc. There are innumerable advantages to enveloping data within frames, and only a few disadvantages. Frames make it possible to break long messages into shorter units which promote network efficiency. They provide a mechanism for guaranteeing proper serial receipt of data. They allow error correction within message units. They also provide a useful degree of data abstraction, allowing different protocol layers, stacks, and network hardware devices to cooperate in the exchange of data without necessarily comprehending the contents. Some of the few disadvantages include an increase in the total number of bits transmitted and additional computational requirements as frames are created, relayed, checked, and decoded. Data transparency is ensured by "escaping" or "backslashing" any control codes within a frame that could otherwise be interpreted as data bits.
- d. **Poll-select** Poll-select mode is a common way for hosts to communicate with terminals over a shared data link. When the central computer is ready to receive data from the DTE, it sends a "poll" message to that terminal. If the terminal is ready to send data at that instant, it goes ahead



and transmits data over the link. If the terminal is not ready to send data, then the poll is ignored. Likewise, select tokens are issued by the host when it desires to send data to a terminal. If the terminal is not ready to receive data, the data remains blocked at the host, but otherwise the new information—typically a new screen or form—is sent to the DTE.

- e. **Primary and secondary** The primary process is typically taken to be the one attempting to send data, and the secondary process is the recipient of the data. Even though both processes may be communicating with each other, say over a duplex connection, the stream of data—not control information—determines the primary and secondary ordering.

**5.3 With the aid of sketches, identify the scope of operation of the data link protocol in the following application environments:**

- a. **Point-to-point**
- b. **Multipoint (multidrop)**
- c. **WANs**
- d. **LANs**

Two sample point-to-point data link operations are shown in figures 5.3.1 and 5.3.2. In the first, a data link has been established directly between two computers, perhaps through serial cords or a cross-wired 10BASE-T cable. In the second example, two computers are communicating via modems over the public switched telephone network. In both cases the DLP provides a direct point-to-point connection service.

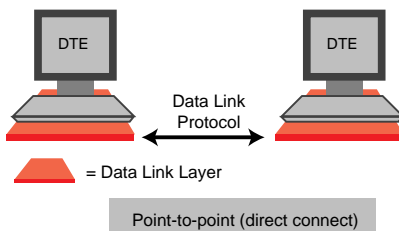


FIGURE 5.3.1

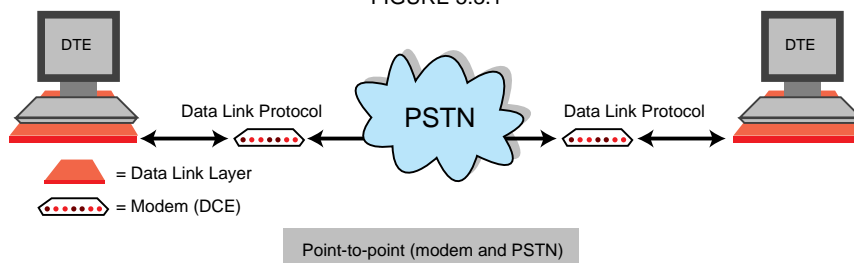


FIGURE 5.3.2

Shown in figure 5.3.3 is a typical multipoint network in which a single host, or “master” computer needs to communicate with various slaves (DTEs).

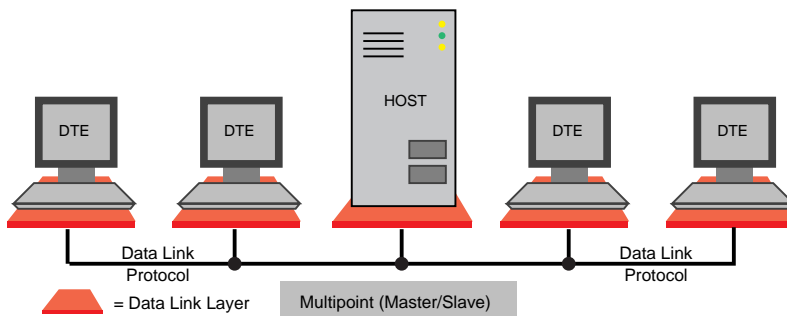


FIGURE 5.3.3

Figure 5.3.4 shows a more elaborate situation in which two systems are communicating over a wide-area network through packet switch exchanges. Unlike the peer-to-peer communication provided in each of the other examples, in this case the DLP just handles communication between the DTE and the PSE, and data is enveloped in different frame formats across the X.25 network (I think).

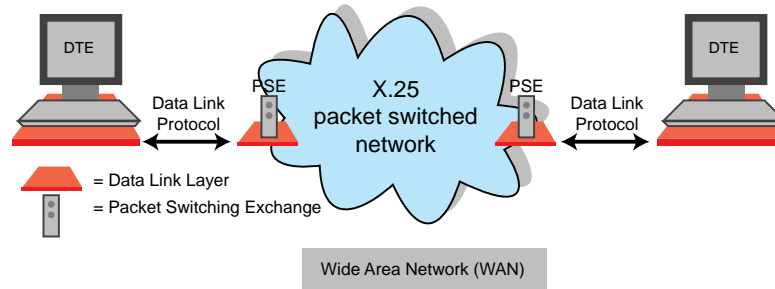


FIGURE 5.3.4

Finally, the DLP handles station-to-station communication over a typical LAN as shown in figure 5.3.5.

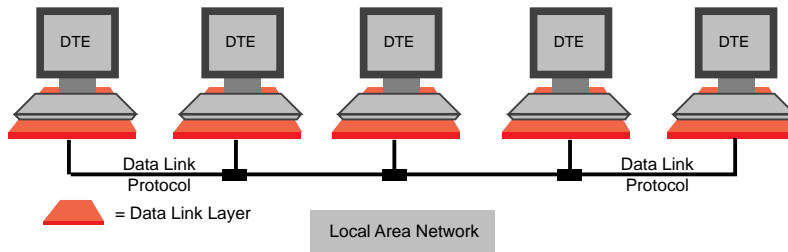


FIGURE 5.3.5

**5.4 Explain the operation of the Kermit protocol used for the transfer of files of data from one computer to another. Include in your description:**

- a. The user commands
- b. The frame format and frame types used
- c. Example frame sequences, including retransmissions

Kermit used to be a popular simplex point-to-point transfer protocol for both PCs and hosts, although most PCs have since gravitated first to X/Y/Z-modem and then to FTP. A summary of the basic command set is shown in figure 5.4.1.

Command	Function
KERMIT	Opens program
CONNECT	Establishes data link
RECEIVE	Configures slave to prepare to accept incoming data stream
SEND <i>filename</i>	Directs master to begin transmission of specified file
EXIT	Terminates (concludes) program session

FIGURE 5.4.1

There are several kinds of frames in use by Kermit. They all fall into one of two categories: *information* (I) frames and *acknowledgement*. The first to be transmitted is the send-invitation frame (S), which contains parameters such as the maximum frame length for the forthcoming transmission, a reasonable timeout period, etc. Upon receipt of an S-frame, the receiving station returns an acknowledgement (Y) frame to signal agreement on the supplied parameters. Then the primary station sends a File Header (F) frame which holds the filename of the file being transmitted. (An acknowledgement frame should be returned from the secondary process after this and every subsequent information frame.) Then the meat of the transfer occurs with a series of data (D) frames. These hold the actual bits, bytes, and what-not being exchanged. At the end of a file, an eof (Z) frame is sent. If more than one file is being sent in a given session, additional "FD+Z" (regexp) sets may follow. When all is through and done with, an end-of-transmission (B) frame is sent to close the connection. A sample frame diagram is provided in figure 5.4.2.

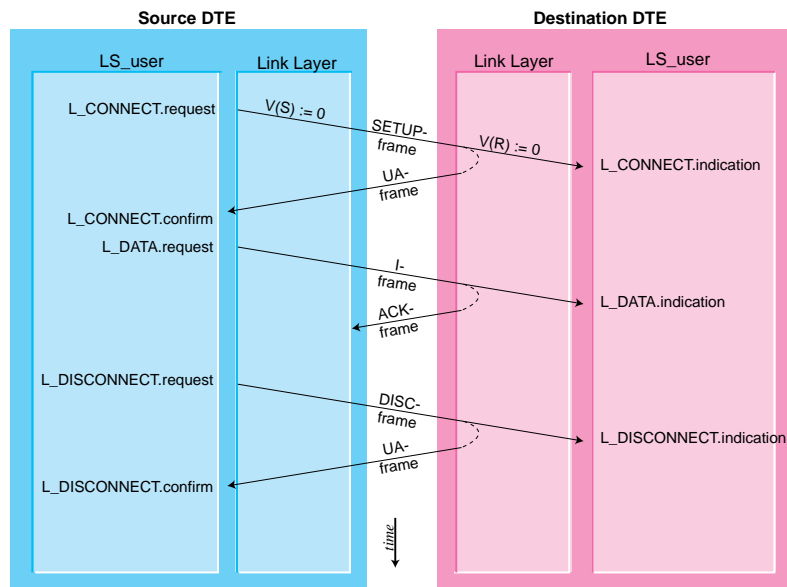


FIGURE 5.4.2

- 5.17 With the aid of a time sequence diagram, list the user service primitives associated with the LLC protocol with the following operation:
- Connectionless
  - Connection-oriented
- In relation to the above, discriminate between the terms:
- Send data with no acknowledge
  - Send data with acknowledge
  - Request data with reply

Connection-oriented control protocols use an L\_CONNECT service to establish a logical connection and an L\_DISCONNECT service to break the connection. There is also an L\_DATA service to provide for the actual transmission. The connection services provide the following three primitives: .request, .indication, and .confirm. The .request primitive is sent in a “setup” frame to the link layer of the receiving computer to ask whether a connection might be set up. If this is agreeable to the recipient, two primitives are returned: a “.confirm” primitive is fired back to the requesting computer, and an “.indication” primitive sent to the application layers of the receiving station.

Once that’s been taken care of, the L\_DATA service can take the field. L\_DATA only provides two primitives: .request and .indication. Although confirmation of transmitted data does take place (via ACK-frames), it’s handled within the DLP and shielded from the user layers. The entire point of confirming to the user layers than a connection has been established is to let them know that data can be exchanged more-or-less error free, and that the applications don’t have to worry about it. Finally, once all of the data has been transmitted, the L\_DISCONNECT service, with its three primitives, is used to shut down the connection and alert all interested parties (including both logical users).



- 5.18 Use a time sequence diagram to show the user service primitives associated with the medium access control (MAC) sublayer used in the data link layer of LANs. Hence show how the supervisory and information frames associated with the above LLC sublayer user services are transferred using the various MAC user primitives.

The MAC layer service provides three primitives: MA\_UNITDATA.request, .indication, and .confirmation. The .request and .indication primitives operate much like one would expect, initiating a data transfer between corresponding LLC layers and alerting processes of the arrival of transmitted data. However, the .confirm primitive is LAN dependant and may represent confirmation of received data, or merely confirmation that it was sent.

The specific parameters to the .request primitive include the destination address, which supports multicast and broadcast capabilities as well as point-to-point, the class of service desired (where available), and of course the embedded LLC frame containing the data itself. The LLC frame can contain all of the usual high-level data link control entities such as RR, RNR, etc.

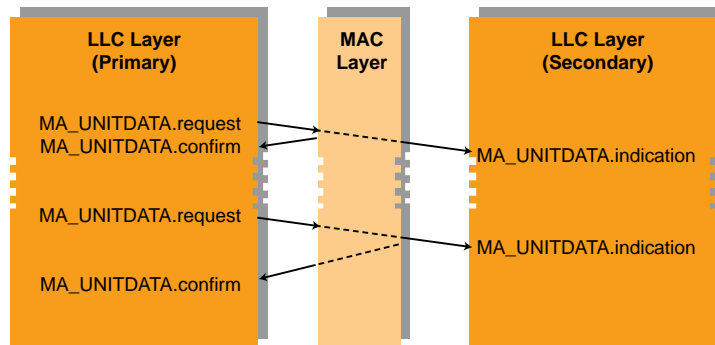


FIGURE 5.18.1

## Chapter 6: Local Area Networks

### 6.1 List the four main types of network topology currently in widespread use for LANs and, with the aid of sketches, explain their operation.

Star topologies, also known as “homeruns,” are very common for small and low-budget LANs, largely due to their low cost and ease of configuration. The basic idea is that each node is directly connected to a stack of hubs. Any given node can be another hub, providing easy expansion and a high degree of flexibility to move things around without ripping out walls. Hubs can be nested this way up to three deep, which provides a fairly scalable system without necessitating a lot of expensive head-end equipment (although it’s still a good idea for speed reasons). Typically, such networks are fairly easy to support, since if one node dies (unless it’s a hub itself or a server), nobody else is brought down. As you will see, this is not always the case with the other three topologies. Data is usually transmitted in broadcast fashion, with switches blocking the worst of it, but otherwise leaving nodes on specific segments to ferret out what data applies to them and which to ignore.

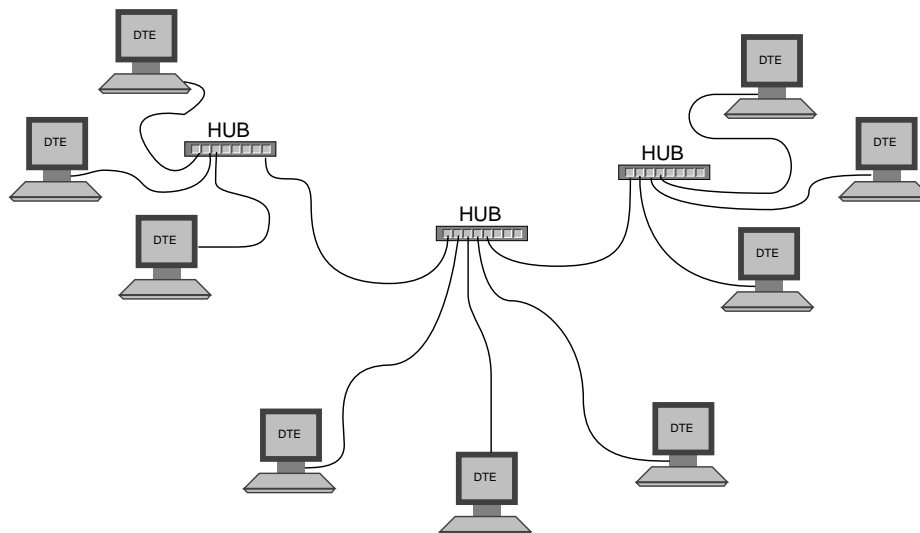


FIGURE 6.1.1: STAR TOPOLOGY

Ring topologies, made famous by IBM’s popular “token ring” implementation, are logically connected in big circles. One computer is connected to exactly two others (the “left” and “right” nodes) and the ends are joined in a

loop. Messages circulate around the ring in round-robin fashion, and usually there is a secondary “backup” ring which can also be used to carry data in the opposite direction.

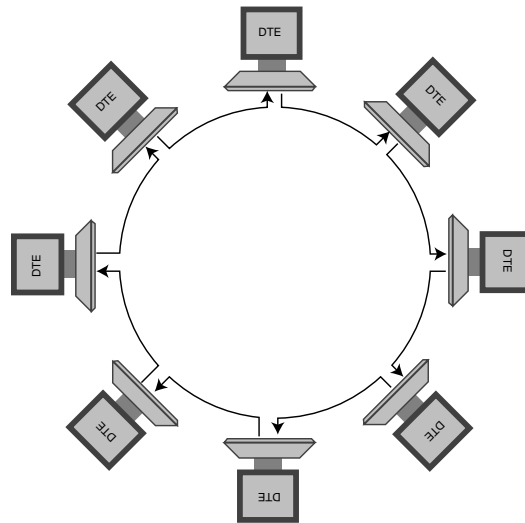


FIGURE 6.1.2: RING TOPOLOGY

Bus topologies have everything all layed out in a straight line (conceptually, anyway), with workstations and servers (or terminals and hosts) dropped off the main bus. Multiple busses can be bridged with extenders, although that affects attenuation and latency in much the same way as nested hubs eventually bog down stars.

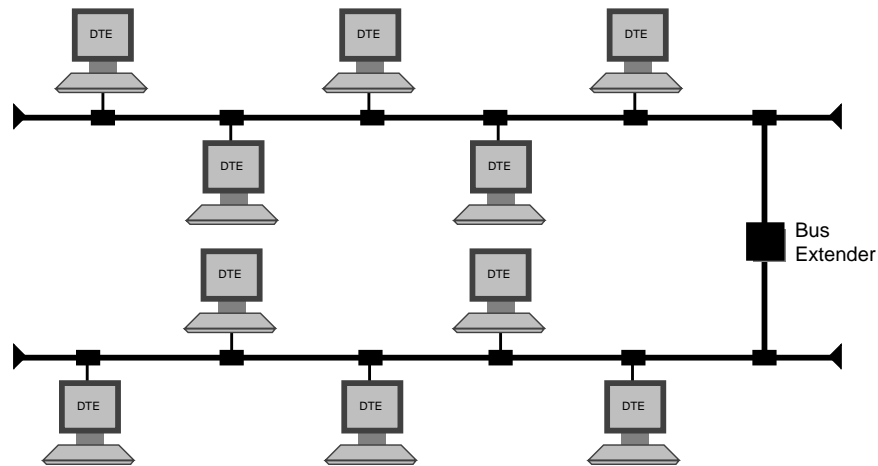


FIGURE 6.1.3: BUS TOPOLOGY

Hub or “tree” topologies aren’t unique layouts in the conceptual sense, because all they really do is mimic the operation of a bus or star network in a physical installation resembling a star. This is accomplished by sending two “homeruns” to each node rather than one, so that both the “left” and “right” connections of a bus or ring can be simulated.

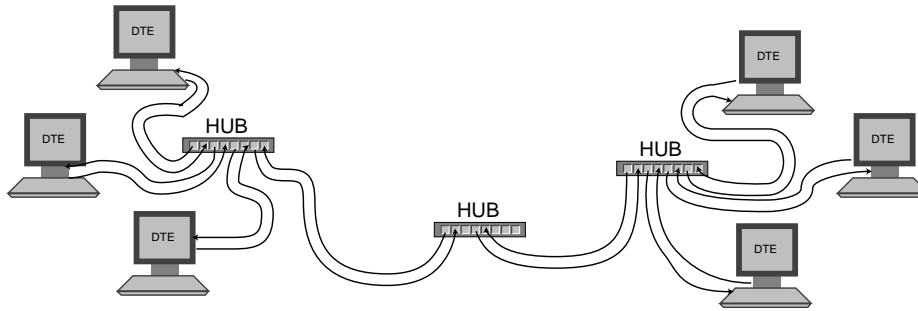


FIGURE 6.1.4: HUB/TREE TOPOLOGY

**6.4 Describe the principle of operation of the following MAC methods as used in LANs:**

- a. **CSMA/CD**
- b. **Control token**
- c. **Slotted ring**

CSMA/CD (Carrier Sensing Multiple Access/Collision Detection) is the MAC method I'm most familiar with, since it underlies the phenomenally successful ethernet layer which seems to dominate American business installations. The idea is that every connected node can detect ("sense") when there is no traffic on the shared line (ie, the default carrier signal is present). The catch, of course, is that once an opening appears on the line, several computers jump in at once and start broadcasting their own signal. This is where the "CD" (collision detection) comes in. Each unit listens on the line at the same time as they transmit, and can immediately detect when a jumble of overlapping messages collide. Such collisions occur more often than might be thought, because computers don't have to transmit literally simultaneously to cause one; the natural latency of a network allows one computer to start transmitting at one end of a network, and several "computer seconds" later another could start transmitting at the far end without having yet heard the first transmission begin. At any rate, once the competing computers have detected their collision, each transmitting computer gives up and agrees to wait a *random* interval of time before trying again. The random interlude is crucial so that their attempted re-entries onto the circuit occur at staggered intervals.

Control token networks use a different method of regulating access to a network line and avoiding collisions. A control token is a digital "marker" that gets passed around between computers. Only the computer currently in possession of the token is allowed to transmit, and once the computer has "handed off" the token to one-and-only-one peer machine, it has to shut up. This system requires at least one process on the network of being able to create a new token when the network is first brought up, or if the token somehow disappears and no one seems to have it. These are usually employed on ring and bus networks, whose physical "left-right" layout provides a simple and effective way to distribute the token.

Slotted rings are similar to token rings except that they have a number of control bits circulating at any one time. However, each unit holding a bit is only allowed to send a single frame, which fits into a rigidly formatted slot of available bandwidth. A monitor station initializes the network, keeps the number of control bits constant, and corrects any errors a DTE may commit (like failing to relinquish a slot). This is a somewhat slow but quite egalitarian way to share network access.

**6.10 Explain the operation of the CSMA/CD MAC method in relation to a hub wiring configuration.**

When a DTE wants to send data (a frame) over the shared medium, first it has to wait for a free interval on the cable. Even after other DTEs have stopped transmitting, it still pauses briefly in what's called an interframe gap to make sure that the recipient of the most recently transmitted frame has had time to digest and potentially prepare for another mouthful.

Once the DTE is confident that the line is clear and all ears are open, it begins transmitting its frame. The minimum frame size used is determined by the size of the network (end-to-end transmission time) to ensure that collisions are detected while DTEs are still transmitting and have the opportunity to cancel their colliding frames. If no collision is detected, the frame is sent upstream to the attached hub. Assuming the hub isn't of the new switching varieties, all it will do is take each frame received from one port and retransmit the same frame across every other port.

The repeated message will then be received by every other DTE hooked up to the hub. If hubs are "stacked" or daisy-chained, then computers linked to other hubs will also receive the frame, albeit a little later. Also, if hubs are nested in a tree fashion, all computers linked to downstream hubs will receive the frame. The only block to the

frame will be switches and routers which determine that the frame is of the “internal” variety and should propagate no further.

On the other hand, a collision could be detected while the source DTE is still transmitting to the hub. The source DTE detects such collisions by listening to the Rx line even while talking over the Tx line. If it hears something besides the carrier, it knows its local hub is at that moment sending another DTE’s frame downstream. When such a collision is detected, the source DTE (and presumably any other colliding DTE as soon as they hear the collision as well) will cancel their own transmission and start transmitting a special collision signal instead. Once everyone has heard the collision signal, they all stop transmitting and fall back to regroup. Each waits a random period before starting to transmit again. The random provision helps ensure that re-entries are interleaved. If the same terminal encounters several collisions in a row, it starts a “back off” sequence of doubling the wait-before-transmit duration, up to a predefined limit. That helps relieve sorely congested networks.

**6.11 Produce a schematic diagram showing the components necessary to attach a DTE (station) to a token ring network and give an outline of the function of each component. Include sufficient detail to show how a DTE, once attached to the network, may operate in either inserted or bypassed mode. Show also the location and function of a wiring concentrator.**

**MAC unit (Medium Access Control)** The MAC subsystem builds frames for transmit by the DTE, disassembles frames sent to the DTE, and relays other frames around the ring. It also handles initialization when the DTE is first hooked (inserted) into the ring, and provides diagnostic capabilities. All of these functions require buffering and a fair amount of intelligent circuitry.

**Concentrator** Concentrators provide the physical convenience of ethernet hubs on a token ring network. As with any device which sits on a token ring network, they have their upstream and downstream (“left” and “right”) connectors to maintain ring communications. However, they also provide a number of simple “homerun” ports allowing single-drop simplicity for adding new terminals and moving old ones. Homerun drops have both Tx and Rx pairs, allowing the ring to be extended by rolling “loops” out to individual desks and workstations. Concentrator ports provide the same features as TCUs, therefore, and unused ports can be left in “bypass” mode with the use of a simple loopback.

**Trunk Coupling Unit (TCU)** Trunk coupling units connect (couple) individual DTEs to a ring network. They have four conceptual ports: upstream and downstream ports for the ring, and Tx/Rx ports for the connected terminal. When placed in bypass mode, the upstream and downstream flows are joined so that information passes directly past the terminal. When the MAC activates the TCU and inserts the terminal into the ring’s data flow (as opposed to merely a physical presence), the TCU routes upstream signals into the terminal’s Rx port and sends the terminal Tx data into the downstream port. In this way all ring traffic is directed through the MAC unit.

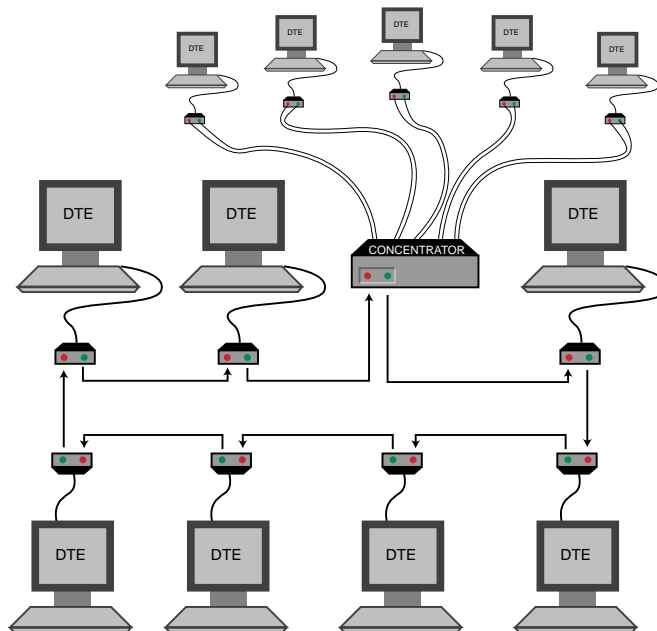
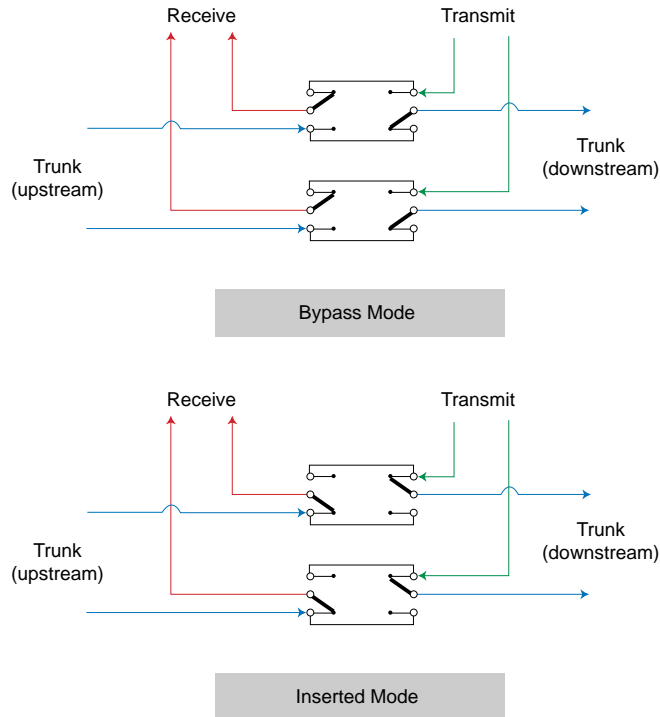


FIGURE 6.11.1

These are sample schematics showing how a TCU might be designed to allow either bypass or inserted operation:



**6.19 Discuss the influence of the following on the design of a radio receiver to be used with a wireless LAN: signal-to-noise ratio, thermal noise, signal bandwidth, range of coverage, transmitter power level.**

The signal-to-noise ratio represents the power of a received signal (wave amplitude or volume) to the mean level of background noise (interference, radiation, EM flux, etc). Every radio receiver is built to tolerate a degree of background noise, but the cost and complexity of circuitry to overcome and filter high noise levels means that higher tolerances to noise (hence low SNR ratio) cost more than low tolerances (high ratios) which don't provide much filtering technology.

Thermal noise is one source of background noise. The more heat a device generates, the "noisier" it becomes. Therefore, a low operating temperature is a desirable goal in devices utilizing radio reception. Actually, this is usually a good idea anyway, since radiant heat is usually an indication of wasted power. Radio units are generally portable devices running off battery packs, and if a device gives off a lot of heat that's a good indication that battery usage is less than efficient. As an example, the Thinkpad 760 I'm using to write this is currently hot enough that my 2yr-old son, when touching it, immediately withdraws his fingers and solemnly declares, "Hot!" (The laptop is also practically useless without an AC hookup, in contrast to the wonderfully efficient eMate 300, which, sadly, has a less readable display.)

Signal bandwidth is another factor in determining noise levels. Broadband applications are more susceptible to this form of noise than narrowband. Range of coverage is an everpresent issue, since signal power decreases at an inverse square to distance. Indoor distance is even more destructive than outdoor distance, because rather than simply travelling through the comparatively free airspace of pollution, smog, and storms, indoor signals must brave a littered course of coffeemakers, cubicle partitions, and staff meetings congested with antiquated ideas.

Of course, if you can't decrease the level of ambient noise, another option is simply to boost the transmitter power level. That works well for central servers broadcasting data to passive clients (ie, WJRR blasting *Ron and Ron* at millions of hapless Central Florida commuters each morning), but usually an option for two-way communication with portable clients. That is because broadcast power pulls from battery packs, which are perennially overburdened on portable units.



**6.20 Discuss the following in relation to the radio environment and wireless LANs:**

- a. adjacent channel interference and how it can be reduced
- b. multipath and its effects
- c. equalization
- d. directional antennas

Adjacent channel interference results from neighboring radio LAN segments operating on overlapping frequencies. Along the common boundary of the two coverage areas, signals from one zone may cross into the other zone and interfere with same-band transmissions in that segment. The easiest solution is to design the network with such issues in mind and simply not assign duplicate frequency zones to neighboring segments. As long as coverage areas are predictably uniform, it can be shown mathematically that 3 frequencies are sufficient to prevent overlap in a single floor. Assuming that most operating ranges are greater than the 10-15' which separate most office floors, network designers would also have to take into account vertical signal propagation as well. Signals wouldn't propagate well through cement floors, but they'd probably make some sort of intrusion. It would make an interesting mathematical study, although not one I have time to explore at the moment ☺ (Personally, I think a better approach would be to limit working hours between neighboring segments; ie allow the "Blue" team to work from 8am through noon, and the "Gold" team to work from lunch until dinner. This would also relieve congestion of other productivity-sensitive devices, such as photocopiers and parking spaces.)

Multipath refers to a problem specific to indoor networks, in which dispersed signals get bounced around differently in complex interiors. Some signals are fortunate enough to beam directly from the transmitter to the receiver, while others must first ricochet off a clock, bank off another terminal screen, and finally bounce off a co-worker's forehead before finally getting to the receiver. Since these two paths are of differing lengths, the signals can arrive out of phase with one another, especially if the co-worker has dry skin. One solution is to use two receivers, which between them can deduce the true signal from the noise.

Equalization is another way to deal with the multipath problem. In this technique, the receiver itself generates weak, simulated echos of the received signal which vary slightly in phase. Then the circuit subtracts those simulations from the received signal and, theoretically, removes actual disturbances which match the simulated reflections.

**6.22 Discuss the following relating to infrared systems:**

- a. Why adjacent channel interference is lower than with radio
- b. The need for optical filters
- c. The effect of the modulation bandwidth of the infrared source on the maximum bit rate that can be obtained

Within typical enclosed office environments, adjacent channel interference is less of a problem with infrared communication than with radio. This is because infrared, being very close to visible light, is subject to opacity and does not travel through walls. Therefore, identical frequencies can be used in adjacent rooms, as long as there are no doorways, pass-throughs, or one-way mirrors between them. (One-way mirrors take on all-new functionality with infrared wireless networks; while the boss thinks that he's spying on the employees, in fact they can tap into his network broadcasts while he can't receive any of theirs!)

Optical filters—both digital circuits and commonplace red gels—are useful in subduing light frequencies which fall outside of typical signal ranges. This is important since, with the exception of the Programmer Lestat, most work is done in lighted workplaces (either indoors or out), and most light sources, including Sol, produce a fair amount of infrared noise while performing their normal duties (like powering our calculators and ecology-friendly cars).

The bit-rate of infrared systems is directly related to the modulation bandwidth available. Laser diodes, which produce high-power signals in a limited frequency band of only a few nanometers, can provide very high bit-rates (enough for Fast Ethernet or ATM, for instance). Lower-powered (and lower-priced) LEDs, on the other hand, aren't as focused and spread their signals over a wide range of frequencies, and therefore cap out at ethernet speeds.

**6.28 With the aid of sketches, explain the principle of operation of both fast and slow frequency-hopping spread spectrum systems. Clearly identify the information that is being transmitted on each carrier when it is active with each scheme and how the receiver determines the transmitted bit sequence.**

Both fast and slow frequency-hopping spread spectrum transmission systems use a semi-random algorithm to hop between different frequencies in mid-transmission. The difference lies in whether several frequencies are used to transfer a single bit (fast), or several bits are transmitted within a single frequency (slow). Figure 6.28.1 shows the fast method and figure 6.28.2 shows the slow method.

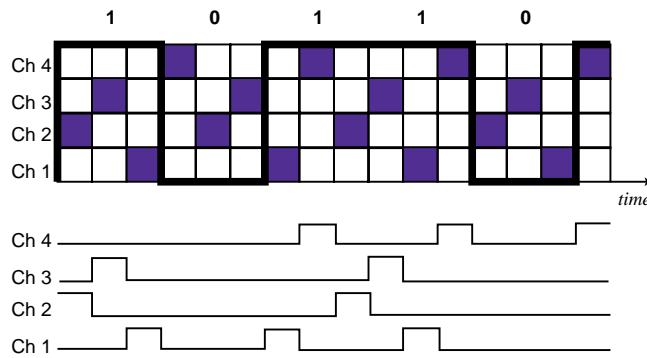


FIGURE 6.28.1: FAST FREQUENCY HOPPING

In each case, four frequencies are used (1-4) and the freq-hopping algorithm specifies a repeated sequence of 2-3-1-4. The fast method sends uses three hops for each bit, while the slow method sends three bits for each hop. As you can see, the individual channels in the fast method more closely resemble true noise, and also provide three samples to ascertain the value of each bit. However, fast hopping requires more intelligent and henceforth expensive hardware.

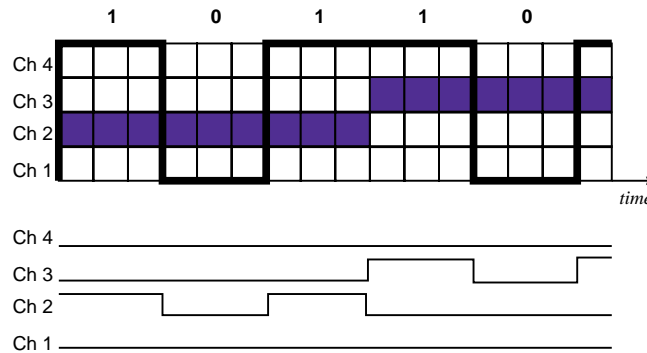


FIGURE 6.28.2: SLOW FREQUENCY HOPPING

**6.40 Given the graphs of Figure 6.23(a) and (b), explain the relative performance of a CSMA/CD, token bus, and token ring LAN.**

According to the graphs on page 316 of the textbook, neither CSMA/CD nor token ring are especially sensitive to frame size. CSMA/CD is an efficient and responsive system on networks with less than 50% throughput, whether the frame size is 512 bits or 12000. Likewise, token ring seems to hold up well under loads of up to 80%, irrespective of frame size. Token bus, on the other hand, shows a significant improvement in performance by increasing the frame size. The most likely conclusion to draw from the charts is that under high loads, token-based systems do better because there is a management system specifying how to best exploit limited resources. The more "capitalistic" ethernet, meanwhile, destroys itself in a frenzied grab for bandwidth. An interesting analogy could be drawn to planned economies versus free enterprise, but I doubt I would like the conclusion, and anyway, people aren't bits and efficiency isn't always the highest value ☺

## Chapter 7: High Speed and Bridged LANs

**7.2 Produce a schematic diagram of a switching hub. Explain its operation including the role of the FIFO buffers and routing table. Why is a separate collision detect line required for each DTE?**

The FIFO buffers are provided to give the switch a "holding area" to store a frame long enough to briefly analyze the header and decide if and where to send it. Each time a DTE or server communicates over a port, the incoming frame is copied into the FIFO buffer for that port. If the ethernet address of the DTE connected to that port is not known, the "source" address is copied from the frame header and stored in a routing table. Then the

switch checks the “destination” address against the routing table to see if it knows which port is assigned to that computer. If it finds a match, and no other frame is currently being sent to that port, it sends the packet just down that path (in the case of a potential collision, it activates the “CD” line back to the sending terminal, but *just* that terminal). If it doesn’t find a match but has “unknown” ports, it broadcasts the frame to all unknown ports, including other hubs. If it knows the address of every attached terminal and still can’t find a match, it assumes the frame belongs elsewhere on the network and sends it to other hubs (if available).

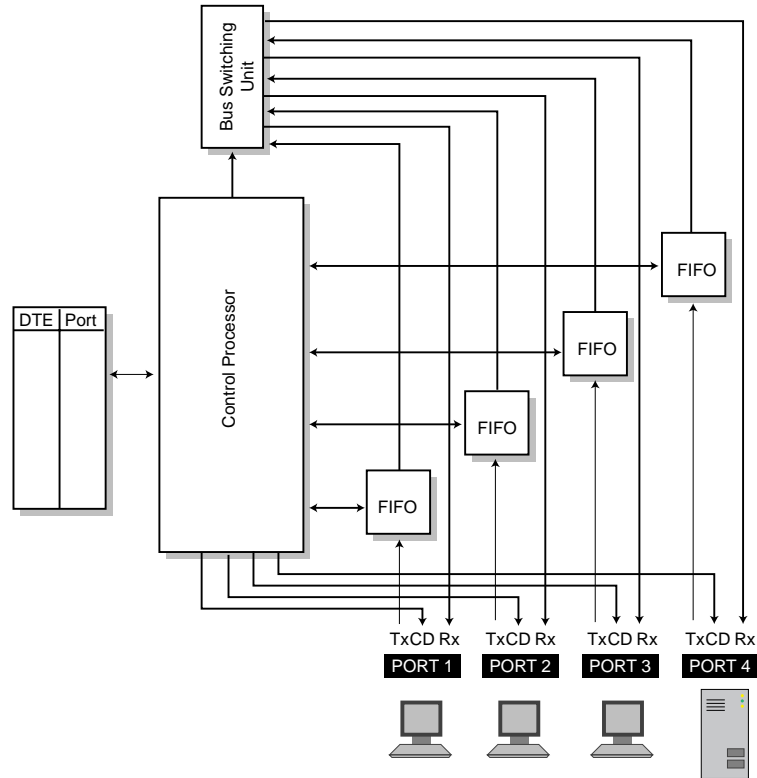


FIGURE 7.2.1

**7.4 Produce a sketch of a single-level 100VG AnyLAN (IEEE 802.12) network. A single source DTE makes a request to send a frame over this network. Use a sketch to show signals that are generated by the hub repeater (and all the other DTEs that are attached to the hub) as a result of the request. Also show the signals that are generated during and after frame transmission.**

This question didn't seem to request any actual text *per se*, but I figure I'd better explain some of the diagrams that follow © Figure 7.4.1 shows a simple 3-station 100VG AnyLAN network...not very impressive, really. The protocol really comes into its own linking multiple segments from various vendors, of course, but that would require much more drawing.

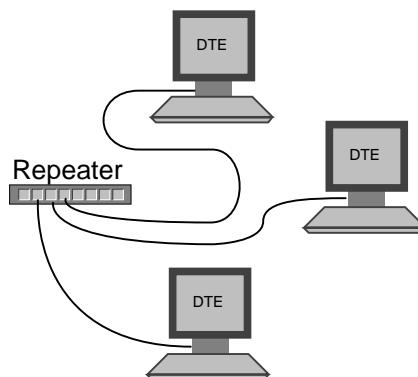


FIGURE 7.4.1

Figure 7.4.2 shows the sequence of requests issued between clients as one DTE (the source) sends a frame to a second DTE (the destination) across the repeater. Meanwhile, three peer stations stand by helplessly and watch, unable to do anything while the repeater is otherwise occupied.

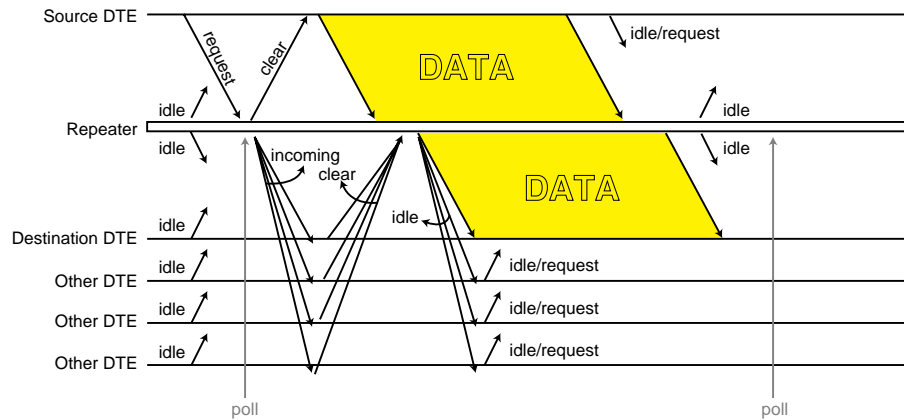


FIGURE 7.4.2

Figure 7.4.3 is the most interesting one. This shows a simulation of the decision-making process that goes on inside the repeater while receiving normal and high-priority frame transmission requests from four clients. In the first poll ( $t_1$ ), a normal request and a high-priority request are detected on ports 2 and four, respectively. The repeater handles the high-priority request, and while doing so receives a request for a normal frame delivery. Since the NNPP is still holding at its initialized value of "1", the normal frame on port 1 is handled before the frame on port 2, even though port 2 asked first. The simulation continues with several additional HP and N requests to demonstrate the repeater's prioritized round-robin algorithm.

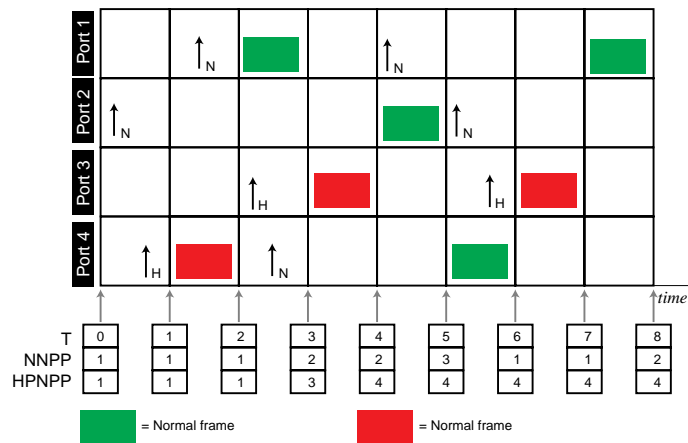


FIGURE 7.4.3

**7.8 With the aid of a diagram, describe the meaning of the following components relating to an FDDI network:**

- a. **Primary and secondary ring**
- b. **Single attach and dual attach station**
- c. **Optical coupling unit**
- d. **Wiring concentrator**
- e. **Polarized duplex connectors**

The primary ring is the main traffic route holding all the traffic which circulates around the network in some arbitrary direction (let's say clockwise). The secondary ring is provided as a backup in case something really nasty happens to the network medium—say, a construction worker rips through an underground backbone run between buildings with a backhoe. Since the secondary ring runs in the opposite direction (for instance, counter-clockwise),

data is still able to move back and forth around the ring—if monitor stations are available in both buildings, the network could even continue to operate as two unconnected segments if the loop is cut in two places! Some implementations also use the secondary ring as an additional data path during normal affairs as well. Dual attach stations are those which are hooked to both the primary and secondary rings, while single-attach units are hooked only to the primary ring (making them, paradoxically, second-class network citizens).

When a unit needs to be placed in bypass mode (temporarily disconnected from the ring), an optical coupling unit is used. This is basically a loopback connector to create a closed circuit on the ring. Other interesting components you might find in a FDDI technician's toolbox include polarized duplex connectors, which are keyed connectors you (carefully) attach onto the end of fiber strands. They're keyed so that you can't accidentally plug an Rx cable into a Tx jack, but in practice you can just look at the cable end-wise to see if it's got a red light showing (that's usually the Tx end). Of course, that would involve bringing down the network, so keyed connectors probably aren't such bad ideas after all.

Finally, wiring concentrators are used to allow FDDI logical-rings to be physically laid out in a star topology. Although this doubles cable runs (a little expensive with fiber), it allows a highly convenient, structured, and above all maintainable wiring cabinet with patch-cord simplicity. Wiring concentrators have the same upstream/downstream ports as other devices attached to the ring, but the concentrator then loops them in and out of a series of terminal ports for use in homerun installations. Unused ports are just patched together to maintain the ring.

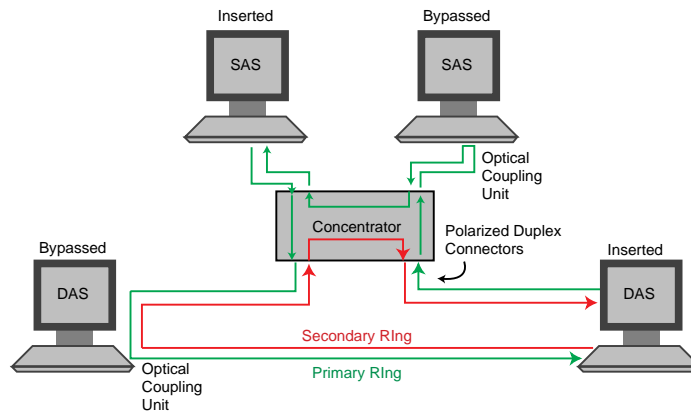


FIGURE 7.8.1

**7.13 Explain the meaning of the following terms relating to bridged LANs:**

- a. **Bridge**
- b. **Multipoint bridge**
- c. **Building backbone**
- d. **Establishment backbone**

A bridge is a repeater with brains. Instead of forwarding every packet it receives, it looks at the frame header and uses an (address,port) lookup table to decide where the frame should go (like a switching hub). However, bridges are also able to link segments using different MAC protocols (unlike a hub) and convert frames structures when necessary. A multipoint bridge is a bridge that connects more than two segments. Building and establishment backbones are, in my mind, contrived and subjective terms that don't convey much information about scale and scope, other than the fact that establishments are typically bigger than buildings (unless you work in a leased prewired floor of a commercial office building, in which your entire corporate enterprise represents only one segment of the building backbone—it happens). However, the intended concept is that a building backbone connects a handful of local devices, bridges, and hubs, while the enterprise backbone connects other backbones, typically with routers and other WAN gear.

**7.14 List and discuss the advantages and disadvantages of bridges relative to a repeater.**

Advantages	Disadvantages
<ul style="list-style-type: none"> <li>• Very scalable—you can add segments almost <i>ad infinitum</i></li> <li>• Can relay frames across different protocols</li> <li>• SNMP management</li> <li>• Segmentation improves uptime, response, and</li> </ul>	<ul style="list-style-type: none"> <li>• Store-and-forward process slows down relayed frames somewhat</li> <li>• Bridge can overload during peak traffic</li> <li>• Additional potential for error</li> </ul>

almost everything else that's good and noble and true. introduced

**7.15 Produce a sketch of a typical bridge architecture and, with the aid of an example, describe its principle of operation. Include in your description the following terms:**

- a. Promiscuous mode**
- b. Forwarding database**
- c. Bridge learning**
- d. Spanning tree**

Promiscuous mode merely refers to a multiport bridge's ability to communicate with more than two ports. This is shown in the accompanying illustration in Bridge #1. The forwarding databases are also provided for each bridge, which would have been filled in automatically via bridge learning. Bridge learning is a bridge's ability to remember which frames have come in on which ports from which source addresses. Those records can later be used to direct frames being sent to those addresses onto the proper port. The sample network shown in the diagram is a spanning tree, which means that there is one-and-only-one path between any two nodes.

The operation of a bridge is simple enough. It receives a packet via one of its ports, which presumably had been broadcast to all nodes on an ethernet or sent spinning 'round the ring on a token architecture. It's the bridge's job to determine whether the frame belongs on its originating segment or somewhere else. If the destination address matches an entry in its database for the originating port, then the bridge disregards the frame, as the intended recipient will receive it automatically through the same mechanism as the bridge did. Or the destination address may appear in its database assigned to another port, in which case the bridge will send the packet out onto that segment (performing frame conversion if appropriate). Otherwise, it will have no idea where the frame should go and will send it out onto every port in the hope that it will end up at the target DTE somehow. Whenever the bridge decides to forward a frame, it provides the added courtesy of performing a checksum scan to make sure the frame hasn't been corrupted.

**7.25 With the aid of a sketch showing the format of the different basic LAN types, discuss the issues involved in producing a bridged LAN comprising the different LAN segment types.**

The two major LAN types to be bridged are CSMA/CD ethernet and token ring. The two main considerations would be making sure that frames could be successfully passed from token ring to ethernet, and vice versa. A visual aid is provided in figure 7.25.1.

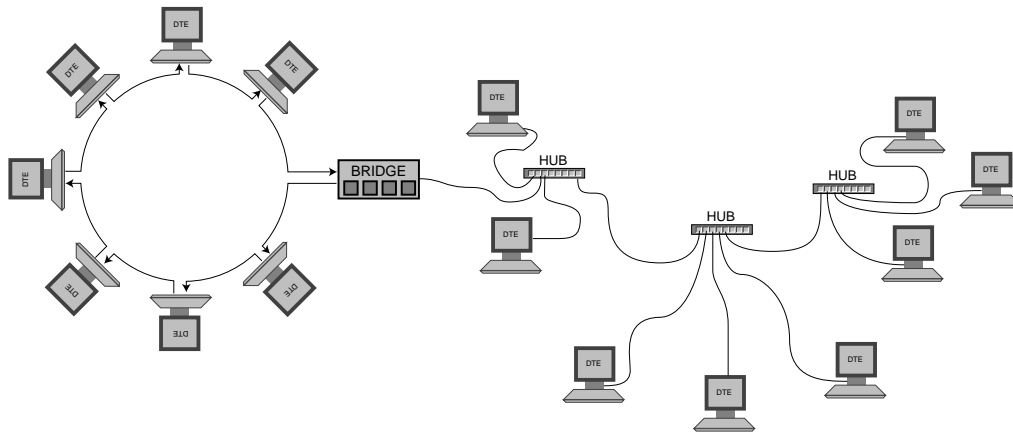


FIGURE 7.25.1

Consider the token ring→ethernet example. The source station on the token ring would generate a frame containing the destination address and start sending that around the ring. Each node on the ring would inspect the packet header and decide whether the frame was intended for them or not. The bridge could be hooked onto the ring just like any other device. When the frame made it to the bridge, the bridge would disassemble the frame and analyze the destination address. If the bridge had already received other frames from the destination address on the token ring port, it could assume that the destination DTE was another node on the ring and that the bridge could therefore discard the frame and do nothing. On the other hand, if the bridge either had a listing in its routing table indicating that the destination DTE was in fact on the ethernet segment, or if the bridge didn't recognize the address at all, then it would have to move the frame over to the ethernet segment. The main work involved here is pulling

out the various header and data fields from the token ring frame and reassembling them into a new ethernet frame. Finally, the ethernet frame is send out through the appropriate port (avoiding collisions in the usual manner), whereupon it will be detected by the nearest hub and promogulated throughout the ethernet segment. The reverse action would take place in pretty much the same way.

## Chapter 8: Wide Area Networks

### 8.1 Describe the differences between a circuit switched data network and a packet switched data network. Clearly identify the effects on the users of these networks.

Circuit-switched data networks, like the PSTN, provide a dedicated physical connection between two communicating DTE's. However, it can take a while to establish that connection (often incurring fees for each successful establishment), so once a connection is created it makes sense to hold onto it for awhile. The connection established will include a maximum bit rate for data transfer. Although that rate may be smaller than available through all-digital networks, at least it is guaranteed and won't be shared or subject to random variations. Also, all a CSDN does is pipe data; it doesn't check for errors or provide any of the other "value-added" services some people have come to expect of modern networks.

Packet-switched data networks, on the other hand, operate essentially like extended computer networks. No single dedicated path is provided between DTE's, other than a basic assurance that at least one such path exists and that data will be routed along it or similar routes. Since the communication medium is shared along most of the communication path, frames are assembled into packets and transmitted serially. As with computer networks, packets don't always arrive in the order they were sent, so PSDN's provide a facility for rearranging jumbled packets back into the appropriate sequence before extracting and relaying the embedded frames. Also, the PSDN checks for errors that might have been introduced during transmission using a suite of error-correction procedures. However, the data rate provided by PSDNs is a little unpredictable. Although it can be very high, it can also drop unexpectedly due to congestion from other users, something that just doesn't happen with circuit-switched networks.

Due to these factors, users should decide which sort of network makes the most sense for their particular application. If a lot of data is to be transfered, and total throughput matters more than the speed of any particular chunk, the PSDN is usually the way to go. Likewise, if scalability is a concern, I think that the PSTN makes it a lot easier to gradually add bandwidth. On the other hand, if the small-but-stable reliability of physical connections is desirable, CSDN's still have a lot of life left in them, especially for welterweight applications.

### 8.3 Use sketches to illustrate the applicability and components of the X.25 network access protocol and write explanatory notes describing the function of each component.

**DTE (Data Terminal Equipment)** These are the terminals communicating over the PSPDN. Typically, one will be a PC or workstation, while the other will be a server or host. With respect to any given transmission, one will be the primary (source) and the other will be the secondary (recipient).

**DCE (Data Circuit Termination Equipment)** Also called terminal adapters, these are supplied by the local PTT (Post, Telephone, and Telecommunications authority). They act much like modems, providing a duplex serial synchronous path from the DTE to the local PSE at rates comparable to ISDN.

**PSE (Packet Switch Exchange)** This is conceptually like a NIC, in that it connects a single terminal to a network, but it functions more like a router, with all of the intelligence and optimization algorithms that entails. It accepts packets from the DCE into a buffer and extracts the destination address. This address is compared against a routing directory to determine which link the outgoing packet should use. Incoming packets are handled in much the same way. The routing table suggests whether the incoming packet is intended for the attached DTE, or if it should be promogulated further through the PSPDN to the next link in its path.

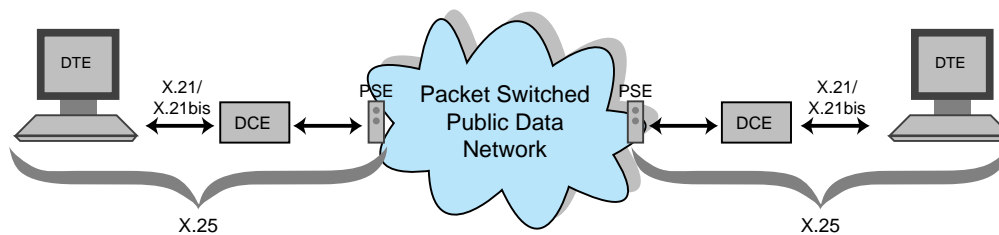


FIGURE 8.3.1

- 8.14 a. Outline the function of the three lowest network-dependent protocol layers used with a circuit switched data network  
 b. Sketch a diagram showing the various interchange circuits associated with X.21 and outline their functions

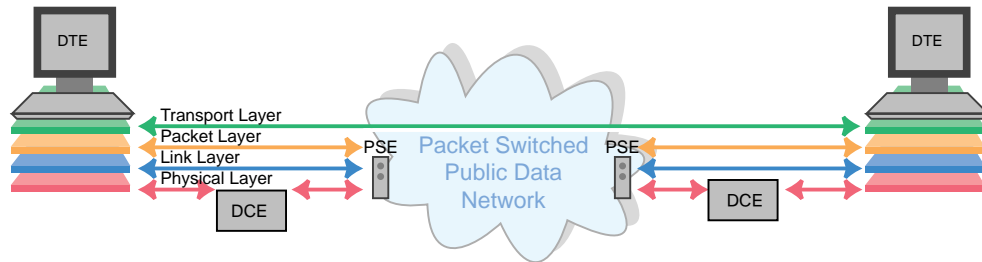


FIGURE 8.14.1

**Physical Layer** This is defined by ITU-T recommendation X.21/X.21bis, detailed below.

**Link Layer** The purpose of the link layer is to provide the packet layer with error-free, unduplicated packet transport, across the physical layer, between the DTE and the PSE, using HDLC (LAPB).

**Packet Layer** The packet layer provides six services and a number of primitives to the transport layer, which in turn deals with the actual message units (TPDUs, or transport packet data units). The packets used by the packet layer protocol (PLP) are called PPDU (packet protocol data units).

These are the services and primitives provided by the packet layer to the transport layer:

N_CONNECT	.request .indication .response .confirm	N_RESET	.request .indication .response .confirm
N_DATA	.request .indication	N_EXPEDITED_DATA	.request .indication
N_DATA_ACKNOWLEDGE	.request .indication	N_DISCONNECT	.request .indication

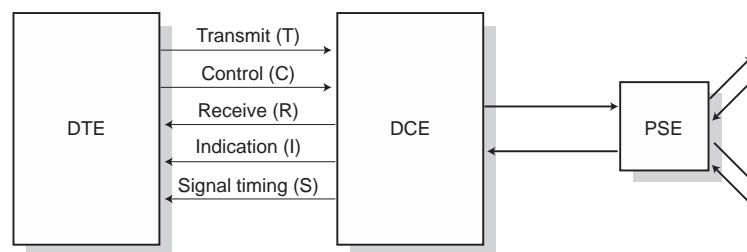


FIGURE 8.14.2

Both the DTE and the DCE physically reside with the user. Figure 8.14.2 shows the data circuits that connect the two units with a serial cable. Separate lines are reserved for transmit and receive, permitting duplex transmission. A control path is provided from the DTE to the DCE for issuing requests and other primitives, while an indication path exists for the DCE to return the results of requests. Finally, it is noteworthy that the timing circuit flows from the DCE to the DTE, and not the other way around.

- 8.15 a. Explain the meaning of the term 'ISDN.'

ISDN refers to a Integrated Services Digital Network, which is a relatively quick all-digital interface to the PSTN. Many PTTs are in the processes of converting their networks to ISDN, since the all-digital format allows a much



more flexible and controlled use of available bandwidth. ISDN has four standard bearer channels, plus newer features such as fax, videotex, etc. An NTE (network termination equipment) or terminal adapters are used to connect DTEs to ISDN networks. The standard ISDN package, Basic Rate Interface (BRI), has two 64K B-channels for data and one 16K D-channel for control information. Most subscribers use one B-channel for voice and the other for data, although "bonded" BRI allows both B-channels to be combined for a 128K data channel. Some new proposals even allow folding the D-channel into that line for a little extra kick.

**8.20 Explain the principle of operation of frame relay and how it differs from X.25 packet switching. Include in your description how frames are used.**

Frame relay requires establishment of a virtual path, similar to an X.25 switched circuit. Semipermanent connections analogous to leased lines are also supported. The virtual path is first initialized, and then held for the duration of the connection so that frames can be pumped right through in rapid-fire queue. A best-try scheme is used without error or flow control, so transmission is pretty fast (faster than packet switching, for instance, which incurs a higher overhead). Although frame relay is technically an ISDN specification, it is now commonly used over private networks for high-speed internal transit. The functionality of individual protocol layers also differs: whereas X.25 performs multiplexing of virtual circuits in the packet layer and provides error control of the DTE-DCE path in the link layer, frame relay does all of its multiplexing and routing on the link layer, for a significantly faster overall connection.

Frame relay frame headers have no control field, and need none, since no error or flow processing is performed. However, they do contain a data link connection identifier (DLCI), which identifies the local downstream link path. (This value is set by each link node and is of significance only within a specific node.) Frame relay frames also have interesting congestion control bits which can be used to inform end-users of pending network conditions. There are both Forward and Backward Explicit Congestion Notification bits (FECN and BECN, or just CF and BF) to tell downstream (forward) and upstream (backward) users that a queue-crunch is developing. This information can be used by the end stations to either slow down packet streams, or possibly avail themselves of the Discard Eligibility (DE) bit which indicates that a packet can be dropped without catastrophic results.