ANSWER 1 :

AS-4225

B. Tech. (Seventh Sem.) Exam - 2013

(ECE Branch)

Data communication

(ECE 4105)

Solution of ESE Q. Paper

Ans. 1 ⇒ (i) Transmission in only one direction. (TV remote)

(ii) 100 Hz - 5 MHz , 100 KHz - 500 MHz

(iii) AMI → Alternate Mark Inversion

HDB3 → High Density Bipolar - 3

(iv) RZ - 3 levels

Bipolar - 3 levels

(v) (d) None of these

(vi) (a) More cabling than tree, ring & bus

(b) Problem in hub will stop transmission.

(vii) (c) Control Unit

(viii) Data link connection Identifier (DLCI)

Forward Explicit congestion Notification (FECN)

(ix) (d) All of the Above

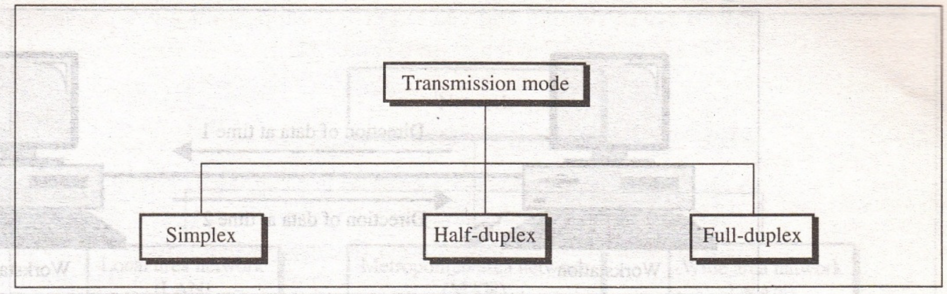
(x) FDDI ⇒ fiber Distributed Data Interface

ATM ⇒ Asynchronous Transfer mode.

ANSWER 2 : TRANSMISSION MODES

The term *transmission mode* is used to define the direction of signal flow between two linked devices. There are three types of transmission modes: *simplex*, *half-duplex*, and *full-duplex* (see Figure 2.11).

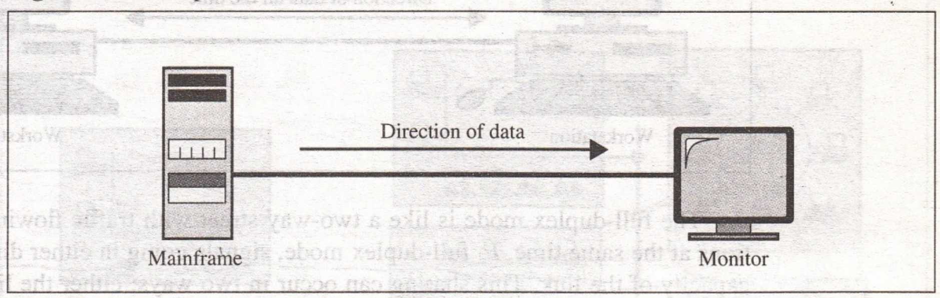
Figure 2.11 *Transmission modes*



Simplex

In **simplex mode,** the communication is unidirectional, as on a one-way street. Only one of the two stations on a link can transmit; the other can only receive (see Figure 2.12).

Figure 2.12 *Simplex*



The term *transmission mode* refers to the direction of information flow between two devices.

Keyboards and traditional monitors are both examples of simplex devices. The keyboard can only introduce input; the monitor can only accept output.

Half-Duplex

In **half-duplex mode,** each station can both transmit and receive, but not at the same time. When one device is sending, the other can only receive, and vice versa (see Figure 2.13).

The half-duplex mode is like a one-lane road with two-directional traffic. While cars are traveling one direction, cars going the other way must wait. In a half-duplex transmission, the entire capacity of a channel is taken over by whichever of the two devices is transmitting at the time. Walkie-talkies and CB (citizen's band) radios are both half-duplex systems.

Full-Duplex

In **full-duplex mode** (also called **duplex**), both stations can transmit and receive simultaneously (see Figure 2.14).

Figure 2.13 *Half-duplex*

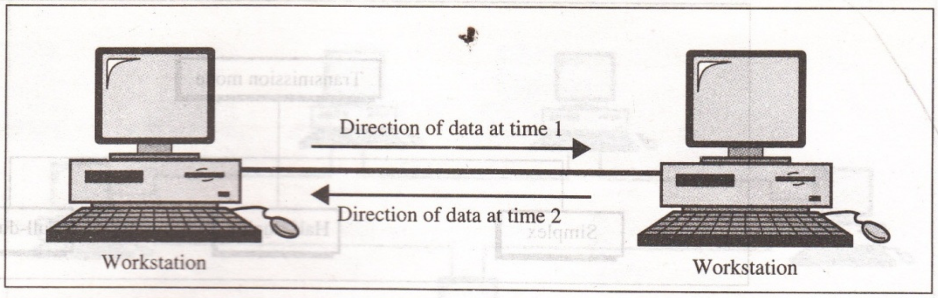
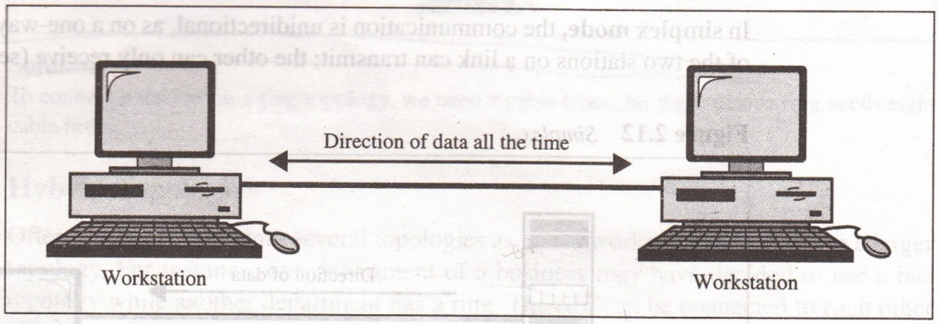


Figure 2.14 *Full-duplex*



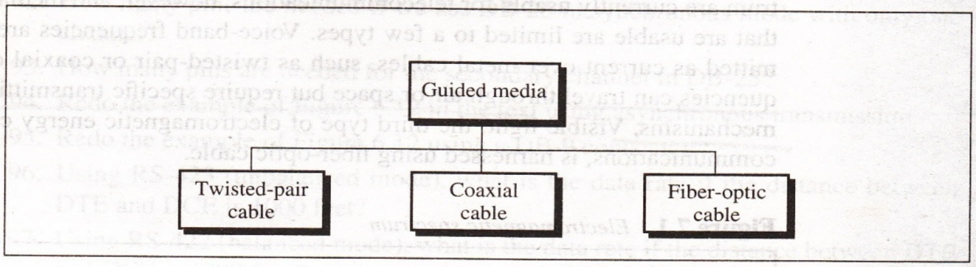
The full-duplex mode is like a two-way street with traffic flowing in both directions at the same time. In full-duplex mode, signals going in either direction share the capacity of the link. This sharing can occur in two ways: either the link must contain two physically separate transmission paths, one for sending and the other for receiving, or the capacity of the channel is divided between signals traveling in opposite directions.

One common example of full-duplex communication is the telephone network. When two people are communicating by a telephone line, both can talk and listen at the same time.

ANSWER 3: GUIDED MEDIA

Guided media, which are those that provide a conduit from one device to another, include **twisted-pair cable, coaxial cable,** and fiber-optic cable (see Figure 7.3). A signal traveling along any of these media is directed and contained by the physical limits of the medium. Twisted-pair and coaxial cable use metallic (copper) conductors that accept and transport signals in the form of electrical current. **Optical fiber** is a glass or plastic cable that accepts and transports signals in the form of light.

Figure 7.3 *Categories of guided media*



Twisted-Pair Cable

Twisted-pair cable comes in two forms: unshielded and shielded.

Unshielded Twisted-Pair (UTP) Cable

Unshielded twisted-pair (UTP) cable is the most common type of telecommunication medium in use today. Although most familiar from its use in telephone systems, its frequency range is suitable for transmitting both data and voice (see Figure 7.4). A twisted pair consists of two conductors (usually copper), each with its own colored plastic insulation. The plastic insulation is color-banded for identification (see Figure 7.5). Colors are used both to identify the specific conductors in a cable and to indicate which wires belong in pairs and how they relate to other pairs in a larger bundle.

A twisted pair consists of two conductors each surrounded by an insulating material.

Figure 7.4 *Frequency range for twisted-pair cable*

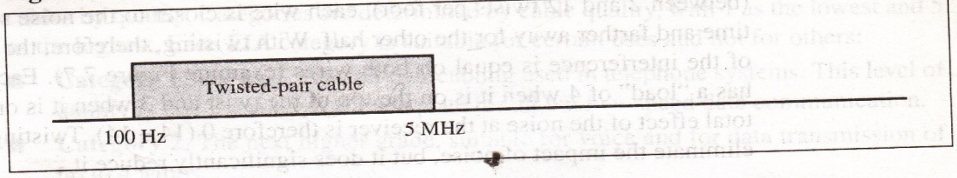
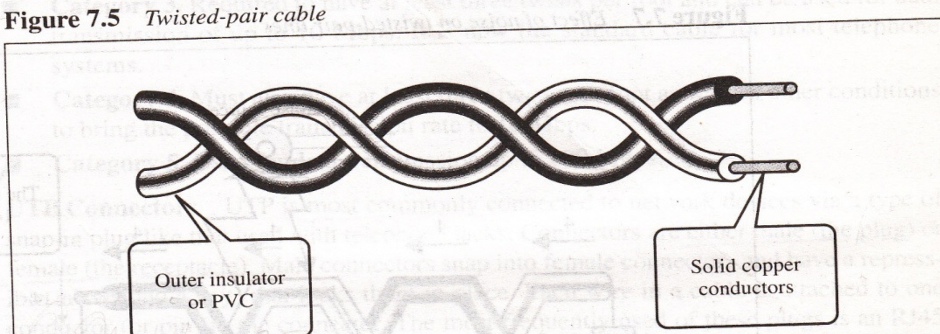


Figure 7.5 *Twisted-pair cable*

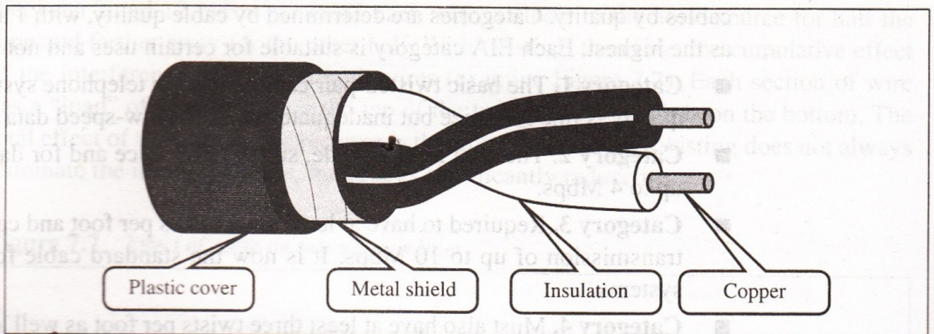


In the past, two parallel flat wires were used for communication. However, electromagnetic interference from devices such as a motor can create noise over those wires. If the two wires are parallel, the wire closest to the source of the noise gets more interference and ends up with a higher voltage level than the wire farther away, which results in an uneven load and a damaged signal (see Figure 7.6).

Shielded Twisted-Pair (STP) Cable

Shielded twisted-pair (STP) cable has a metal foil or braided-mesh covering that encases each pair of insulated conductors (see Figure 7.10). The metal casing prevents the penetration of electromagnetic noise. It also can eliminate a phenomenon called **crosstalk,** which is the undesired effect of one circuit (or channel) on another circuit (or channel). It occurs when one line (acting as a kind of receiving antenna) picks up some of the signals traveling down another line (acting as a kind of sending antenna). This effect can be experienced during telephone conversations when one can hear other conversations in the background. Shielding each pair of a twisted-pair cable can eliminate most crosstalk.

Figure 7.10 *Shielded twisted-pair cable*

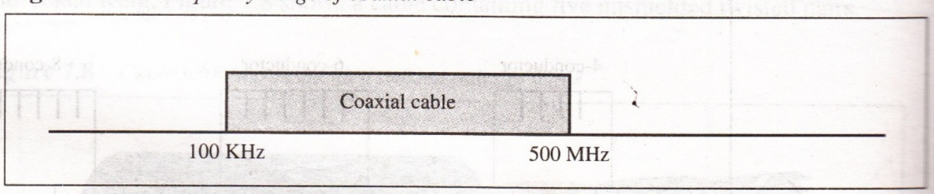


STP has the same quality considerations and uses the same connectors as UTP, but the shield must be connected to a ground. Materials and manufacturing requirements make STP more expensive than UTP but less susceptible to noise.

Coaxial Cable

Coaxial cable (or *coax*) carries signals of higher frequency ranges than twisted-pair cable (see Figure 7.11), in part because the two media are constructed quite differently. Instead of having two wires, coax has a central core conductor of solid or stranded wire (usually copper) enclosed in an insulating sheath, which is, in turn, encased in an outer conductor of metal foil, braid, or a combination of the two (also usually copper). The outer metallic wrapping serves both as a shield against noise and as the second conductor, which completes the circuit. This outer conductor is also enclosed in an insulating sheath, and the whole cable is protected by a plastic cover (see Figure 7.12).

Figure 7.11 *Frequency range of coaxial cable*



ANSWER 4: POLAR ENCODING TECHNIQUES

Polar

Polar encoding uses two voltage levels: one positive and one negative. By using both levels, in most polar encoding methods the average voltage level on the line is reduced and the DC component problem of unipolar encoding is alleviated. In **Manchester** and **differential Manchester encoding** (see page 97), each bit consists of both positive and negative voltages, so the DC component is totally eliminated.

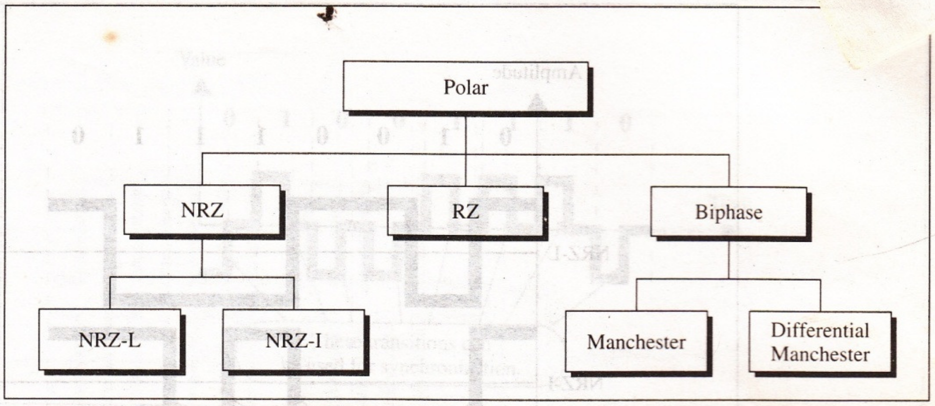
Polar encoding uses two levels (positive and negative) of amplitude.

Of the many existing variations of polar encoding, we will examine only the three most popular: **nonreturn to zero (NRZ), return to zero (RZ),** and **biphase.** NRZ encoding includes two methods: **nonreturn to zero, level (NRZ-L),** and **nonreturn to zero, invert (NRZ-I).** Biphase also refers to two methods. The first, Manchester, is the method used by ethernet LANs. The second, Differential Manchester, is the method used by Token Ring LANs (see Figure 5.5).

Nonreturn to Zero (NRZ)

In NRZ encoding, the level of the signal is always either positive or negative. The two most popular methods of NRZ transmission are discussed below.

Figure 5.5 *Types of polar encoding*



NRZ-L In NRZ-L encoding, the level of the signal depends on the type of bit it represents. A positive voltage usually means the bit is a 0, and a negative voltage means the bit is a 1 (or vice versa); thus, the level of the signal is dependent upon the state of the bit.

> In NRZ-L the level of the signal is dependent upon the state of the bit.

A problem can arise when there is a long stream of 0s or 1s in the data. The receiver receives a continuous voltage and should determine how many bits are sent by relying on its clock, which may or may not be synchronized with the sender clock.

NRZ-I In NRZ-I, an inversion of the voltage level represents a 1 bit. It is the transition between a positive and a negative voltage, not the voltages themselves, that represents a 1 bit. A 0 bit is represented by no change. NRZ-I is superior to NRZ-L due to the synchronization provided by the signal change each time a 1 bit is encountered. The existence of 1s in the data stream allows the receiver to resynchronize its timer to the actual arrival of the transmission. A string of 0s can still cause problems, but because 0s are not as likely, they are less of a problem.

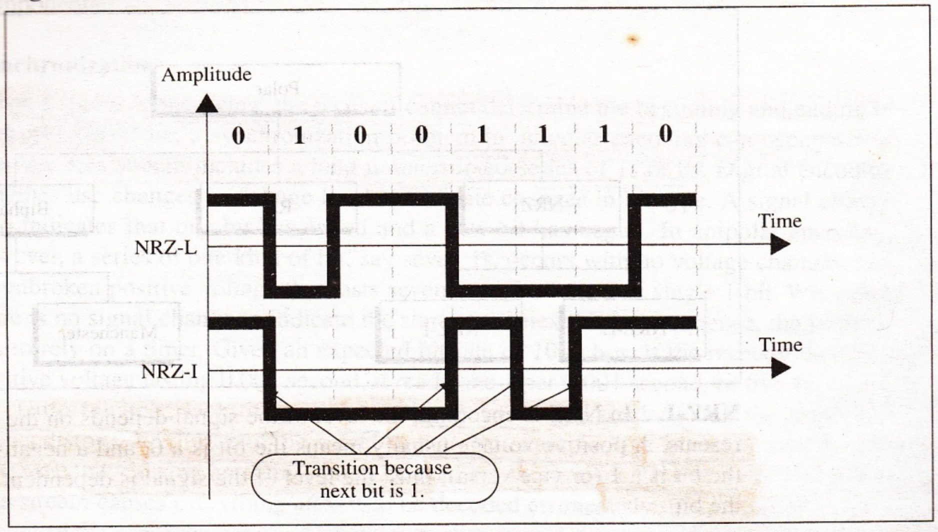
> In NRZ-I the signal is inverted if a 1 is encountered.

Figure 5.6 shows the NRZ-L and NRZ-I representations of the same series of bits. In the NRZ-L sequence, positive and negative voltages have specific meanings: positive for 0 and negative for 1. In the NRZ-I sequence, the voltages per se are meaningless. Instead, the receiver looks for changes from one level to another as its basis for recognition of 1s.

Return to Zero (RZ)

As you can see, anytime the original data contain strings of consecutive 1s or 0s, the receiver can lose its place. As we mentioned in our discussion of unipolar encoding, one

Figure 5.6 *NRZ-L and NRZ-I encoding*



way to assure synchronization is to send a separate timing signal on a separate channel. However, this solution is both expensive and prone to errors of its own. A better solution is to somehow include synchronization in the encoded signal, something like the solution provided by NRZ-I, but one capable of handling strings of 0s as well as 1s.

To assure synchronization, there must be a signal change for each bit. The receiver can use these changes to build up, update, and synchronize its clock. As we saw above, NRZ-I accomplishes this for sequences of 1s. But to change with every bit, we need more than just two values. One solution is return to zero (RZ) encoding, which uses three values: positive, negative, and zero. In RZ, the signal changes not between bits but during each bit. Like NRZ-L, a positive voltage means 1 and a negative voltage means 0. But, unlike NRZ-L, halfway through each bit interval, the signal returns to zero. A 1 bit is actually represented by positive-to-zero and a 0 bit by negative-to-zero, rather than by positive and negative alone. Figure 5.7 illustrates the concept.

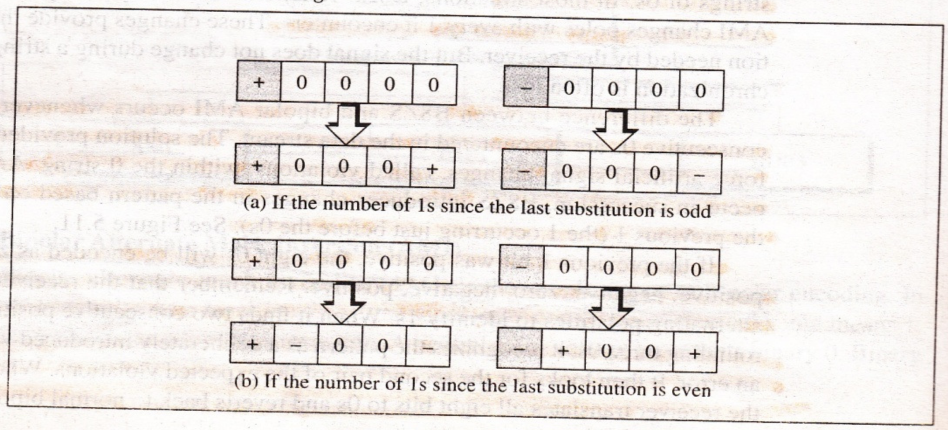
The main disadvantage of RZ encoding is that it requires two signal changes to encode one bit and therefore occupies more bandwidth. But of the three alternatives we have examined so far, it is the most effective.

ANSWER 5: HDB3 ENCODING

High-Density Bipolar 3 (HDB3)

The problem of synchronizing strings of consecutive 0s is solved differently in Europe and Japan than in the United States. This convention, called HDB3, introduces changes into the bipolar AMI pattern every time four consecutive 0s are encountered instead of waiting for the eight expected by B8ZS in North America. Although the name is HDB3, the pattern changes whenever there are four 0s in succession (see Figure 5.12).

Figure 5.12 *HDB3 encoding*

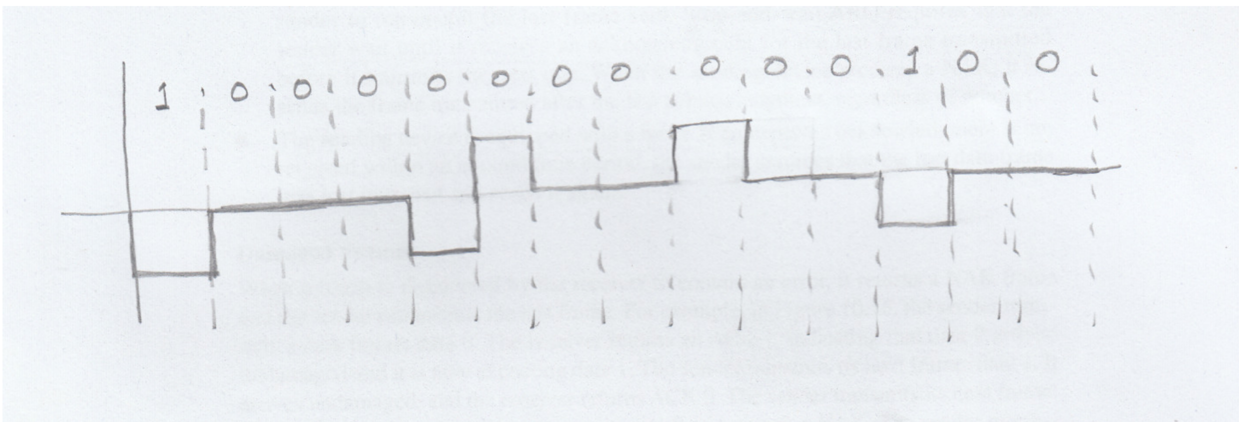


In HDB3 if four 0s come one after another, we change the pattern in one of four ways based on the polarity of the previous 1 and the number of 1s since the last substitution.

As in B8ZS, the pattern of violations in HDB3 is based on the polarity of the previous 1 bit. But unlike B8ZS, HDB3 also looks at the number of 1s that have occurred in the bit stream since the last substitution. Whenever the number of 1s since the last substitution is odd, HOB3 puts a violation in the place of the fourth consecutive 0. If the polarity of the previous bit was positive, the violation is positive. If the polarity of the previous bit was negative, the violation is negative.

Whenever the number of 1s since the last substitution is even, HOB3 puts violations in the places of both the first and the fourth consecutive 0s. If the polarity of the previous bit was positive, both violations are negative. If the polarity of the previous bit was negative, both violations are positive. All four patterns are shown in Figure 5.12.

As you can see, the point is to violate the standard pattern in ways that a machine can recognize as deliberate, and to use those violations to synchronize the system.



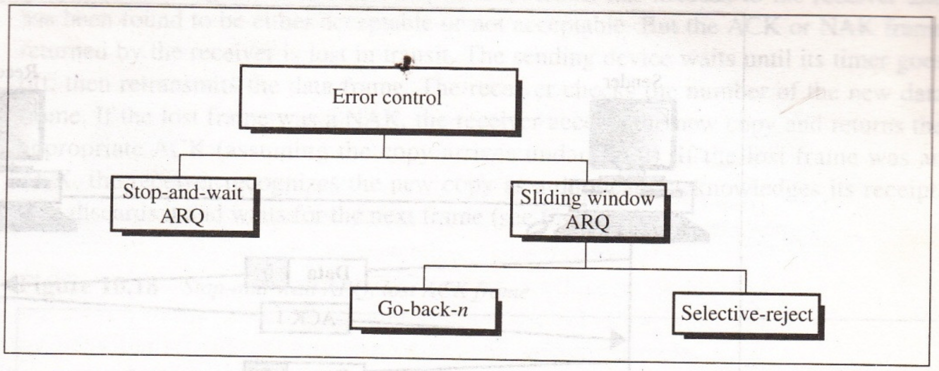
ANSWER 6: STOP AND WAIT ARQ ERROR CONTROL:

Stop-and-Wait ARQ

Stop-and-wait ARQ is a form of stop-and-wait flow control extended to include retransmission of data in case of lost or damaged frames. For retransmission to work, four features are added to the basic flow control mechanism:

- The sending device keeps a copy of the last frame transmitted until it receives an acknowledgment for that frame. Keeping a copy allows the sender to retransmit lost or damaged frames until they are received correctly.

Figure 10.15 *Categories of error control*



- For identification purposes, both data frames and ACK frames are numbered alternately 0 and 1. A data 0 frame is acknowledged by an ACK 1 frame, indicating that the receiver has gotten data 0 and is now expecting data 1. This numbering allows for identification of data frames in case of duplicate transmission (important in the case of lost acknowledgments, as we will see below).
- If an error is discovered in a data frame, indicating that it has been corrupted in transit, a NAK frame is returned. NAK frames, which are not numbered, tell the sender to retransmit the last frame sent. Stop-and-wait ARQ requires that the sender wait until it receives an acknowledgment for the last frame transmitted before it transmits the next one. When the sending device receives a NAK, it resends the frame transmitted after the last acknowledgment, regardless of number.
- The sending device is equipped with a timer. If an expected acknowledgment is not received within an allotted time period, the sender assumes that the last data frame was lost in transit and sends it again.

Damaged Frames

When a frame is discovered by the receiver to contain an error, it returns a NAK frame and the sender retransmits the last frame. For example, in Figure 10.16, the sender transmits a data frame: data 0. The receiver returns an ACK 1, indicating that data 0 arrived undamaged and it is now expecting data 1. The sender transmits its next frame: data 1. It arrives undamaged, and the receiver returns ACK 0. The sender transmits its next frame: data 0. The receiver discovers an error in data 0 and returns a NAK. The sender retransmits data 0. This time data 0 arrives intact, and the receiver returns ACK 1.

Lost Frame

Any of the three frame types can be lost in transit.

Lost Data Frame Figure 10.17 shows how stop-and-wait ARQ handles the loss of a data frame. As noted above, the sender is equipped with a timer that starts every time a data frame is transmitted. If the frame never makes it to the receiver, the receiver can never acknowledge it, positively or negatively. The sending device waits for an ACK or NAK frame until its timer goes off, at which point it tries again. It retransmits the last data frame, restarts its timer, and waits for an acknowledgment.

Figure 10.16 *Stop-and-wait ARQ, damaged frame*

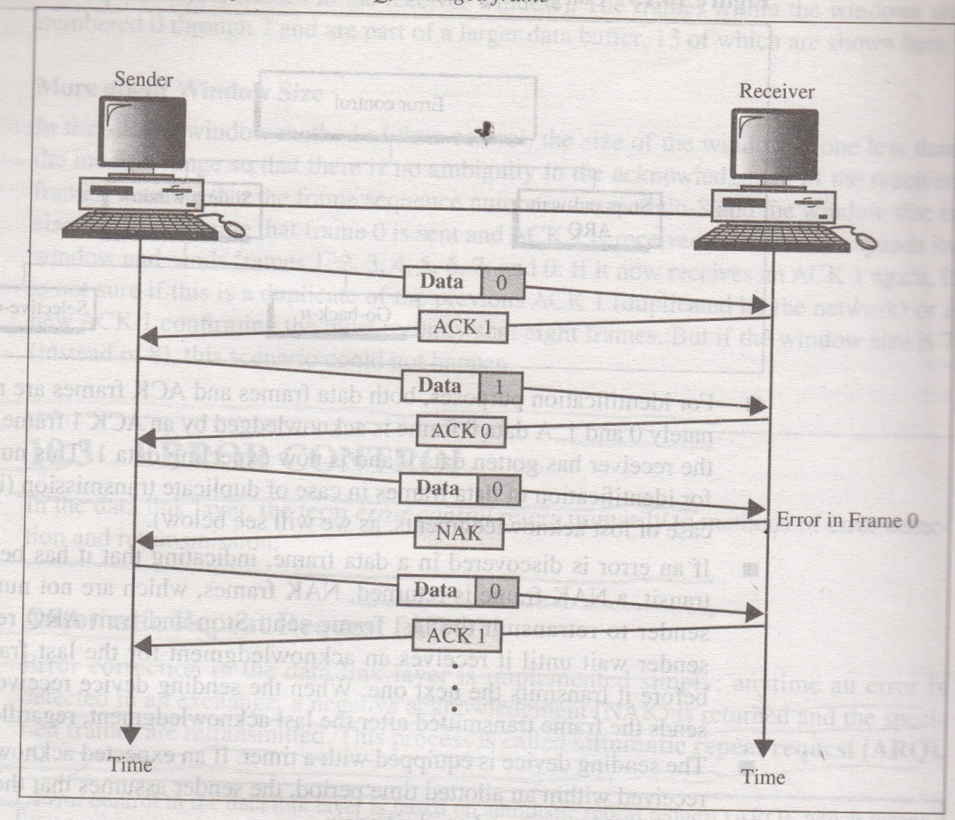
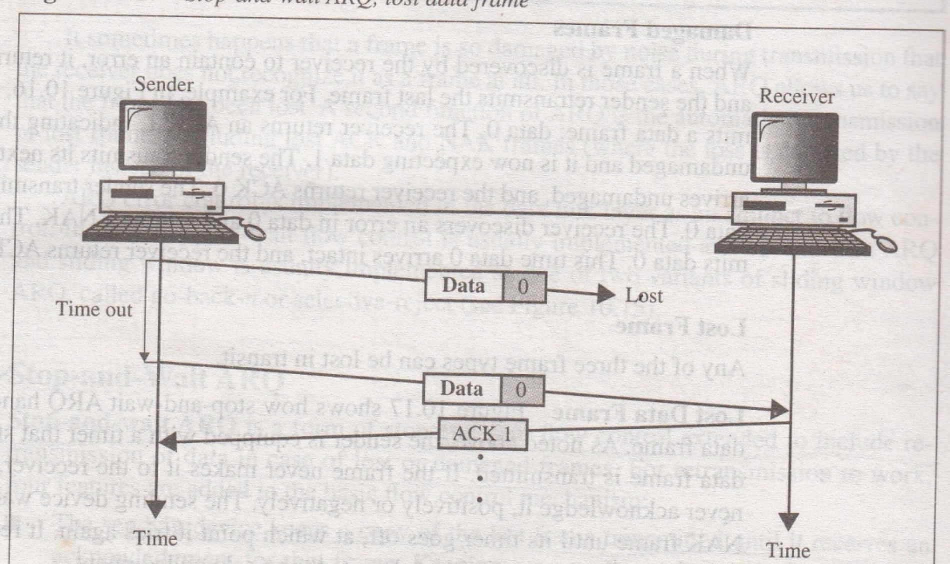
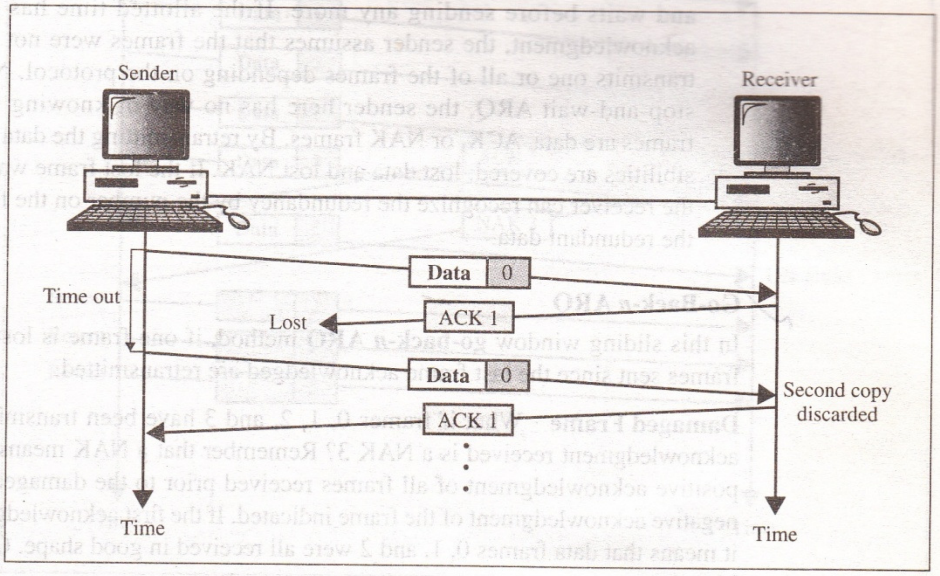


Figure 10.17 *Stop-and-wait ARQ, lost data frame*



Lost Acknowledgment In this case, the data frame has made it to the receiver and has been found to be either acceptable or not acceptable. But the ACK or NAK frame returned by the receiver is lost in transit. The sending device waits until its timer goes off, then retransmits the data frame. The receiver checks the number of the new data frame. If the lost frame was a NAK, the receiver accepts the new copy and returns the appropriate ACK (assuming the copy arrives undamaged). If the lost frame was an ACK, the receiver recognizes the new copy as a duplicate, acknowledges its receipt, then discards it and waits for the next frame (see Figure 10.18).

Figure 10.18 *Stop-and-wait ARQ, lost ACK frame*

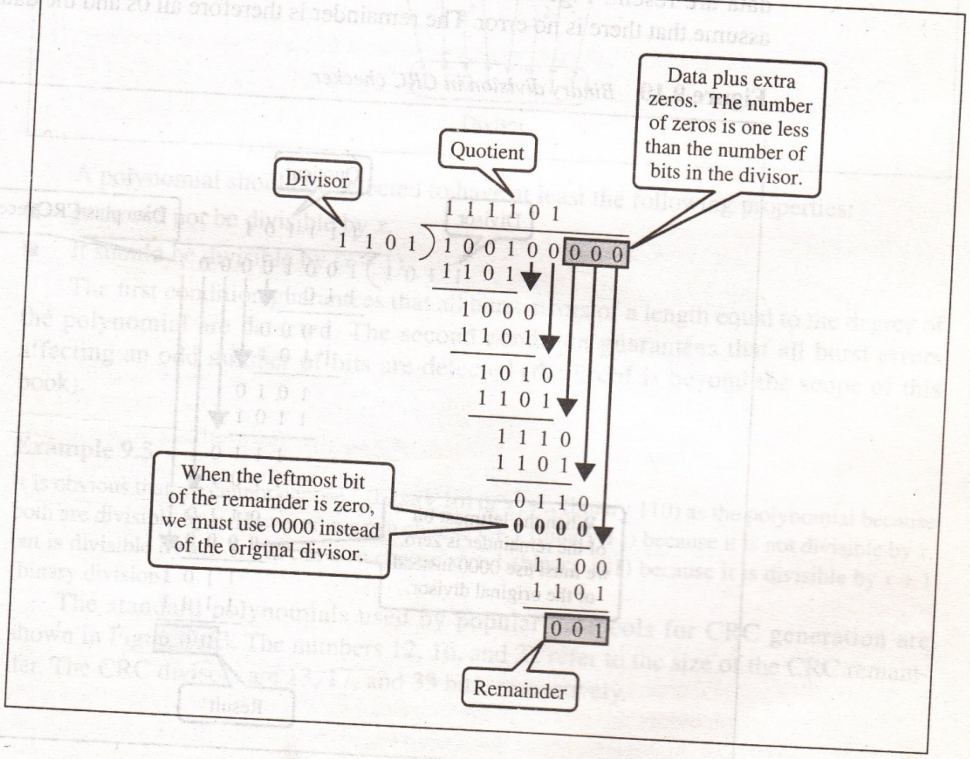


ANSWER 7: CRC GENERATOR & CHECKER

The CRC Generator

A CRC generator uses modulo-2 division. Figure 9.9 shows this process. In the first step, the four-bit divisor is subtracted from the first four bits of the dividend. Each bit of the divisor is subtracted from the corresponding bit of the dividend without disturbing the next higher bit. In our example, the divisor, 1101, is subtracted from the first four bits of the dividend, 1001, yielding 100 (the leading 0 of the remainder is dropped off).

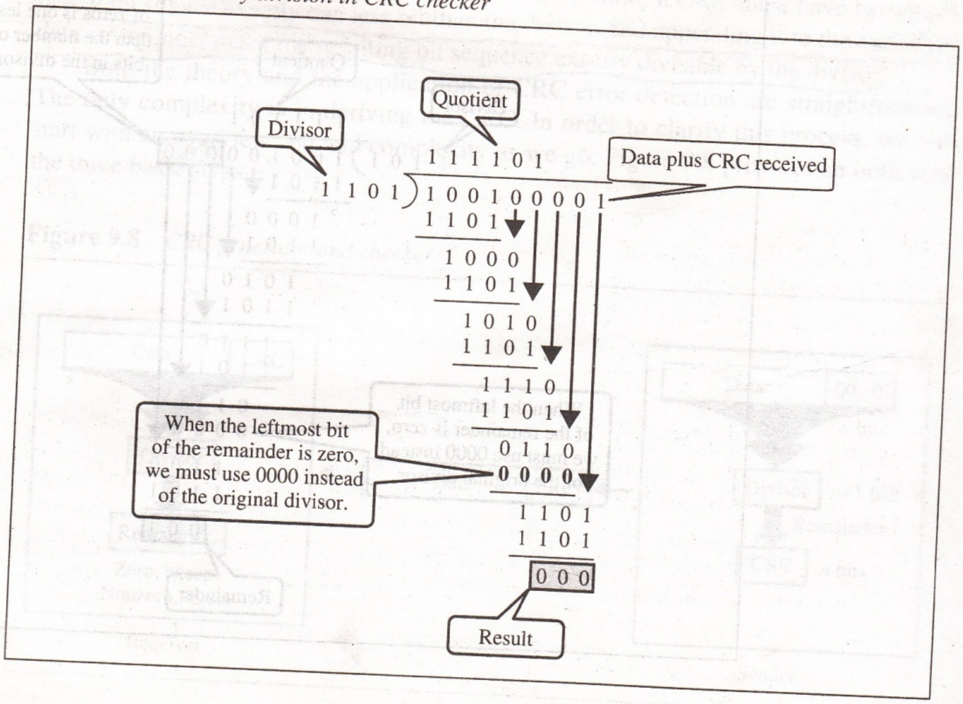
Figure 9.9 *Binary division in a CRC generator*



The CRC Checker

A CRC checker functions exactly like the generator. After receiving the data appended with the CRC, it does the same modulo-2 division. If the remainder is all 0s, the CRC is dropped and the data accepted; otherwise, the received stream of bits is discarded and data are resent. Figure 9.10 shows the same process of division in the receiver. We assume that there is no error. The remainder is therefore all 0s and the data are accepted.

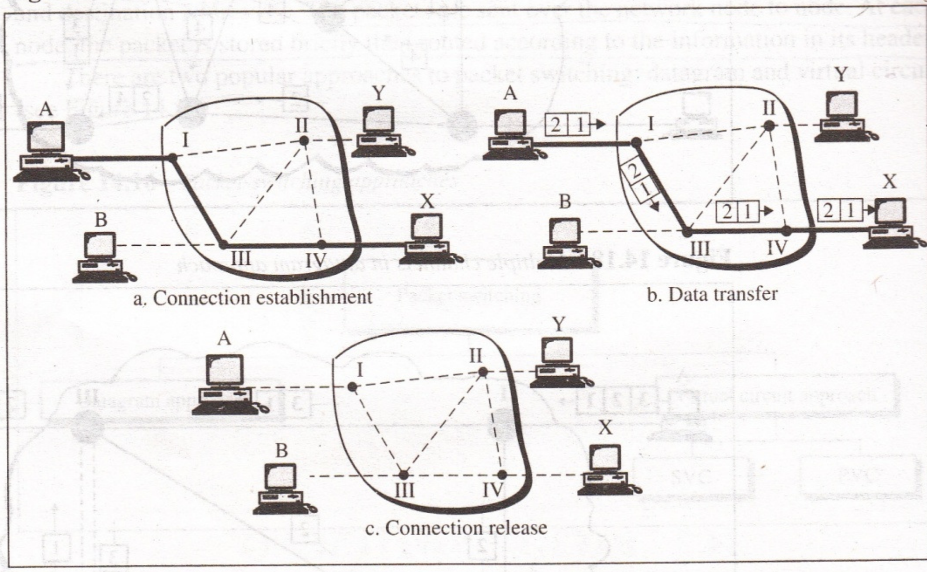
Figure 9.10 *Binary division in CRC checker*



ANSWER 8: SWITCHED VIRTUAL CIRCUIT (SVC) NETWORK

The SVC format is comparable conceptually to dial up lines in the circuit switc - hing. In this method , a virtual circuit is created whenever it is needed need and exists only for the duration of the specific exchange. For example, imagine that station A wants to send four packets to station X. First, A requests the establishment of a connection to X. Once the connection is in place, the packets are sent one after another and in sequential order. When the last packet has been received and, if necessary, acknowledged, the connection is released and that virtual circuit ceases to exist (see Figure 14.19). Only one single route exists for the duration of transmission, although the network could pick an alternate route in response to failure or congestion.

Figure 14.19 *Switched virtual circuit (SVC)*



Each time that A wishes to communicate with X, a new route is established. The route may be the same each time, or it may differ in response to varying network conditions.

ANSWER 9: DISTANCE VECTOR ROUTING

In **distance vector routing,** each router periodically shares its knowledge about the entire network with its neighbors. The three keys to understanding how this algorithm works are as follows:

1. **Knowledge about the whole network.** Each router shares its knowledge about the entire network. It sends all of its collected knowledge about the network to its neighbors. At the outset, a router's knowledge of the network may be sparse. How much it knows, however, is unimportant: it sends whatever it has.
2. **Routing only to neighbors.** Each router periodically sends its knowledge about the network only to those routers to which it has direct links. It sends whatever knowledge it has about the whole network through all of its ports. This information is received and kept by each neighboring router and used to update that router's own information about the network.
3. **Information sharing at regular intervals.** For example, every 30 seconds, each router sends its information about the whole network to its neighbors. This sharing occurs whether or not the network has changed since the last time information was exchanged.

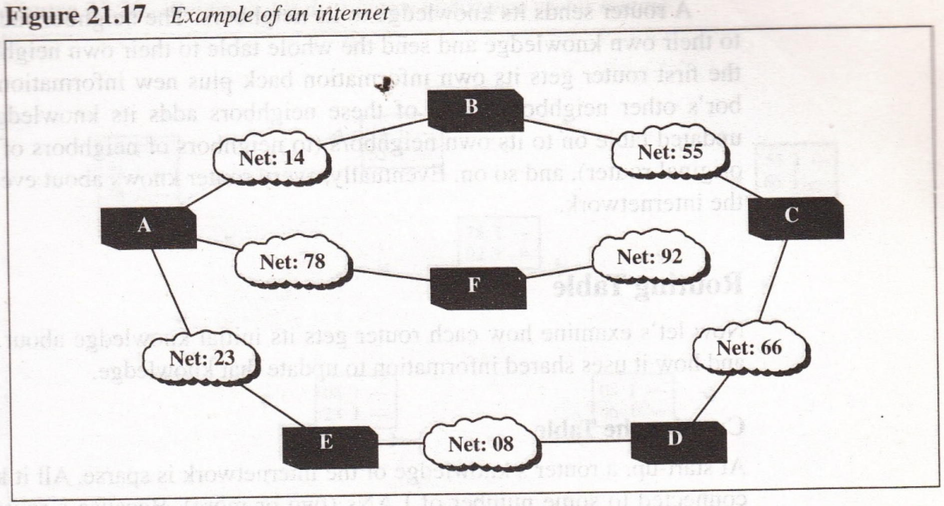
> In distance vector routing, each router periodically shares its knowledge about the entire network with its neighbors.

Sharing Information

To understand how distance vector routing works, examine the internet shown in Figure 21.17. In this example, the clouds represent local area networks (LANs). The number inside each cloud is that LAN's network ID. These LANs can be of any type (Ethernet, Token Ring, FDDI, etc.). The LANs are connected by routers (or gateways), represented by the boxes labeled A, B, C, D, E, and F.

Distance vector routing simplifies the routing process by assuming a cost of one unit for every link. In this way, the efficiency of transmission is a function only of the

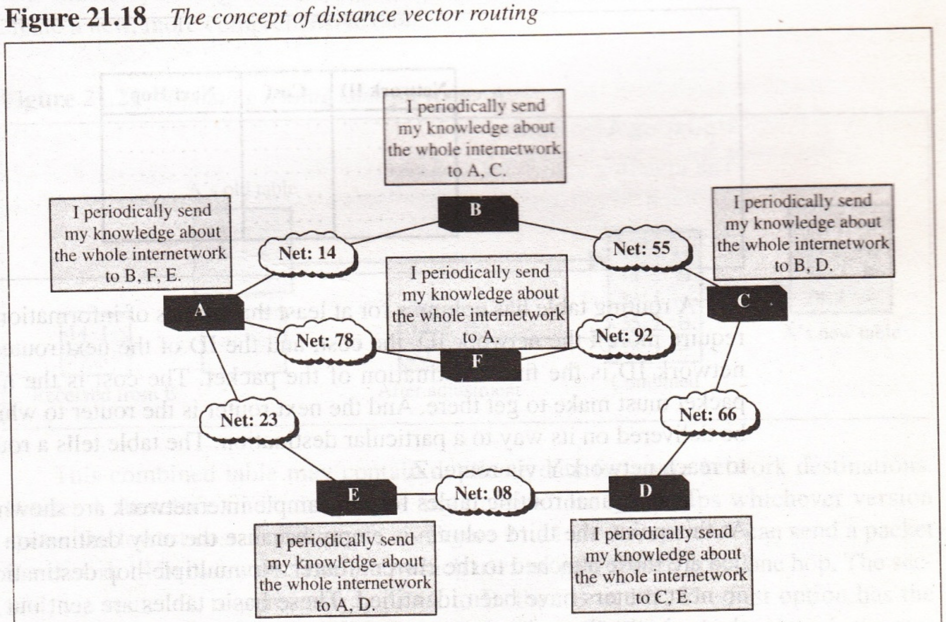
Figure 21.17 *Example of an internet*



number of links required to reach a destination. In distance vector routing, the cost is based on hop count.

Figure 21.18 shows the first step in the algorithm. The text boxes indicate the relationships of the routers in Figure 21.17 to their neighbors. As you can see, each router sends its information about the internetwork only to its immediate neighbors. How, then, do nonneighboring routers learn about each other and share knowledge?

Figure 21.18 *The concept of distance vector routing*



ANSWER 10: ISDN

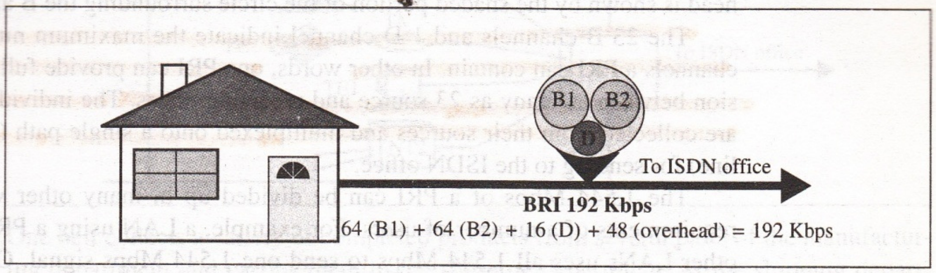
User Interfaces:

Digital subscriber loops are of two types: basic Rate Interface (BRI) & Primary Rate Interface (PRI). Both include one D and some no. of either B or H channel

BRI

The **basic rate interface (BRI)** specifies a digital pipe consisting of two B channels and one 16 Kbps D channel (see Figure 16.7).

Figure 16.7 *BRI*



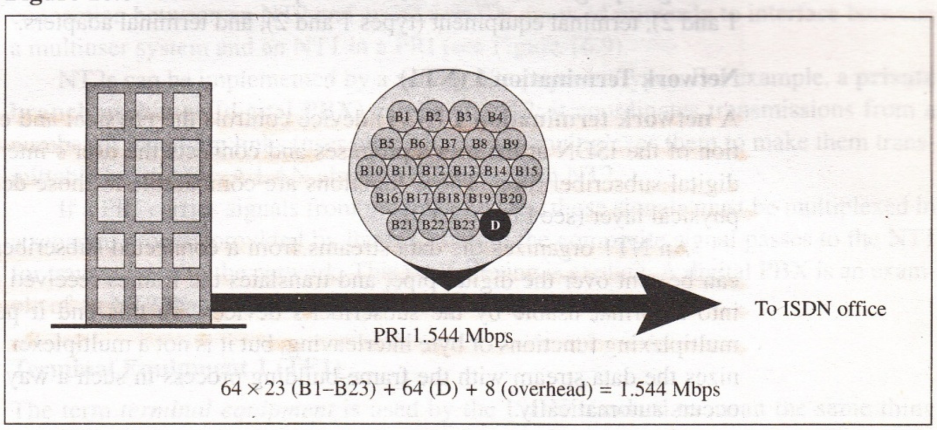
Two B channels of 64 Kbps each, plus one D channel of 16 Kbps, equals 144 Kbps. In addition, the BRI service itself requires 48 Kbps of operating overhead. BRI therefore requires a digital pipe of 192 Kbps. Conceptually, the BRI service is like a large pipe that contains three smaller pipes, two for the B channels and one for the D channel. The remainder of the space inside the large pipe carries the overhead bits required for its operation. In Figure 16.7, the overhead is shown by the shaded portion of the circle surrounding the B and D channels.

The BRI is designed to meet the needs of residential and small-office customers. In most cases, there is no need to replace the existing local-loop cable. The same twisted-pair local loop that delivers analog transmission can be used to handle digital transmission. Occasionally, however, some conditioning of the line is necessary.

PRI

The usual **primary rate interface (PRI)** specifies a digital pipe with 23 B channels and one 64 Kbps D channel (see Figure 16.8).

Figure 16.8 *PRI*



Twenty-three B channels of 64 Kbps each, plus one D channel of 64 Kbps equals 1.536 Mbps. In addition, the PRI service itself uses 8 Kbps of overhead. PRI therefore requires a digital pipe of 1.544 Mbps. Conceptually, the PRI service is like a large pipe containing 24 smaller pipes, 23 for the B channels and 1 for the D channel. The rest of the pipe carries the overhead bits required for its operation. In Figure 16.8, the overhead is shown by the shaded portion of the circle surrounding the B and D channels.

The 23 B channels and 1 D channel indicate the maximum number of separate channels a PRI can contain. In other words, one PRI can provide full-duplex transmission between as many as 23 source and receiving nodes. The individual transmissions are collected from their sources and multiplexed onto a single path (digital subscriber line) for sending to the ISDN office.

The 1.544 Mbps of a PRI can be divided up in many other ways to meet the requirements of a number of users. For example, a LAN using a PRI to connect it to other LANs uses all 1.544 Mbps to send one 1.544 Mbps signal. Other applications can use other combinations of the 64 Kbps B channels. At 1.544 Mbps, the capacity of the PRI digital pipe is exactly the same as the capacity of the T-1 line used to support the North American DS-1 telephone service. This similarity is not a coincidence. PRI was designed to be compatible with existing T-1 lines. In Europe, the PRI includes 30 B channels and 2 D channels, giving it a capacity of 2.048 Mbps—the capacity of an E-1 line.

For more specialized transmission needs, other channel combinations are also supported by the PRI standard. They are 3H0 + D, 4H0 + D, and H12 + D.

Functional Grouping

In the ISDN standard, the devices that enable users to access the services of the BRI or PRI are described by their functional duties and collected in functional groupings. Subscribers choose the specific devices best suited to their needs from these groupings. Remember that the ISDN defines only the functional behavior of each group. The standard does not say anything about implementation. Each functional grouping is a model that can be implemented using devices or equipment chosen by the subscriber. Functional groupings used at the subscriber's premises include network terminations (types 1 and 2), terminal equipment (types 1 and 2), and terminal adapters.

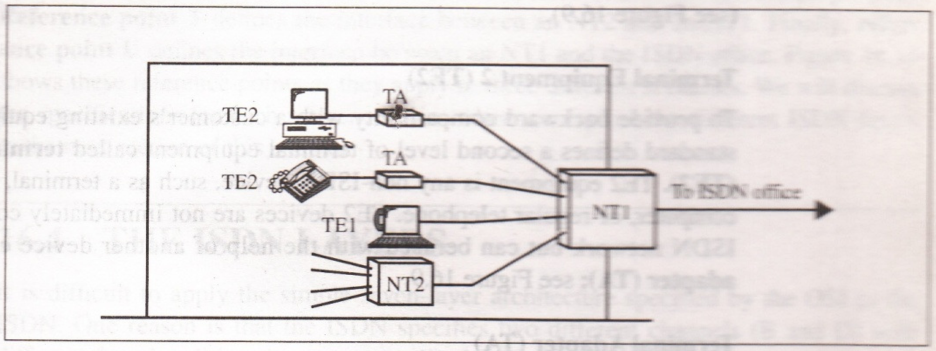
Network Termination 1 (NT1)

A **network termination 1 (NT1)** device controls the physical and electrical termination of the ISDN at the user's premises and connects the user's internal system to the digital subscriber loop. These functions are comparable to those defined for the OSI physical layer (see Figure 16.9).

An NT1 organizes the data streams from a connected subscriber into frames that can be sent over the digital pipe, and translates the frames received from the network into a format usable by the subscriber's devices. To this end it performs the basic multiplexing functions of byte interleaving, but it is not a multiplexer. An NT1 synchronizes the data stream with the frame-building process in such a way that multiplexing occurs automatically.

The easiest way to visualize how frame building in an NT1 can result in an interleaved signal is by analogy. Imagine a manufacturing plant with two conveyor belts.

Figure 16.9 *Functional grouping*



One belt collects a variety of completed products from several parts of the manufacturing department and carries them to the shipping department. At the shipping department, that belt meets a conveyor belt carrying boxes, each of which is designed to hold a specific product. The conveyer belt of products meets the conveyor belt of boxes at right angles. The two belts are synchronized so that as a given product reaches the end of its belt, it falls off into the appropriate box. The ordering of the boxes and products and the timing of the two belts must be controlled to keep the synchronization accurate. Discrepancies can result in a product's landing in the wrong box or missing the boxes altogether. With adequate synchronization, however, product packaging occurs accurately without switching or other manipulation. In the same way, an NT1 synchronizes the timing of the contributing data streams to the building of the outgoing frames so that bytes are interleaved without the need for multiplexing devices.

Network Termination 2 (NT2)

A **network termination 2 (NT2)** device performs functions at the physical, data link, and network layers of the OSI model (layers 1, 2, and 3). NT2s provide multiplexing (layer 1), flow control (layer 2), and packetizing (layer 3). An NT2 provides intermediate signal processing between the data-generating devices and an NT1. The NT1 is still required to provide a physical interface to the network. There must be a point-to-point connection between an NT2 and an NT1. NT2s are used primarily to interface between a multiuser system and an NT1 in a PRI (see Figure 16.9).

NT2s can be implemented by a variety of equipment types. For example, a **private branch exchange** (digital **PBX**) can be an NT2; it coordinates transmissions from a number of incoming links (user phone lines) and multiplexes them to make them transmittable by an NT1. A LAN also can function as an NT2.

If a PRI carries signals from multiple devices, those signals must be multiplexed in a separate process provided by the NT2 before the composite signal passes to the NT1 for transmission to the network. This multiplexing is explicit. A digital PBX is an example of an NT2 that contains explicit multiplexing functions.

Terminal Equipment 1 (TE1)

The term *terminal equipment* is used by the ISDN standard to mean the same thing as DTE in other protocols. It refers to digital subscriber equipment. **Terminal equipment 1 (TE1)** is any device that supports the ISDN standards. Examples of

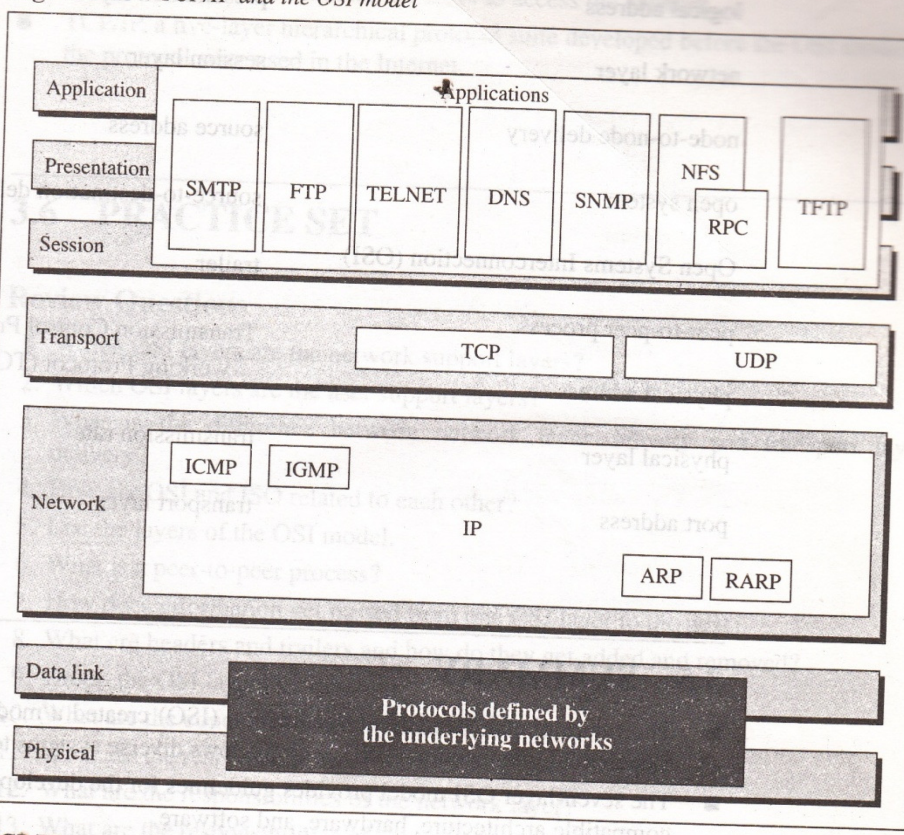
TE1 are digital telephones, integrated voice/data terminals.

ANSWER 11: OSI Vs TCP/IP Model

The TCP/IP protocol suite, used in the Internet, was developed prior to the OSI model. Therefore, the layers in the **Transmission Control Protocol/Internetworking Protocol (TCP/IP)** protocol suite do not match exactly with those in the OSI model. The TCP/IP protocol suite is made of five layers: physical, data link, network, transport, and application. The first four layers provide physical standards, network interface, internetworking, and transport functions that correspond to the first four layers of the OSI model. The three topmost layers in the OSI model, however, are represented in TCP/IP by a single layer called the *application layer* (see Figure 3.15).

TCP/IP is a hierarchical protocol made up of interactive modules, each of which provides a specific functionality, but they are not necessarily interdependent. Whereas the OSI model specifies which functions belong to each of its layers, the layers of the

Figure 3.15 *TCP/IP and the OSI model*



TCP/IP protocol suite contain relatively independent protocols that can be mixed and matched depending on the needs of the system. The term *hierarchical* means that each upper-level protocol is supported by one or more lower-level protocols.

At the transport layer, TCP/IP defines two protocols: Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). At the network layer, the main protocol defined by TCP/IP is Internetworking Protocol (IP), although there are some other protocols that support data movement in this layer. See Chapters 24 and 25 for a discussion of TCP/IP protocols.