**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SUBJECT NAME: COMPUTER NETWORKS SUBJECT CODE: CST52**

**UNIT II**

Data link layer – design issues – Services - Framing - Error Control - Flow Control - Error detection and correction codes - data link layer protocols -Simplex Protocol – Sliding window Protocols - Medium Access control sublayer – Channel allocation problem – Multiple Access protocols – ALOHA – CSMA Protocols - Collision-Free Protocols - Limited-Contention Protocols - Wireless LANs - 802.11 Architecture - 802.16 Architecture – Data link layer Switching - Uses of Bridges - Learning Bridges - Spanning Tree Bridges - Repeaters, Hubs, Bridges, Switches, Routers, and Gateways -   Virtual LANs

**2 MARKS**

1. **What are the responsibilities of data link layer?**

Specific responsibilities of data link layer include the following.

* Framing
* Physical addressing
* Flow control
* Error control
* Access control

1. **Mention the types of errors.**

There are 2 types of errors

* Single-bit error.
* Burst-bit error.

1. **Define the following terms.**
2. Single bit error: The term single bit error means that only one bit of a given data unit (such as byte character/data un it or packet) is changed from 1 to 0 or from 0 to 1.
3. Burst error: Means that 2 or more bits in the data unit have changed from 1 to 0 from 0 to 1.
4. **List out the available detection methods.**

There are 4 types of redundancy checks are used in data communication.

1. Vertical redundancy checks (VRC).
2. Longitudinal redundancy checks ( LRC).
3. Cyclic redundancy checks (CRC).
4. Checksum.
5. **What is redundancy?**

It is the error detecting mechanism, which means a shorter group of bits or extra bits may be app ended at the destination of each unit.

1. **Write short notes on VRC.**

The most common and least expensive mechanism for error detection is the vertical redundancy check (VRC) often called a parity check. In this technique a redundant bit called a parity bit, is appended to every data unit so, that the total number of 0’s in the unit (including the parity bit) becomes even.

1. **Write short notes on LRC.**

In longitudinal redundancy check (LRC), a block of bits is divided into rows and a redundant row of bits is added to the whole block.

1. **State the purpose of CRC code?(NOV 2012)**

A CRC-enabled device calculates a short, fixed-length binary sequence, known as the *CRC code*, for each block of data and sends or stores them both together. When a block is read or received the device repeats the calculation; if the new CRC does not match the one calculated earlier, then the block contains a data error and the device may take corrective action such as rereading or requesting the block be sent again, otherwise the data is assumed to be error free.

1. **Write short notes on CRC generator.**

A CRC generator uses a modulo-2 division.

1. In the first step, the 4 bit divisor is subtracted from the first 4 bit of the dividend.
2. Each bit of the divisor is subtracted from the corresponding
3. bit of the dividend without disturbing the next higher bit.
4. **Write short notes on CRC checker.**

A CRC checker functions exactly like a generator. After receiving the data appended with the CRC it does the same modulo-2 division. If the remainder is all 0’s the CRC is dropped and the data accepted. Otherwise, the received stream of bits are discarded and the dates are resent.

1. **Define checksum.**

The error detection method used by the higher layer protocol is called checksum. Checksum is based on the concept of redundancy.

1. **What are the steps followed in checksum generator?**

The sender follows these steps

* The units are divided into k sections each of n bits.
* All sections are added together using 2’s complement to get the sum.
* The sum is complemented and become the checksum.
* The checksum is sent with the data.

1. **List out the steps followed is checksum checker side.**

The receiver must follow these steps

1. The unit is divided into k section each of n bits.
2. All sections are added together using 1’s complement to get the sum.
3. The sum is complemented.
4. If the result is zero.
5. **Write short notes on error correction.**

It is the mechanism to correct the errors and it can be handled in 2 ways.

1. When an error is discovered, the receiver can have the sender retransmit the entire data unit.
2. A receiver can use an error correcting coder, which automatically corrects certain errors.
3. **Mention the types of error correcting methods.**

There are 2 error-correcting methods.

1. Single bit error correction
2. Burst error correction.
3. **What is the purpose of hamming code?**

A hamming code can be designed to correct burst errors of certain lengths. So the simple strategy used by the hamming code to correct single bit errors must be redesigned to be applicable for multiple bit correction

1. **Compare Error Detection and Error Correction:**

The correction of errors is more difficult than the detection. In error detection, checks only any error has occurred. In error correction, the exact number of bits that are corrupted and location in the message are known. The number of the errors and the size of the message are important factors.

1. **What is Forward Error Correction:**

Forward error correction is the process in which the receiver tries to guess the message by using redundant bits.

1. **Define Retransmission:**

Retransmission is a technique in which the receiver detects the occurrence of an error and asks the sender to resend the message. Resending is repeated until a message arrives that the receiver believes is error-freed.

1. **What are Data Words?**

In block coding, we divide our message into blocks, each of k bits, called datawords. The block coding process is one-to-one. The same dataword is always encoded as the same codeword.

1. **What are Code Words?**

“r” redundant bits are added to each block to make the length n = k + r. The resulting n-bit blocks are called codewords. 2n – 2k codewords that are not used. These codewords are invalid or illegal.

1. **What is a Linear Block Code?**

A linear block code is a code in which the exclusive OR (addition modulo-2) of two valid codeword’s creates another valid codeword.

1. **What are Cyclic Codes?**

Cyclic codes are special linear block codes with one extra property. In a cyclic code, if a codeword is cyclically shifted (rotated), the result is another codeword.

1. **Define Encoder:**

A device or program that uses predefined algorithms to encode, or compress audio or video data for storage or transmission use. A circuit that is used to convert between digital video and analog video.

1. **Define Decoder**

A device or program that translates encoded data into its original format (e.g., it decodes the data). The term is often used in reference to MPEG-2 video and sound data, which must be decoded before it is output.

1. **What is framing?**

Framing in the data link layer separates a message from one source to a destination, or from other messages to other destinations, by adding a sender address and a destination address. The destination address defines where the packet has to go and the sender address helps the recipient acknowledge the receipt.

1. **What is Fixed –Size Framing?**

In fixed-size framing, there is no need for defining the boundaries of the frames. The size itself can be used as a delimiter.

1. **What is Bit Stuffing?**

Bit stuffing is the process of adding one extra 0 whenever five consecutive Is follow a 0 in the data, so that the receiver does not mistake the pattern 0111110 for a flag.

1. **Define Character Stuffing.**

In byte stuffing (or character stuffing), a special byte is added to the data section of the frame when there is a character with the same pattern as the flag. The data section is stuffed with an extra byte. This byte is usually called the escape character (ESC), which has a predefined bit pattern. Whenever the receiver encounters the ESC character, it removes it from the data section and treats the next character as data, not a delimiting flag.

1. **What is Flow Control?**

Flow control refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgment.

1. **What is Error Control?**

Error control is both error detection and error correction. It allows the receiver to inform the sender of any frames lost or damaged in transmission and coordinates the retransmission of those frames by the sender. In the data link layer, the term error control refers primarily to methods of error detection and retransmission.

1. **What Automatic Repeat Request (ARQ)?**

Error control is both error detection and error correction. It allows the receiver to inform the sender of any frames lost or damaged in transmission and coordinates the retransmission of those frames by the sender. In the data link layer, the term error control refers primarily to methods of error detection and retransmission. Error control in the data link layer is often implemented simply: Any time an error is detected in an exchange, specified frames are retransmitted. This process is called automatic repeat request (ARQ).

1. **What is Stop-and-Wait Protocol?**

In Stop and wait protocol, sender sends one frame, waits until it receives confirmation from the receiver (okay to go ahead), and then sends the next frame

1. **What is Stop-and-Wait Automatic Repeat Request?**

Error correction in Stop-and-Wait ARQ is done by keeping a copy of the sent frame and retransmitting of the frame when the timer expires.

1. **What is usage of Sequence Number in Reliable Transmission?**

The protocol specifies that frames need to be numbered. This is done by using sequence numbers. A field is added to the data frame to hold the sequence number of that frame. Since we want to minimize the frame size, the smallest range that provides unambiguous communication. The sequence numbers can wrap around.

1. **What is Pipelining?**

In networking and in other areas, a task is often begun before the previous task has ended. This is known as pipelining.

1. **What is Sliding Window?**

The sliding window is an abstract concept that defines the range of sequence numbers that is the concern of the sender and receiver. In other words, he sender and receiver need to deal with only part of the possible sequence numbers.

1. **What is Piggy Backing?**

A technique called piggybacking is used to improve the efficiency of the bidirectional protocols. When a frame is carrying data from A to B, it can also carry control information about arrived (or lost) frames from B; when a frame is carrying data from B to A, it can also carry control information about the arrived (or lost) frames from A.

1. **What is frame bursting?(NOV 2011)**

Frame-bursting is a communication protocol feature used at the link layer in communication networks to alter the transmission characteristics in order to benefit from higher data transfer throughput.

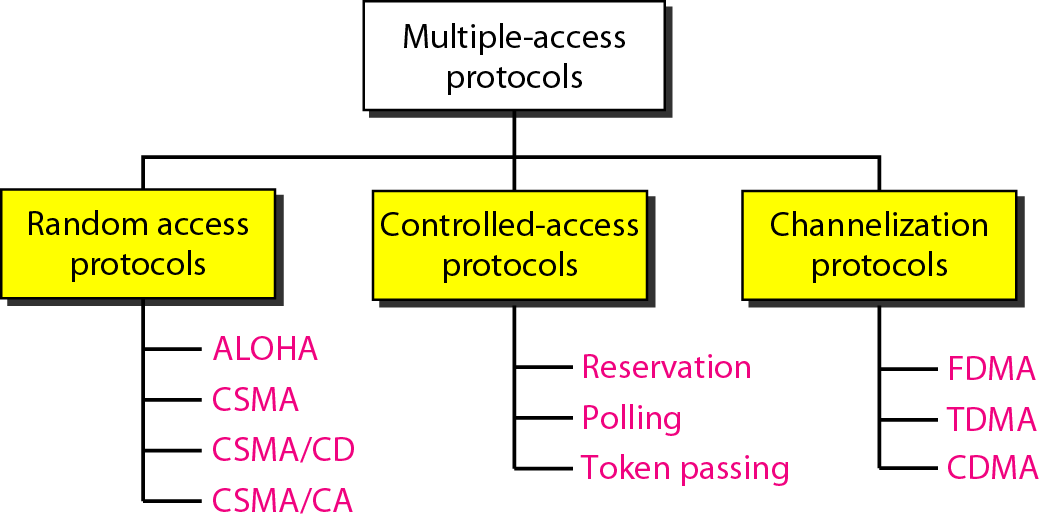
1. **Define FDDI? (NOV 2011)**

FDDI (Fiber Distributed Data Interface) is a set of ANSI and ISO standards for data transmission on fiber optic lines in a local area network (LAN) that can extend in range up to 200 km (124 miles). The FDDI protocol is based on the token ring protocol. In addition to being large geographically, an FDDI local area network can support thousands of users. FDDI is frequently used on the backbone for a wide area network (WAN).

1. **What are adaptive algorithms?(APR 2011)**

An **adaptive algorithm** is an algorithm that changes its behavior based on information available at the time it is run. This might be information about computational resources available, or the history of data recently received

1. **Name the Multiple Access Protocols (Apr 2013)**

****

1. **Define ARP(APR 2012)**

ARP stands for Address resolution protocol, maps an IP address to a MAC address

1. **What do you mean by RARP?**

RARP stands for Reverse Address resolution protocol, maps an MAC address to a IP address

1. **Define DHCP**

The Dynamic Host Configuration Protocol has been derived to provide dynamic configuration. DHCP is also needed when a host moves from network to network or is connected and disconnected from a network.

1. **What is Ethernet**

A system for connecting a number of computer systems to form a local area network, with protocols to control the passing of information and to avoid simultaneous transmission by two or more systems

1. **List out various internetworking devices**

* Gateway
* Routers
* Switches
* Access points
* Repeaters
* Hubs
* Bridges

1. **Define Hubs**

A common connection point for devices in a network. Hubs are commonly used to connect segments of a LAN. A hub contains multiple ports. When a packet arrives at one port, it is copied to the other ports so that all segments of the LAN can see all packets

1. **Define Bridges**

A **bridge** device filters data traffic at a network boundary. Bridges reduce the amount of traffic on a local area network (LAN) by dividing it into two segments. Bridges operate at the data link layer (Layer 2) of the OSI model. Bridges inspect incoming traffic and decide whether to forward or discard it.

1. **Define Switch and its uses**

A **network switch** is a small hardware device that joins multiple computers together within one local area network (LAN).Ethernet switch devices were commonly used on home networks before home routers became popular; broadband routers integrate Ethernet switches directly into the unit as one of their many functions. High-performance network switches are still widely used in corporate networks and data centers.

1. **What is the use of repeater?**

A repeater is used to amplify signals carried by a network. The function of a repeater is to receive incoming signals or a packet of data, regenerate the signals to their original strength and retransmit them. When a repeater amplifies the electric signals in a network, they allow transmissions to travel a greater distance. For a repeater to work, both network segments must be identical.

1. **What are the problems overcome by bridge when compared with hub?(NOV 2012)**

The biggest problem with hubs is their simplicity. Since every packet is sent out to every computer on the network, there is a lot of wasted transmission. This means that the network can easily become bogged down.A bridge goes one step up on a hub in that it looks at the destination of the packet before sending. If the destination address is not on the other side of the bridge it will not transmit the data

1. **Define PPP**

PPP (Point-to-Point Protocol) is a protocol for communication between two computers using a serial interface, typically a personal computer connected by phone line to a server

1. **What is ATM Network**

The **Asynchronous Transfer Mode (ATM)** protocol architecture is designed to support the transfer of data with a range of guarantees for quality of service. The user data is divided into small, fixed-length packets, called cells, and transported over virtual connections. ATM operates over high data rate physical circuits, and the simple structure of ATM cells allows switching to be performed in hardware, which improves the speed and efficiency of ATM switches.

1. **Define ATM adaptation layer**

The basic function of the ATM adaptation layer is to convert the user data supplied by higher layers into 48-byte blocks of data. The ATM adaptation layer is divided into two sub-layers –

* The convergence sub-layer, and
* The segmentation and re-assembly sub-layer

1. **What is convergence layer?**

The convergence sub-layer provides services to higher layers through a set of protocols

1. **What is segmentation and re-assembly sub-layer?**

The segmentation and re-assembly sub-layer separates the messages from the convergence sub-layer into ATM cells.

1. **Define MPLS**

**Multiprotocol Label Switching** (**MPLS**) is a mechanism in high-performance telecommunications networks that directs data from one network node to the next based on short path labels rather than long network addresses, avoiding complex lookups in a routing table. The labels identify virtual links (*paths*) between distant nodes rather than endpoints. MPLS can encapsulate packets of various network protocols

1. **What is Ring Topology**

A **ring network** is a network topology in which each node connects to exactly two other nodes, forming a single continuous pathway for signals through each node - a ring. Data travel from node to node, with each node along the way handling every packet.

1. **Define Physical Ring**

Devices are attached via a series of point-to-point links that form a closed loop. In most physical ring topologies, the links were typically simplex, resulting in transmissions that always moved in one direction around the ring. Each device took the signal it received on its input link and repeated the signal to its output link.

1. **Define Logical Ring**

**Logical topology**, or **signal topology**, is the arrangement of devices on a computer network and how they communicate with one another. How devices are connected to the network through the actual cables that transmit data, or the physical structure of the network, is called the physical topology. Physical topology defines how the systems are physically connected. It represents the physical layout of the devices on the network. The logical topology defines how the systems communicate across the physical topologies.

**11 MARKS**

1. **Explain about the data link layer**

**Design Issues**

The data link layer uses the services of the physical layer to send and receivebits over communication channels. It has a number of functions, including:

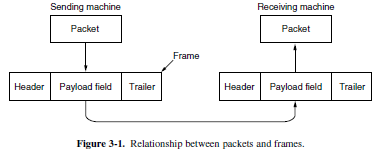
1. Providing a well-defined service interface to the network layer.

2. Dealing with transmission errors.

3. Regulating the flow of data so that slow receivers are not swamped

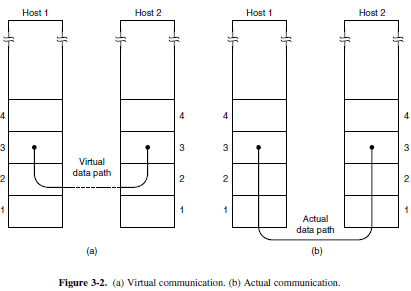
by fast senders.

To accomplish these goals, the data link layer takes the packets it gets from thenetwork layer and encapsulates them into **frames** for transmission. Each framecontains a frame header, a payload field for holding the packet, and a frametrailer, as illustrated in Fig. 3-1. Frame management forms the heart of what thedata link layer does. In the following sections we will examine all the abovementionedissues in detail.

****

**Services Provided to the Network Layer**

The function of the data link layer is to provide services to the network layer.The principal service is transferring data from the network layer on the source machineto the network layer on the destination machine. On the source machine isan entity, call it a process, in the network layer that hands some bits to the datalink layer for transmission to the destination. The job of the data link layer is totransmit the bits to the destination machine so they can be handed over to the networklayer there, as shown in Fig. 3-2(a). The actual transmission follows thepath of Fig. 3-2(b), but it is easier to think in terms of two data link layer processescommunicating using a data link protocol.

****

The data link layer can be designed to offer various services. The actual servicesthat are offered vary from protocol to protocol. Three reasonable possibilitiesthat we will consider in turn are:

1. Unacknowledged connectionless service.

2. Acknowledged connectionless service.

3. Acknowledged connection-oriented service.

Unacknowledged connectionless service consists of having the source machinesend independent frames to the destination machine without having thedestination machine acknowledge them. Ethernet is a good example of a data linklayer that provides this class of service. No logical connection is established beforehandor released afterward. If a frame is lost due to noise on the line, noattempt is made to detect the loss or recover from it in the data link layer. Thisclass of service is appropriate when the error rate is very low, so recovery is leftto higher layers. It is also appropriate for real-time traffic, such as voice, in whichlate data are worse than bad data.

The next step up in terms of reliability is acknowledged connectionless service.When this service is offered, there are still no logical connections used, buteach frame sent is individually acknowledged. In this way, the sender knowswhether a frame has arrived correctly or been lost. If it has not arrived within aspecified time interval, it can be sent again. This service is useful over unreliablechannels, such as wireless systems. 802.11 (WiFi) is a good example of this classof service.

When connection-oriented service is used, transfers go through three distinctphases. In the first phase, the connection is established by having both sides initializevariables and counters needed to keep track of which frames have been receivedand which ones have not. In the second phase, one or more frames are actuallytransmitted. In the third and final phase, the connection is released, freeingup the variables, buffers, and other resources used to maintain the connection.

**Framing**

To provide service to the network layer, the data link layer must use the serviceprovided to it by the physical layer. What the physical layer does is accept araw bit stream and attempt to deliver it to the destination. If the channel is noisy,as it is for most wireless and some wired links, the physical layer will add someredundancy to its signals to reduce the bit error rate to a tolerable level. However,the bit stream received by the data link layer is not guaranteed to be error free.Some bits may have different values and the number of bits received may be lessthan, equal to, or more than the number of bits transmitted. It is up to the datalink layer to detect and, if necessary, correct errors.The usual approach is for the data link layer to break up the bit stream intodiscrete frames, compute a short token called a checksum for each frame, and include

the checksum in the frame when it is transmitted. When a frame arrives at the destination,the checksum is recomputed. If the newly computed checksum is different fromthe one contained in the frame, the data link layer knows that an error has occurredand takes steps to deal with it (e.g., discarding the bad frame and possiblyalso sending back an error report).Breaking up the bit stream into frames is more difficult than it at first appears.

A good design must make it easy for a receiver to find the start of new frameswhile using little of the channel bandwidth. We will look at four methods:

1. Byte count.

2. Flag bytes with byte stuffing.

3. Flag bits with bit stuffing.

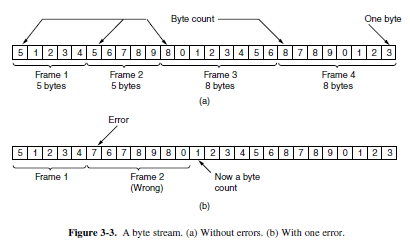
4. Physical layer coding violations.

The first framing method uses a field in the header to specify the number ofbytes in the frame. When the data link layer at the destination sees the byte count,it knows how many bytes follow and hence where the end of the frame is. Thistechnique is shown in Fig. 3-3(a) for four small example frames of sizes 5, 5, 8,

and 8 bytes, respectively.The trouble with this algorithm is that the count can be garbled by a transmission

error. For example, if the byte count of 5 in the second frame of Fig. 3-3(b)becomes a 7 due to a single bit flip, the destination will get out of synchronization.

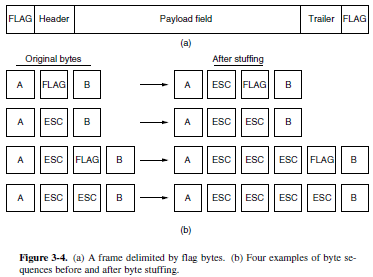
It will then be unable to locate the correct start of the next frame. Even if thechecksum is incorrect so the destination knows that the frame is bad, it still has noway of telling where the next frame starts. Sending a frame back to the sourceasking for a retransmission does not help either, since the destination does notknow how many bytes to skip over to get to the start of the retransmission. Forthis reason, the byte count method is rarely used by itself.

****

The second framing method gets around the problem of resynchronizationafter an error by having each frame start and end with special bytes. Often thesame byte, called a **flag byte**, is used as both the starting and ending delimiter.This byte is shown in Fig. 3-4(a) as FLAG. Two consecutive flag bytes indicatethe end of one frame and the start of the next. Thus, if the receiver ever loses synchronizationit can just search for two flag bytes to find the end of the currentframe and the start of the next frame.

However, there is a still a problem we have to solve. It may happen that theflag byte occurs in the data, especially when binary data such as photographs orsongs are being transmitted. This situation would interfere with the framing. Oneway to solve this problem is to have the sender’s data link layer insert a specialescape byte (ESC) just before each ‘‘accidental’’ flag byte in the data. Thus, aframing flag byte can be distinguished from one in the data by the absence orpresence of an escape byte before it. The data link layer on the receiving end removes the escape bytes before giving the data to the network layer. This technique is called **byte stuffing**.

The byte-stuffing scheme depicted in Fig. 3-4 is a slight simplification of theone used in **PPP** (**Point-to-Point Protocol**), which is used to carry packets overcommunications links.

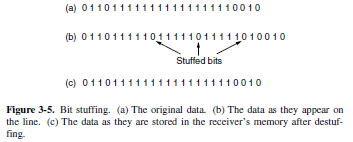
****

The third method of delimiting the bit stream gets around a disadvantage ofbyte stuffing, which is that it is tied to the use of 8-bit bytes. Framing can be alsobe done at the bit level, so frames can contain an arbitrary number of bits made upof units of any size. It was developed for the once very popular **HDLC** (**Highlevel**

**Data Link Control**) protocol. Each frame begins and ends with a specialbit pattern, 01111110 or 0x7E in hexadecimal. This pattern is a flag byte. Wheneverthe sender’s data link layer encounters five consecutive 1s in the data, itautomatically stuffs a 0 bit into the outgoing bit stream. This **bit stuffing** is analogousto byte stuffing, in which an escape byte is stuffed into the outgoing characterstream before a flag byte in the data. It also ensures a minimum density oftransitions that help the physical layer maintain synchronization. USB (UniversalSerial Bus) uses bit stuffing for this reason.

When the receiver sees five consecutive incoming 1 bits, followed by a 0 bit,it automatically destuffs (i.e., deletes) the 0 bit. Just as byte stuffing is completelytransparent to the network layer in both computers, so is bit stuffing. If the userdata contain the flag pattern, 01111110, this flag is transmitted as 011111010 but

stored in the receiver’s memory as 01111110. Figure 3-5 gives an example of bitstuffing.With bit stuffing, the boundary between two frames can be unambiguouslyrecognized by the flag pattern. Thus, if the receiver loses track of where it is, allit has to do is scan the input for flag sequences, since they can only occur at frameboundaries and never within the data.

****

With both bit and byte stuffing, a side effect is that the length of a frame nowdepends on the contents of the data it carries. For instance, if there are no flagbytes in the data, 100 bytes might be carried in a frame of roughly 100 bytes. If,however, the data consists solely of flag bytes, each flag byte will be escaped and

the frame will become roughly 200 bytes long. With bit stuffing, the increasewould be roughly 12.5% as 1 bit is added to every byte.

**Error Control**

Having solved the problem of marking the start and end of each frame, wecome to the next problem: how to make sure all frames are eventually delivered tothe network layer at the destination and in the proper order.For unacknowledged connectionless service itmight be fine if the sender just kept outputting frames without regard to whetherthey were arriving properly. But for reliable, connection-oriented service it would

not be fine at all.

The usual way to ensure reliable delivery is to provide the sender with somefeedback about what is happening at the other end of the line. Typically, the protocolcalls for the receiver to send back special control frames bearing positive ornegative acknowledgements about the incoming frames. If the sender receives apositive acknowledgement about a frame, it knows the frame has arrived safely.On the other hand, a negative acknowledgement means that something has gonewrong and the frame must be transmitted again.

An additional complication comes from the possibility that hardware troublesmay cause a frame to vanish completely (e.g., in a noise burst). In this case, thereceiver will not react at all, since it has no reason to react. Similarly, if the acknowledgementframe is lost, the sender will not know how to proceed. It should

be clear that a protocol in which the sender transmits a frame and then waits foran acknowledgement, positive or negative, will hang forever if a frame is ever lostdue to, for example, malfunctioning hardware or a faulty communication channel.

This possibility is dealt with by introducing timers into the data link layer.When the sender transmits a frame, it generally also starts a timer. The timer isset to expire after an interval long enough for the frame to reach the destination,be processed there, and have the acknowledgement propagate back to the sender.Normally, the frame will be correctly received and the acknowledgement will getback before the timer runs out, in which case the timer will be canceled.However, if either the frame or the acknowledgement is lost, the timer will gooff, alerting the sender to a potential problem. The obvious solution is to justtransmit the frame again. However, when frames may be transmitted multipletimes there is a danger that the receiver will accept the same frame two or moretimes and pass it to the network layer more than once. To prevent this from happening,it is generally necessary to assign sequence numbers to outgoing frames,so that the receiver can distinguish retransmissions from originals.

The whole issue of managing the timers and sequence numbers so as to ensurethat each frame is ultimately passed to the network layer at the destination exactlyonce, no more and no less, is an important part of the duties of the data link layer(and higher layers). Later in this chapter, we will look at a series of increasinglysophisticated examples to see how this management is done.

**Flow Control**

Another important design issue that occurs in the data link layer (and higherlayers as well) is what to do with a sender that systematically wants to transmitframes faster than the receiver can accept them. This situation can occur whenthe sender is running on a fast, powerful computer and the receiver is running on a

slow, low-end machine. A common situation is when a smart phone requests aWeb page from a far more powerful server, which then turns on the fire hose andblasts the data at the poor helpless phone until it is completely swamped. Even ifthe transmission is error free, the receiver may be unable to handle the frames asfast as they arrive and will lose some.

Clearly, something has to be done to prevent this situation. Two approachesare commonly used. In the first one, **feedback-based flow control**, the receiversends back information to the sender giving it permission to send more data, or atleast telling the sender how the receiver is doing. In the second one, **rate-basedflow control**, the protocol has a built-in mechanism that limits the rate at whichsenders may transmit data, without using feedback from the receiver.

The study of feedback-based flow control schemes, primarilybecause rate-based schemes are only seen as part of the transport layer. Feedback-based schemes are seen at both the link layer and higher layers. Thelatter is more common these days, in which case the link layer hardware is designedto run fast enough that it does not cause loss. For example, hardware implementationsof the link layer as **NICs** (**Network Interface Cards**) are sometimessaid to run at ‘‘wire speed,’’ meaning that they can handle frames as fast as

they can arrive on the link. Any overruns are then not a link problem, so they arehandled by higher layers.

1. **Explain About Error Detection and Correction (apr 2012)**

**Error detection and correction**

There are many reasons such as noise, cross-talk etc., which may help data to get corrupted during transmission. The upper layers work on some generalized view of network architecture and are not aware of actual hardware data processing. Hence, the upper layers expect error-free transmission between the systems. Most of the applications would not function expectedly if they receive erroneous data. Applications such as voice and video may not be that affected and with some errors they may still function well.

Data-link layer uses some error control mechanism to ensure that frames (data bit streams) are transmitted with certain level of accuracy. But to understand how errors is controlled, it is essential to know what types of errors may occur.

**Types of Errors**

There may be three types of errors:

* **Single bit error**



In a frame, there is only one bit, anywhere though, which is corrupt.

* **Multiple bits error**



Frame is received with more than one bits in corrupted state.

* **Burst error**



Frame contains more than1 consecutive bits corrupted.

Error control mechanism may involve two possible ways:

* Error detection
* Error correction

**Error Detection**

Errors in the received frames are detected by means of Parity Check and Cyclic Redundancy Check (CRC). In both cases, few extra bits are sent along with actual data to confirm that bits received at other end are same as they were sent. If the counter-check at receiver’ end fails, the bits are considered corrupted.

**Parity Check**

One extra bit is sent along with the original bits to make number of 1s either even in case of even parity, or odd in case of odd parity.

The sender while creating a frame counts the number of 1s in it. For example, if even parity is used and number of 1s is even then one bit with value 0 is added. This way number of 1s remains even. If the number of 1s is odd, to make it even a bit with value 1 is added.

Even Parity

The receiver simply counts the number of 1s in a frame. If the count of 1s is even and even parity is used, the frame is considered to be not-corrupted and is accepted. If the count of 1s is odd and odd parity is used, the frame is still not corrupted.

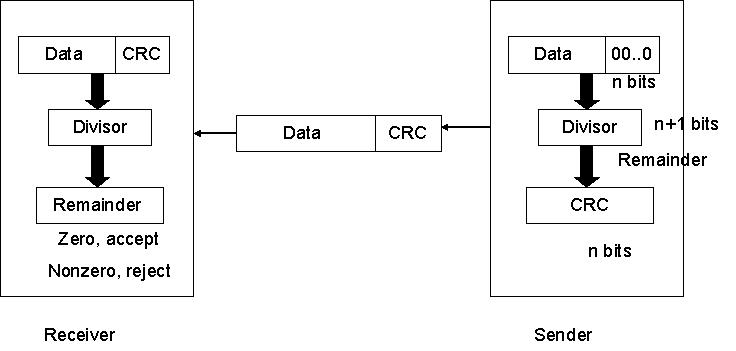
If a single bit flips in transit, the receiver can detect it by counting the number of 1s. But when more than one bits are erroneous, then it is very hard for the receiver to detect the error.

1. **Explain about Cyclic Redundancy Check**

**Cyclic Redundancy Check**

CRC is based on binary division. In this a sequence of redundant bits, called CRC remainder is appended to the end of a data unit so that the resulting data unit becomes exactly divisible by a second predetermined binary number. At its destination, the incoming data unit is divided by the same number. If at this step there is no reminder, the data unit is assumed to be intact and therefore accepted. A remainder indicates that the data unit has been changed in transit and therefore must be rejected.

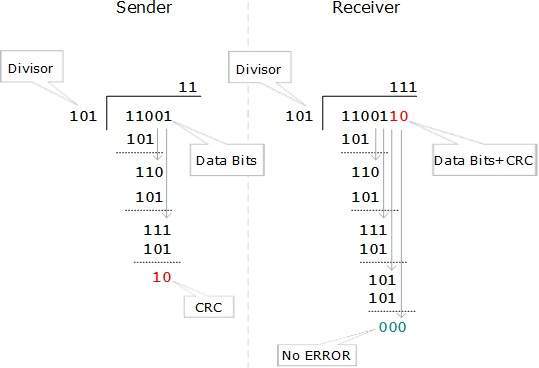
Here, the remainder is the CRC. It must have exactly one less bit than the divisor, and appending it to the end of the data string must make the resulting bit sequence exactly divisible by the divisor.



First, astringofn-10s is appended to the data unit. The number of 0s is one less than the number of bits in the divisor which is n bits. Then the newly elongated data unit is divided by the divisor using a process called binary division. The remainder is CRC. The CRC is replaces the appended 0s at the end of the data unit.

The data unit arrives at the receiver first, followed by the CRC. The receiver treats whole string as the data unit and divides it by the same divisor that was used to find the CRC remainder. If the remainder is 0 then the data unit is error free. Otherwise it having some error and it must be discarded.

CRC is a different approach to detect if the received frame contains valid data. This technique involves binary division of the data bits being sent. The divisor is generated using polynomials. The sender performs a division operation on the bits being sent and calculates the remainder. Before sending the actual bits, the sender adds the remainder at the end of the actual bits. Actual data bits plus the remainder is called a codeword. The sender transmits data bits as codeword’s.



At the other end, the receiver performs division operation on codeword’s using the same CRC divisor. If the remainder contains all zeros the data bits are accepted, otherwise it is considered as there some data corruption occurred in transit.

**Error Correction**

In the digital world, error correction can be done in two ways:

* **Backward Error Correction** When the receiver detects an error in the data received, it requests back the sender to retransmit the data unit.
* **Forward Error Correction** When the receiver detects some error in the data received, it executes error-correcting code, which helps it to auto-recover and to correct some kinds of errors.

The first one, Backward Error Correction, is simple and can only be efficiently used where retransmitting is not expensive. For example, fiber optics. But in case of wireless transmission retransmitting may cost too much. In the latter case, Forward Error Correction is used.

To correct the error in data frame, the receiver must know exactly which bit in the frame is corrupted. To locate the bit in error, redundant bits are used as parity bits for error detection. For example, we take ASCII words (7 bits data), then there could be 8 kind of information we need: first seven bits to tell us which bit is error and one more bit to tell that there is no error.

1. **Explain about Hamming Code**

**Hamming Code:**

**Hamming code** is a set of error-correction **code** s that can be used to detect and correct bit errors that can occur when computer data is moved or stored. **Hamming code** is named for R. W. **Hamming** of Bell LabThe hamming code can be applied to data units of any length and uses therelationship between data and redundancy bitsPositions of redundancy bits in hamming code.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |
|  |  |  |

The combinations used to calculate each of the four r values for a seven bit data sequence are as follows:

r1 :1,3,5,7,9,11

r2 : 2,3,6,7,10,11

r3 : 4,5,6,7

r4 : 8,9,10,11

Here, r1 bit is calculated using all bit positions whose binary representation includes a 1 in the rightmost position (0001, 0011, 0101, 0111, 1001, and 1011). The r2 bit is calculated using all bit positions with a 1 in the second position (0010, 0011, 0110, 0111, 1010 and 1011), and for r3 1 at third bit position (0100, 0101, 0110 and 0111) for r4 1 at fourth bit position (1000, 1001, 1010 and 1011).

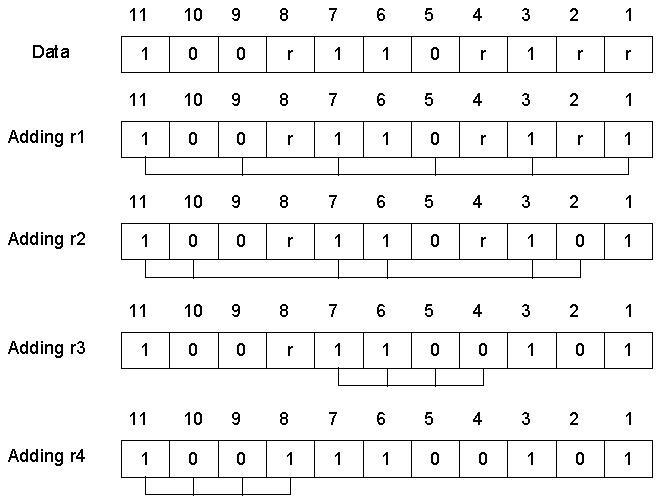
**Calculating the r Values:**

In the first step, we place each bit of the original character in its appropriate positions in the 11 bit unit. Then, we calculate the even parities for the various bit combinations. The parity value of each combination is the value of the corresponding r bit. For example r1 is calculated to provide even parity for a combination of bits 3, 5, 7, 9, 11.

**Error Detection and Correction:**

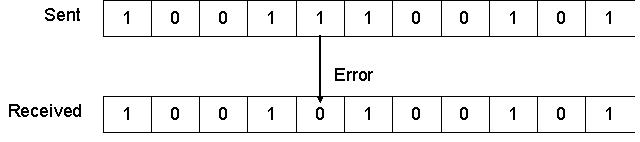
**Example:**

**At the sender:**

Data to be sent: 1001101

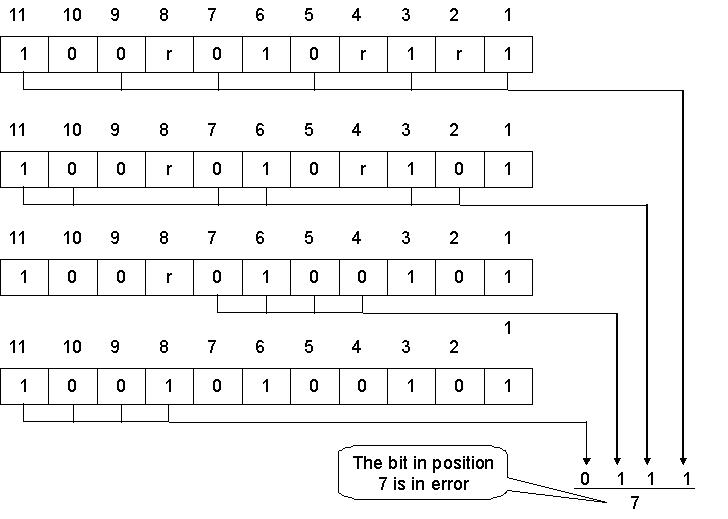
Data sent with redundancy bits: 10011100101

**During transmission:**



**At the receiver:**

The receiver takes the transmission and recalculates four new r values using the same set of bits used by the sender plus the relevant parity (r) bit for each set. Then it assembles the new parity values into a binary number in order of r position (r8, r4, r2, r1).



Once the bit is identified, the receiver can reverse its value and correct the error.

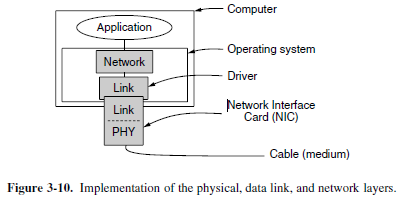
1. **Explain about elementary data link protocols**

The physical layer, data link layer, and networklayer are independent processes that communicate by passing messages back andforth. A common implementation is shown in Fig. 3-10. The physical layer processand some of the data link layer process run on dedicate hardware called a **NIC**(**Network Interface Card**). The rest of the link layer process and the networklayer process run on the main CPU as part of the operating system, with the softwarefor the link layer process often taking the form of a **device driver**. However,other implementations are also possible (e.g., three processes offloaded to dedicatedhardware called a **network accelerator**, or three processes running on themain CPU on a software-defined ratio). Actually, the preferred implementationchanges from decade to decade with technology trade-offs. In any event, treatingthe three layers as separate processes makes the discussion conceptually cleanerand also serves to emphasize the independence of the layers.

Another key assumption is that machine *A* wants to send a long stream of datato machine *B*, using a reliable, connection-oriented service. Later, we will considerthe case where *B* also wants to send data to *A* simultaneously. *A* is assumed tohave an infinite supply of data ready to send and never has to wait for data to beproduced. Instead, when *A*’s data link layer asks for data, the network layer is alwaysable to comply immediately. (This restriction, too, will be dropped later.)We also assume that machines do not crash. That is, these protocols deal withcommunication errors, but not the problems caused by computers crashing andrebooting.

As far as the data link layer is concerned, the packet passed across the interfaceto it from the network layer is pure data, whose every bit is to be delivered tothe destination’s network layer. The fact that the destination’s network layer mayinterpret part of the packet as a header is of no concern to the data link layer.

When the data link layer accepts a packet, it encapsulates the packet in aframe by adding a data link header and trailer to it (see Fig. 3-1). Thus, a frameconsists of an embedded packet, some control information (in the header), and achecksum (in the trailer). The frame is then transmitted to the data link layer onthe other machine. We will assume that there exist suitable library procedures*to physical layer* to send a frame and *from physical layer* to receive a frame.

****

**Simplex Protocol**

As an initial example we will consider a protocol that is as simple as it can bebecause it does not worry about the possibility of anything going wrong. Data aretransmitted in one direction only. Both the transmitting and receiving networklayers are always ready. Processing time can be ignored. Infinite buffer space isavailable. And best of all, the communication channel between the data link layersnever damages or loses frames. This thoroughly unrealistic protocol, whichwe will nickname ‘‘Utopia,’’ is simply to show the basic structure on which wewill build. It’s implementation is shown in Fig. 3-12.

The protocol consists of two distinct procedures, a sender and a receiver. Thesender runs in the data link layer of the source machine, and the receiver runs inthe data link layer of the destination machine. No sequence numbers or acknowledgementsare used here, so *MAX SEQ* is not needed. The only event type possibleis *frame arrival* (i.e., the arrival of an undamaged frame).The sender is in an infinite while loop just pumping data out onto the line asfast as it can. The body of the loop consists of three actions: go fetch a packetfrom the (always obliging) network layer, construct an outbound frame using thevariable *s*, and send the frame on its way. Only the *info* field of the frame is usedby this protocol, because the other fields have to do with error and flow controland there are no errors or flow control restrictions here.

The receiver is equally simple. Initially, it waits for something to happen, theonly possibility being the arrival of an undamaged frame. Eventually, the framearrives and the procedure *wait for event* returns, with *event* set to *frame arrival*(which is ignored anyway). The call to *from physical layer* removes the newlyarrived frame from the hardware buffer and puts it in the variable *r*, where the receivercode can get at it. Finally, the data portion is passed on to the networklayer, and the data link layer settles back to wait for the next frame, effectivelysuspending itself until the frame arrives.

/\* Protocol 1 (Utopia) provides for data transmission in one direction only, from

sender to receiver. The communication channel is assumed to be error free

and the receiver is assumed to be able to process all the input infinitely quickly.

Consequently, the sender just sits in a loop pumping data out onto the line as

fast as it can. \*/

typedefenum {frame arrival} event type;

#include "protocol.h"

void sender1(void)

{

frame s; /\* buffer for an outbound frame \*/

packet buffer; /\* buffer for an outbound packet \*/

while (true) {

from network layer(&buffer); /\* go get something to send \*/

s.info = buffer; /\* copy it into s for transmission \*/

to physical layer(&s); /\* send it on its way \*/

} /\* Tomorrow, and tomorrow, and tomorrow,

Creeps in this petty pace from day to day

To the last syllable of recorded time.

– Macbeth, V, v \*/

}

void receiver1(void)

{

frame r;

event type event; /\* filled in by wait, but not used here \*/

while (true) {

wait for event(&event); /\* only possibility is frame arrival \*/

from physical layer(&r); /\* go get the inbound frame \*/

to network layer(&r.info); /\* pass the data to the network layer \*/

}

}

**Figure 3-12.** A utopian simplex protocol.

The utopia protocol is unrealistic because it does not handle either flow controlor error correction. Its processing is close to that of an unacknowledged connectionlessservice that relies on higher layers to solve these problems, thougheven an unacknowledged connectionless service would do some error detection.

1. **Discuss about Sliding window protocol (nov 2011)**

Data-link layer is responsible for implementation of point-to-point flow and error control mechanism.

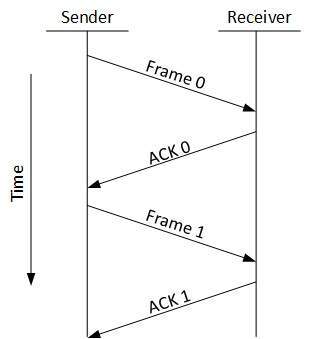
**Flow Control**

When a data frame (Layer-2 data) is sent from one host to another over a single medium, it is required that the sender and receiver should work at the same speed. That is, sender sends at a speed on which the receiver can process and accept the data. What if the speed (hardware/software) of the sender or receiver differs? If sender is sending too fast the receiver may be overloaded, (swamped) and data may be lost.

Two types of mechanisms can be deployed to control the flow:

* **Stop and Wait**

This flow control mechanism forces the sender after transmitting a data frame to stop and wait until the acknowledgement of the data-frame sent is received.



* **Sliding Window**

In this flow control mechanism, both sender and receiver agree on the number of data-frames after which the acknowledgement should be sent. As we learnt, stop and wait flow control mechanism wastes resources, this protocol tries to make use of underlying resources as much as possible.

**Error Control**

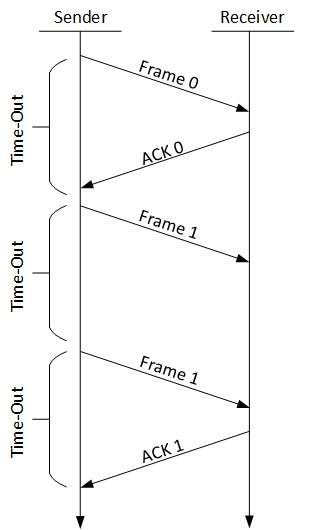
When data-frame is transmitted, there is a probability that data-frame may be lost in the transit or it is received corrupted. In both cases, the receiver does not receive the correct data-frame and sender does not know anything about any loss. In such case, both sender and receiver are equipped with some protocols which helps them to detect transit errors such as loss of data-frame. Hence, either the sender retransmits the data-frame or the receiver may request to resend the previous data-frame.

Requirements for error control mechanism:

* **Error detection** - The sender and receiver, either both or any, must ascertain that there is some error in the transit.
* **Positive ACK** - When the receiver receives a correct frame, it should acknowledge it.
* **Negative ACK** - When the receiver receives a damaged frame or a duplicate frame, it sends a NACK back to the sender and the sender must retransmit the correct frame.
* **Retransmission:**  The sender maintains a clock and sets a timeout period. If an acknowledgement of a data-frame previously transmitted does not arrive before the timeout the sender retransmits the frame, thinking that the frame or it’s acknowledgement is lost in transit.

There are three types of techniques available which Data-link layer may deploy to control the errors by Automatic Repeat Requests (ARQ):

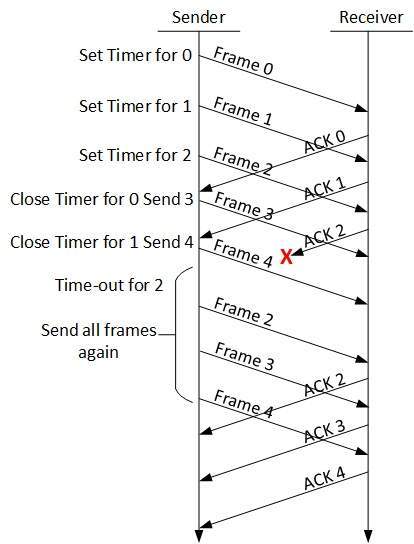
* **Stop-and-wait ARQ**



The following transition may occur in Stop-and-Wait ARQ:

* + The sender maintains a timeout counter.
  + When a frame is sent, the sender starts the timeout counter.
  + If acknowledgement of frame comes in time, the sender transmits the next frame in queue.
  + If acknowledgement does not come in time, the sender assumes that either the frame or its acknowledgement is lost in transit. Sender retransmits the frame and starts the timeout counter.
  + If a negative acknowledgement is received, the sender retransmits the frame.
* **Go-Back-N ARQ**

Stop and wait ARQ mechanism does not utilize the resources at their best.When the acknowledgement is received, the sender sits idle and does nothing. In Go-Back-N ARQ method, both sender and receiver maintain a window.

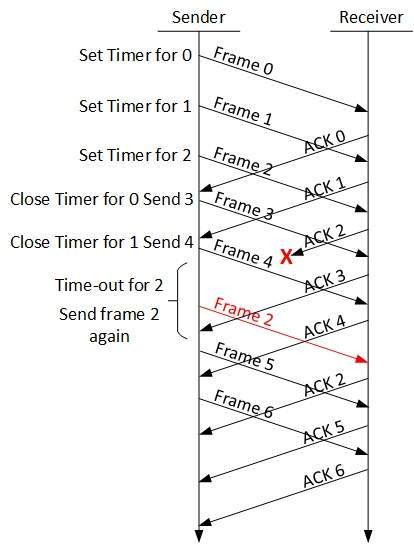


The sending-window size enables the sender to send multiple frames without receiving the acknowledgement of the previous ones. The receiving-window enables the receiver to receive multiple frames and acknowledge them. The receiver keeps track of incoming frame’s sequence number.

When the sender sends all the frames in window, it checks up to what sequence number it has received positive acknowledgement. If all frames are positively acknowledged, the sender sends next set of frames. If sender finds that it has received NACK or has not receive any ACK for a particular frame, it retransmits all the frames after which it does not receive any positive ACK.

* **Selective Repeat ARQ**

In Go-back-N ARQ, it is assumed that the receiver does not have any buffer space for its window size and has to process each frame as it comes. This enforces the sender to retransmit all the frames which are not acknowledged.

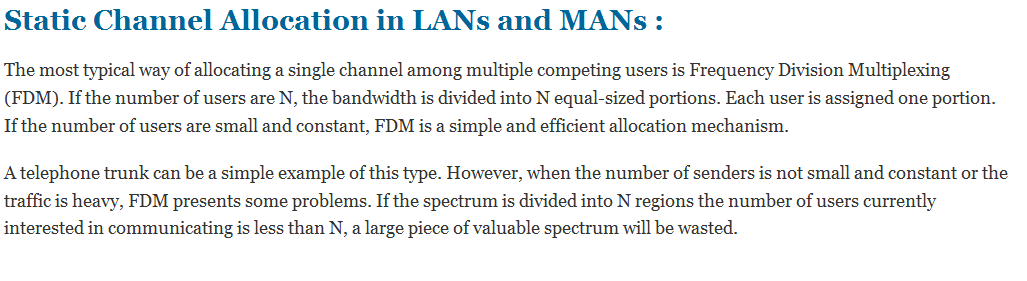


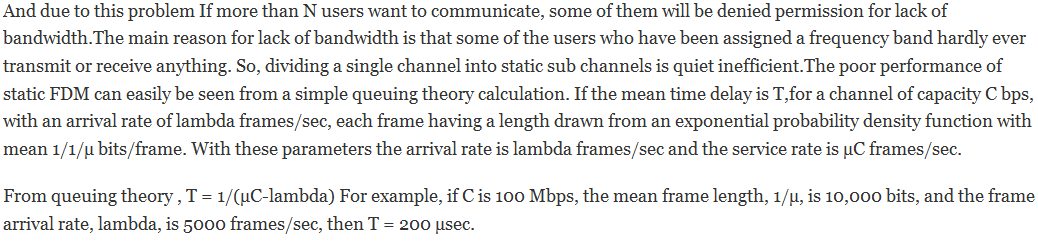
In Selective-Repeat ARQ, the receiver while keeping track of sequence numbers, buffers the frames in memory and sends NACK for only frame which is missing or damaged.

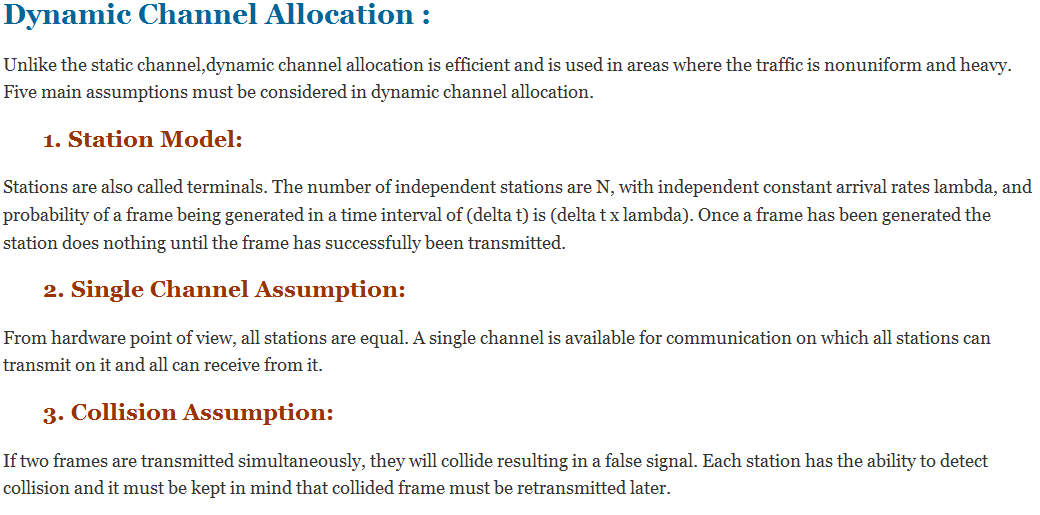
The sender in this case, sends only packet for which NACK is received.

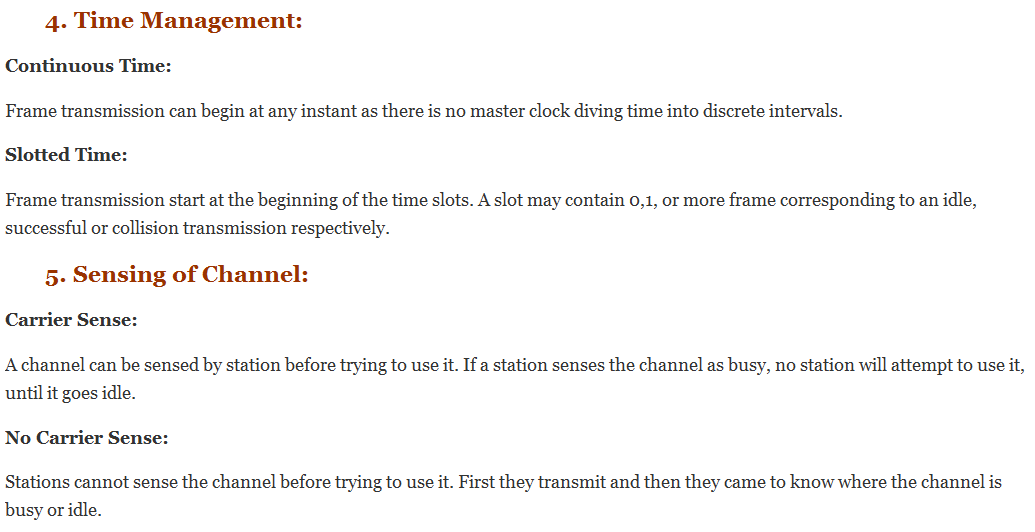
1. **Explain Medium access control sublayer**

**The Channel Allocation Problem**









1. **Explain in detail Multiple Access Protocols (nov 2011) (apr 2011)**

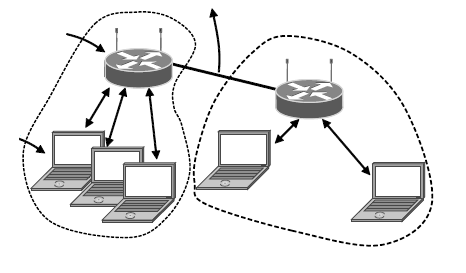
|  |
| --- |
| Aloha, also called the Aloha method, refers to a simple communications scheme in which each source (transmitter) in a network sends data whenever there is a frame to send. If the frame successfully reaches the destination (receiver), the next frame is sent. If the frame fails to be received at the destination, it is sent again.  In a wireless broadcast system or a half-duplex two-way link, Aloha works perfectly. But as networks become more complex, for example in an Ethernet system involving multiple sources and destinations in which data travels many paths at once, trouble occurs because data frames collide (conflict). The heavier the communications volume, the worse the collision problems become.  To minimize the number of collisions, thereby optimizing network efficiency and increasing the number of subscribers that can use a given network, a scheme called slotted Aloha was developed.  Further improvement can be realized by a more sophisticated protocol called Carrier Sense Multiple Access with Collision Detection (CSMA). |
| **The two several of ALOHA are:** |
| * PURE ALOHA * SLOTTED ALOHA  |  | | --- | | **Pure Aloha Protocol**  With Pure Aloha, stations are allowed access to the channel whenever they have data to transmit. Because the threat of data collision exists, each station must either monitor its transmission on the rebroadcast or await an acknowledgment from the destination station. By comparing the transmitted packet with the received packet or by the lack of an acknowledgement, the transmitting station can determine the success of the transmitted packet. If the transmission was unsuccessful it is resent after a random amount of time to reduce the probability of re-collision.  Description: http://www.laynetworks.com/images/aloha14.gif  **Advantages:**   * Superior to fixed assignment when there is a large number of bursty stations. * Adapts to varying number of stations.   **Disadvantages:**   * Theoretically proven throughput maximum of 18.4%. * Requires queueing buffers for retransmission of packets.   **Slotted Aloha Protocol**  By making a small restriction in the transmission freedom of the individual stations, the throughput of the Aloha protocol can be doubled. Assuming constant length packets, transmission time is broken into slots equivalent to the transmission time of a single packet. Stations are only allowed to transmit at slot boundaries. When packets collide they will overlap completely instead of partially. This has the effect of doubling the efficiency of the Aloha protocol and has come to be known as Slotted Aloha.  Description: http://www.laynetworks.com/images/aloha15.gif  **Figure :** Slotted Aloha Protocol  **Advantages:**   * Doubles the efficiency of Aloha. * Adaptable to a changing station population.   **Disadvantages:**   * Theoretically proven throughput maximum of 36.8%. * Requires queueing buffers for retransmission of packets. | |  |   **Carrier Sensed Multiple Accesses (CSMA):**CSMA is a network access method used on shared network topologies such as Ethernet to control access to the network. Devices attached to the network cable listen (carrier sense) before transmitting. If the channel is in use, devices wait before transmitting. MA (Multiple Access) indicates that many devices can connect to and share the same network. All devices have equal access to use the network when it is clear.  [Description: Sharing time with CSMA method](http://ecomputernotes.com/images/Sharing-time-with-CSMA-method.jpg)  **There Are Three Different Type of CSMA Protocols**  (I) I-persistent CSMA  (ii) Non- Persistent CSMA  (iii) p-persistent CSMA  [Description: Type of CSMA](http://ecomputernotes.com/images/Type-of-CSMA.jpg)  **(i) I-persistent CSMA**   * In this method, station that wants to transmit data continuously senses the channel to check whether the channel is idle or busy. * If the channel is busy, the station waits until it becomes idle. * When the station detects an idle-channel, it immediately transmits the frame with probability 1. Hence it is called I-persistent CSMA. * This method has the highest chance of collision because two or more stations may find channel to be idle at the same time and transmit their frames. * When the collision occurs, the stations wait a random amount of time and start allover again.   [Description: 1 Persistent CSMA](http://ecomputernotes.com/images/1-Persistent-CSMA.jpg)  **(ii) Non-persistent CSMA**   * In this scheme, if a station wants to transmit a frame and it finds that the channel is busy (some other station is transmitting) then it will wait for fixed interval of time. * After this time, it again checks the status of the channel and if the channel is.free it will transmit. * A station that has a frame to send senses the channel. * If the channel is idle, it sends immediately. * If the channel is busy, it waits a random amount of time and then senses the channel again. * In non-persistent CSMA the station does not continuously sense the channel for the purpose of capturing it when it detects the end of previous transmission.   [Description: Non persistent](http://ecomputernotes.com/images/Non-persistent.jpg)  **(iii) p-persistent CSMA**   * This method is used when channel has time slots such that the time slot duration is equal to or greater than the maximum propagation delay time. * Whenever a station becomes ready to send, it senses the channel. * If channel is busy, station waits until next slot. * If channel is idle, it transmits with a probability p. * With the probability q=l-p, the station then waits for the beginning of the next time slot. * If the next slot is also idle, it either transmits or waits again with probabilities p and q. * This process is repeated till either frame has been transmitted or another station has begun transmitting. * In case of the transmission by another station, the station acts as though a collision has occurred and it waits a random amount of time and starts again.   [Description: p persistent CSMA](http://ecomputernotes.com/images/p-persistent-CSMA.jpg) CSMA with Collision Avoidance CSMA/CD would break down in wireless networks because of hidden node and exposed nodes problems. Hidden Node Problem In the case of wireless network it is possible that A is sending a message to B, but C is out of its range and hence while "listening" on the network it will find the network to be free and might try to send packets to B at the same time as A. So, there will be a collision at B. The problem can be looked upon as if A and C are hidden from each other. Hence it is called the "hidden node problem". Exposed Node Problem If C is transmitting a message to D and B wants to transmit a message to A, B will find the network to be busy as B hears C trnasmitting. Even if B would have transmitted to A, it would not have been a problem at A or D. CSMA/CD would not allow it to transmit message to A, while the two transmissions could have gone in parallel.  Description: http://www.cse.iitk.ac.in/users/dheeraj/cs425/fig.lec05/image002.gif Addressing hidden node problem (CSMA/CA) Suppose A wants to send a packet to B. Then it will first send a small packet to B called **"Request to Send" (RTS)**. In response, B sends a small packet to A called **"Clear to Send" (CTS)**. Only after A receives a CTS, it transmits the actual data. Now, any of the nodes which can hear either CTS or RTS assume the network to be busy. Hence even if some other node which is out of range of both A and B sends an RTS to C (which can hear at least one of the RTS or CTS between A and B), C would not send a CTS to it and hence the communication would not be established between C and D.  One issue that needs to be addressed is how long the rest of the nodes should wait before they can transmit data over the network. The answer is that the RTS and CTS would carry some information about the size of the data that B intends to transfer. So, they can calculate time that would be required for the transmission to be over and assume the network to be free after that. Another interesting issue is what a node should do if it hears RTS but not corresponding CTS.  One possibility is that it assumes the recipient node has not responded and hence no transmission is going on, but there is a catch in this. It is possible that the node hearing RTS is just on the boundary of the node sending CTS. Hence, it does hear CTS but the signal is so deteriorated that it fails to recognize it as a CTS. Hence to be on the safer side, a node will not start transmission if it hears either of an RTS or CTS. Collision Free ProtocolsBit-Map Method In this method, there N slots. If node 0 has a frame to send, it transmits a 1 bit during the first slot. No other node is allowed to transmit during this period. Next node 1 gets a chance to transmit 1 bit if it has something to send, regardless of what node 0 had transmitted. This is done for all the nodes. In general node j may declare the fact that it has a frame to send by inserting a 1 into slot j. Hence after all nodes have passed, each node has complete knowledge of who wants to send a frame. Now they begin transmitting in numerical order. Since everyone knows who is transmitting and when, there could never be any collision. Binary Countdown In this protocol, a node which wants to signal that it has a frame to send does so by writing its address into the header as a binary number. The arbitration is such that as soon as a node sees that a higher bit position that is 0 in its address has been overwritten with a 1, it gives up. The final result is the address of the node which is allowed to send. After the node has transmitted the whole process is repeated all over again.   |  |  | | --- | --- | | **Nodes** | **Addresses** | | A | 0010 | | B | 0101 | | C | 1010 | | D | 1001 | |  | ---- | |  | 1010 |   Node C having higher priority gets to transmit. The problem with this protocol is that the nodes with higher address always win. Limited Contention Protocols One could combine the best properties of the contention and contention - free protocols, that is, protocol which used contention at low loads to provide low delay, but used a cotention-free technique at high load to provide good channel efficiency. Such protocols do exist and are called Limited contention protocols. Adaptive Tree Walk Protocol The following is the method of adaptive tree protocol. Initially all the nodes are allowed to try to aquire the channel. If it is able to aquire the channel, it sends its frame. If there is collision then the nodes are divided into two equal groups and only one of these groups compete for slot 1. If one of its member aquires the channel then the next slot is reserved for the other group. On the other hand, if there is a collision then that group is again subdivided and the same process is followed. This can be better understood if the nodes are thought of as being organised in a binary tree as shown in the following figure.  Description: http://www.cse.iitk.ac.in/users/dheeraj/cs425/fig.lec05/tree.gif  For example, consider the case of nodes G and H being the only ones wanting to transmit. At slot 1 a collision will be detected and so 2 will be tried and it will be found to be idle. Hence it is pointless to probe 3 and one should directly go to 6,7. |

1. **Discuss about wireless LANs.**

Wireless LANs are increasingly popular, and homes, offices, cafes, libraries, airports, zoos, and other public places are being outfitted with them to connect computers, PDAs, and smart phones to the Internet. Wireless LANs can also be used to let two or more nearby computers communicate without using the Internet.

**The 802.11 Architecture and Protocol Stack**

802.11 networks can be used in two modes. The most popular mode is to connect clients, such as laptops and smart phones, to another network, such as a company intranet or the Internet. In infrastructure mode, each client is associated with an **AP** (**Access Point**) that is in turn connected to the other network. The client sends and receives its packets via the AP. Several access points may be connected together, typically by a wired network called a **distribution system**, to form an extended 802.11 network. In this case, clients can send frames to other clients via their APs. The other mode, shown in Fig is an **ad hoc network**. This mode is a collection of computers that are associated so that they can directly send frames to each other. There is no access point. Since Internet access is the killer application for wireless, ad hoc networks are not very popular.

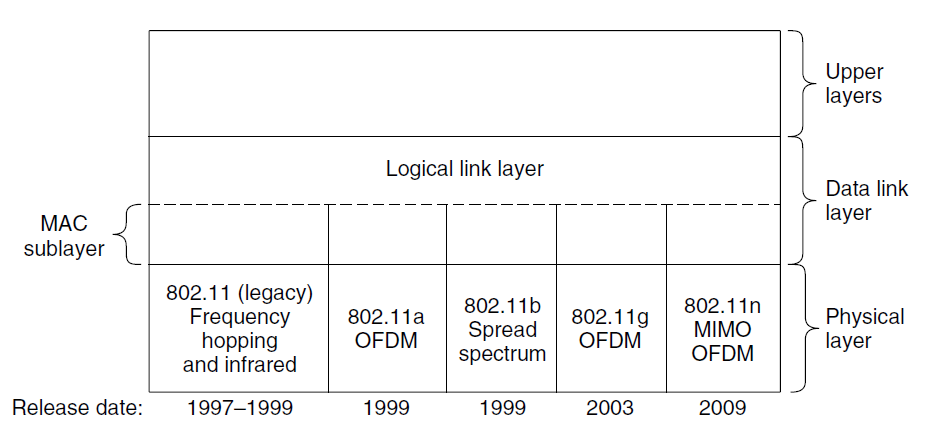


Access  
Point

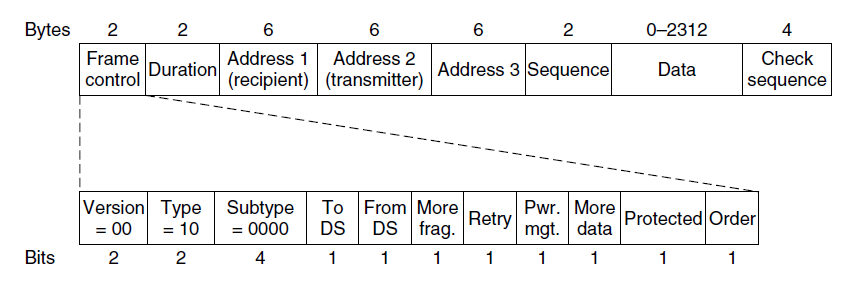
Client

To Network

**802.11 architecture – infrastructure mode**



**802.11 protocol stack**

****

**Format of the 802.11 data frame**

The physical layer corresponds fairly well to the OSI physical layer, but the data link layer in all the 802 protocols is split into two or more sublayers. In 802.11, the MAC (Medium Access Control) sublayer determines how the channel is allocated, that is, who gets to transmit next. Above it is the LLC (Logical Link Control) sublayer, whose job it is to hide the differences between the different 802 variants and make them indistinguishable as far as the network layer is concerned.

Several transmission techniques have been added to the physical layer as 802.11 has evolved since it first appeared in 1997. Two of the initial techniques, infrared in the manner of television remote controls and frequency hopping in the 2.4-GHz band, are now defunct. The third initial technique, direct sequence spread spectrum at 1 or 2 Mbps in the 2.4-GHz band, was extended to run at rates up to 11 Mbps and quickly became a hit. It is now known as 802.11b.

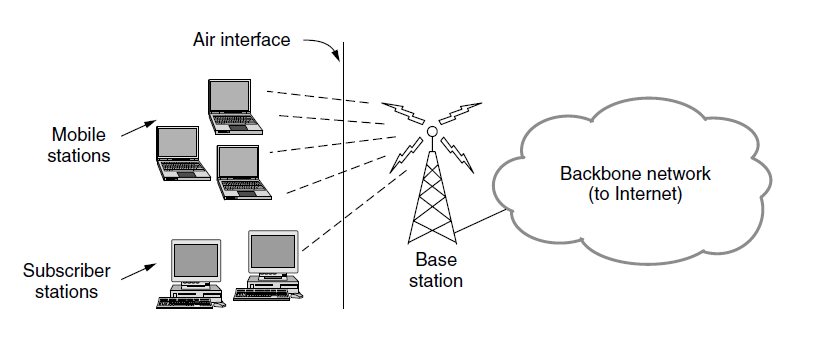
To give wireless junkies a much-wanted speed boost, new transmission techniques based on the OFDM (Orthogonal Frequency Division Multiplexing) scheme we described in Sec. 2.5.3 were introduced in 1999 and 2003. The first is called 802.11a and uses a different frequency band, 5 GHz. The second stuck with 2.4 GHz and compatibility. It is called 802.11g. Both give rates up to 54 Mbps. Most recently, transmission techniques that simultaneously use multiple antennas at the transmitter and receiver for a speed boost were finalized as 802.11n in Oct. 2009. With four antennas and wider channels, the 802.11 standard now defines rates up to a startling 600 Mbps.

**The 802.16 Architecture and Protocol Stack**

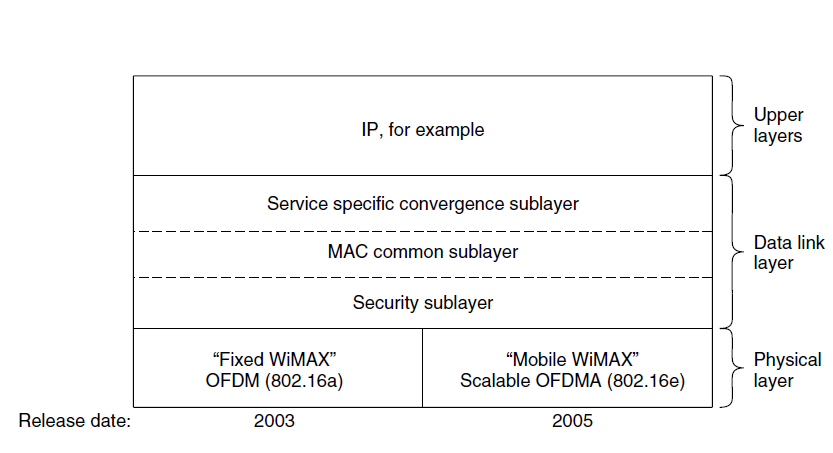
The 802.16 architecture is shown in Fig. Base stations connect directly to the provider’s backbone network, which is in turn connected to the Internet. The base stations communicate with stations over the wireless air interface. Two kinds of stations exist. Subscriber stations remain in a fixed location, for example, broadband Internet access for homes. Mobile stations can receive service while they are moving, for example, a car equipped with WiMAX.

The 802.16 protocol stack that is used across the air interface is shown in Fig. The general structure is similar to that of the other 802 networks, but with more sublayers. The bottom layer deals with transmission, and here we have shown only the popular offerings of 802.16, fixed and mobile WiMAX. There is a different physical layer for each offering. Both layers operate in licensed spectrum below 11 GHz and use OFDM, but in different ways. Above the physical layer, the data link layer consists of three sublayers.

The bottom one deals with privacy and security, which is far more crucial for public outdoor networks than for private indoor networks. It manages encryption, decryption, and key management. Next comes the MAC common sublayer part. This part is where the main protocols, such as channel management, are located. The model here is that the base station completely controls the system. It can schedule the downlink (i.e.,base to subscriber) channels very efficiently and plays a major role in managing the uplink (i.e., subscriber to base) channels as well.



**The 802.16 architecture**

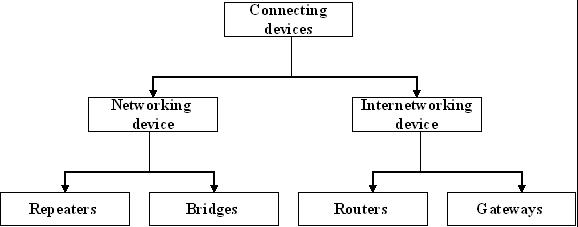


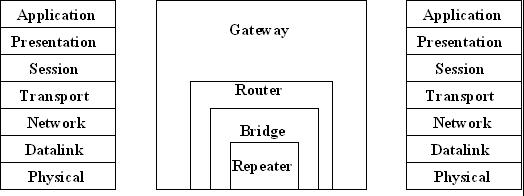
**The 802.16 protocol stack**

An unusual feature of this MAC sublayer is that, unlike those of the other 802 protocols, it is completely connection oriented, in order to provide quality of service guarantees for telephony and multimedia communication.The service-specific convergence sublayer takes the place of the logical link sublayer in the other 802 protocols. Its function is to provide an interface to the network layer. Different convergence layers are defined to integrate seamlessly with different upper layers. The important choice is IP, though the standard defines mappings for protocols such as Ethernet and ATM too. Since IP is connectionless and the 802.16 MAC sublayer is connection-oriented, this layer must map between addresses and connections.

1. **Brief about Repeaters, Bridges, Routers and Gateway**

Networking and internetworking devices are classified into four categories: repeaters, bridges, routers, and gateways.





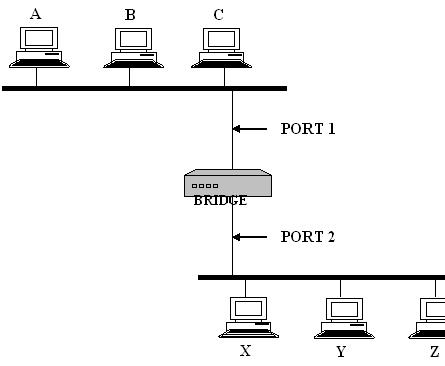
**Bridges and lan switches:**

It is a node that forward frames from one Ethernet to the other. This node would be in promiscuous mode, accepting all frames transmitted on either of the Ethernets, so it could forward them to the other. A bridge is connected between two LANs with port. By using the port number the LANs are addressed. Connected LANs are known as extended LAN

**Learning Bridges:**

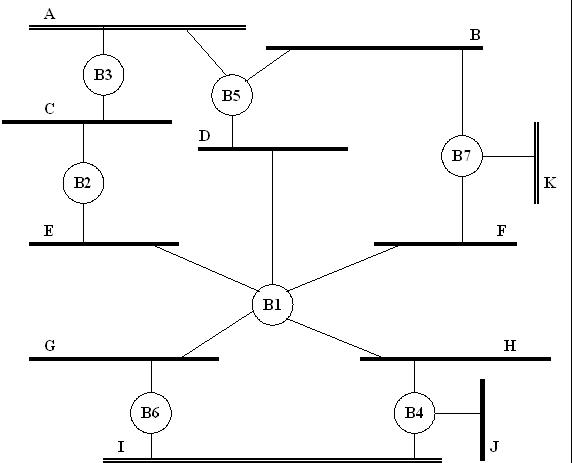
Bridges maintains a forwarding table which contains each host with their port number. Having a human maintain this table is quite a burden, so a bridge can learn this information for itself. The idea is for each bridge to inspect the source address in all the time. Also a timeout is associated with each entry and the bridge is cards the entry after aspecified period of time.

|  |  |
| --- | --- |
| **HOST** | **PORT** |
| A | 1 |
| B | 1 |
| C | 1 |
| X | 2 |
| Y | 2 |
| Z | 2 |



**Spanning Tree Algorithm**

If the extended LAN is having loops then the frames potentially loop through the extended LAN forever. There are two reasons to an extended LAN to have a loop in it. One possibility is that the network is managed by more than one administrator; no single person knows the entire configuration of the network. Second, loops are built in to network on purpose to provide redundancy in case of failure. Bridges must be able to correctly handle loops. This problem is addressed by having the bridges run a distributed spanning tree algorithm.



Thespanning tree algorithm wad developed by Digital Equipment Corporation. The main idea is for the bridges to select the ports over which they will forward frames. The algorithm selects as follows. Each bridge has a unique identifier. In the aboveexample they are labeled as B1, B2, B3 … the algorithm first elects the bridge with smallest ID as the root of the spanning tree. The root bridge always forwards frames out over all of its ports. Then each bridge computes the shortest path to root and notes which of its ports is on this path. This port is also elected as the bridge’s preferred path to the root.

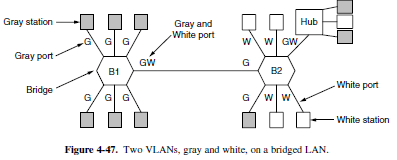
Finally, all the bridges connected to a given LAN elect a single designated bridge that will be responsible for forwarding frames toward the root bridge. Each LANs designated bridge is the one that is closest to the root, and if two or more bridges are equally close to the root, then the bridge which having smallest ID wins. In the above example, B1 is the root bridge since it having the smallest ID. Both B3 and B5 are connected to LAN A, but B5 is the designated bridge since it is closer to the root. Similarly B5 and B7 are connected to LAN B, but B5 is the designated bridge even they are equally closer to the root since B5 having smallest ID.

**VLAN** (**Virtual LAN**)

It has been standardized by the IEEE 802committee and is now widely deployed in many organizations. Let us now take alook at it. For additional information about VLANs, see Seifert and Edwards(2008).

VLANs are based on VLAN-aware switches. To set up a VLAN-based network,the network administrator decides how many VLANs there will be, whichcomputers will be on which VLAN, and what the VLANs will be called. Oftenthe VLANs are (informally) named by colors, since it is then possible to printcolor diagrams showing the physical layout of the machines, with the members ofthe red LAN in red, members of the green LAN in green, and so on. In this way,both the physical and logical layouts are visible in a single view.

As an example, consider the bridged LAN of Fig. 4-47, in which nine of themachines belong to the G (gray) VLAN and five belong to the W (white) VLAN.Machines from the gray VLAN are spread across two switches, including two machinesthat connect to a switch via a hub.



To make the VLANs function correctly, configuration tables have to be set upin the bridges. These tables tell which VLANs are accessible via which ports.When a frame comes in from, say, the gray VLAN, it must be forwarded on allthe ports marked with a G. This holds for ordinary (i.e., unicast) traffic for whichthe bridges have not learned the location of the destination, as well as for multicastand broadcast traffic. Note that a port may be labeled with multiple VLANcolors.

**Hub**

A common connection point for devices in a network. Hubs are commonly used to connect segments of a LAN. A hub contains multiple ports. When a packet arrives at one port, it is copied to the other ports so that all segments of the LAN can see all packets

**GATEWAY**

In the network for an [enterprise](http://searchwinit.techtarget.com/definition/enterprise), a computer [server](http://whatis.techtarget.com/definition/server) acting as a gateway node is often also acting as a [proxy server](http://whatis.techtarget.com/definition/proxy-server) and a [firewall](http://searchsecurity.techtarget.com/definition/firewall) server. A gateway is often associated with both a [router](http://searchnetworking.techtarget.com/definition/router), which knows where to direct a given [packet](http://searchnetworking.techtarget.com/definition/packet) of data that arrives at the gateway, and a [switch](http://searchtelecom.techtarget.com/definition/switch), which furnishes the actual path in and out of the gateway for a given packet.

**Repeaters**

Repeaters remove the unwanted [noise](http://whatis.techtarget.com/definition/noise) in an incoming signal. Unlike an [analog](http://searchcio-midmarket.techtarget.com/definition/analog) signal, the original digital signal, even if weak or distorted, can be clearly perceived and restored. With analog transmission, signals are restrengthened with *amplifiers* which unfortunately also amplify noise as well as information.

1. **WHAT ARE THE SERVICES OFFERED BY A LINK LAYER PROTOCOL AND EXPLAIN (NOV 2012)**

Data Link Layer is second layer of OSI Layered Model. This layer is one of the most complicated layers and has complex functionalities and liabilities. Data link layers hides the details of underlying hardware and represents itself to upper layer as the medium to communicate.

Data link layer works between two hosts which are directly connected in some sense. This direct connection could be point to point or broadcast. Systems on broadcast network are said to beon same link. The work of data link layer tends to get more complex when it is dealing with multiple hosts on single collision domain.

Data link layer is responsible for converting data stream to signals bit by bit and to send that over the underlying hardware. At the receiving end, Data link layer picks up data from hardware which are in the form of electrical signals, assembles them in a recognizable frame format, and hands over to upper layer. Data link layer has two sub-layers:

* **Logical Link Control:** Deals with protocols, flow-control and error control
* **Media Access Control:** Deals with actual control of media

**Functionality of Data-link Layer**

Data link layer does many tasks on behalf of upper layer. These are:

* **Framing:**

Data-link layer takes packets from Network Layer and encapsulates them into Frames. Then, sends each Frame bit-by-bit on the hardware. At receiver’s end Data link layer picks up signals from hardware and assembles them into frames.

* **Addressing:**

Data-link layer provides layer-2 hardware addressing mechanism. Hardware address is assumed to be unique on the link. It is encoded into hardware at the time of manufacturing.

* **Synchronization:**

When data frames are sent on the link, both machines must be synchronized in order to transfer to take place.

* **Error Control:**

Sometimes signals may have encountered problem in transition and bits are flipped. These error are detected and attempted to recover actual data bits. It also provides error reporting mechanism to the sender.

* **Flow Control:**

Stations on same link may have different speed or capacity. Data-link layer ensures flow control that enables both machine to exchange data on same speed.

* **Multi-Access:**

Hosts on shared link when tries to transfer data, has great probability of collision. Data-link layer provides mechanism like CSMA/CD to equip capability of accessing a shared media among multiple Systems

**PONDICHERRY UNIVERSITY QUESTIONS**

**2MARKS**

1. What are adaptive algorithms?**(APR 2011)( Ref.Qn.No.41, Pg.no.6)**
2. Define ARP?**(APR 2012) (Ref.Qn.No.26, Pg.no.6)**
3. What is frame bursting? **(NOV 2011)( Ref.Qn.No.39, Pg.no.6)**
4. Define FDDI? **(NOV 2011)( Ref.Qn.No.40, Pg.no.6)**
5. State the purpose of CRC code? **(NOV 2012)( Ref.Qn.No.8, Pg.no.2)**
6. What are the problems overcome by bridge when compared with hub? **(NOV 2012)( Ref.Qn.No.52,**

**Pg.no.7)**

1. Mention the three categories of multiple access protocols. **(APR 2013)(Ref.Qn.No.42, Pg.no.7)**

**11 MARKS**

**REGULAR**

1. Explain Sliding Window Protocol With Example. **(NOV 2011) ( Ref.Qn.No.6, Pg.no.20)**
2. Briefly Explain Multiple Access Protocols? **(NOV 2011) (Ref.Qn.No.8, Pg.no.25)**
3. What are the Services Offered By a Link Layer Protocol and Explain **(NOV 2012) ( Ref.Qn.No.11,**

**Pg.no.37)**

**ARREAR**

1. What are the versions of aloha? Explain them. **(APR 2011) (Ref.Qn.No.8, Pg.no.25)**
2. Explain The Error-Detection And Correction Techniques **(APR 2012) ( Ref.Qn.No.2, Pg.no.13)**