UNIT V Link Layer and Local area network

Introduction and link layer services

Link layer services provided by a link layer protocol are the following :

Framing

Link access

Reliable delivery

Error detection and correction

Although the basic service of any link layer is to move a datagram from one node an adjacent node over a single communication link, the details of the provided service can vary from one link layer protocol to the next.

Framing

Almost all link layer protocols encapsulate each network layer datagram with a link layer frame before transmission over the link. A frame consists of a data field, in which the network layer datagram is inserted, and a number of header fields. The structure of the frame is specified by the link layer protocol. We'll see several different frame formats when we examine specific link layer protocols in this tutorial.

Link access

A medium access control (MAC) protocol specifies the rules by which a frame is transmitted onto the link. For point-to-point links that have single sender at one end of the link and a single receiver at the other end of the link, the MAC protocol is simple (or non-existent) – the sender can send a frame whenever the link is idle. The more interesting case is when multiple nodes share a single broadcast link – the so-called multiple access problem. Here, the MAC protocol serves to coordinate the frame transmission of the many nodes.

Reliable delivery

When a link layer protocol provides reliable delivery service, it guarantees to move each network layer datagram across the link without error. Recall that certain transport layer protocols (TCP) also provide a reliable delivery service. Similar to a transport layer reliable delivery service, a link layer reliable delivery service can be achieved with acknowledgements and retransmissions.

A link layer reliable delivery service is often used for links that are prone to high error rates, such as a wireless link, with the goal of correcting an error locally – on the link where the link error occurs – rather than forcing an end-to-end retransmission of the data by a transport- or application layer protocol. However, link layer reliable delivery can be considered unnecessary overhead for how low bit-error links, including fibre, coax, and many twisted-pair copper links. For this reason, many wired link layer protocols do not provide a reliable delivery service.

Error Detection & Correction

The link layer hardware in a receiving bode can incorrectly decide that a bit in a frame is zero when it was transmitted as a one, and vice versa. Such bit errors are introduces by signal attenuation and electromagnetic noise. Because there is no need to forward a datagram that has an error, many link layer protocols provide a mechanism to detect such bit errors. This is done by having the receiving node perform an error check. I guess you remember that the transport layer and network layer also provide a limited form or error detection – the internet checksum.

Error detection in the link layer is usually more sophisticated and is implemented in hardware. Error correction is similar to error detection, except that a receiver not only detects when bit errors have occurred in the frame but also determines exactly where in the frame the errors have occurred (and then corrects these errors).

Error Detection and Correction Techniques

Error control can be done in two ways

- **Error detection** − Error detection involves checking whether any error has occurred or not. The number of error bits and the type of error does not matter.

- **Error correction** − Error correction involves ascertaining the exact number of bits that has been corrupted and the location of the corrupted bits.

For both error detection and error correction, the sender needs to send some additional bits along with the data bits. The receiver performs necessary checks based upon the additional redundant bits. If it finds that the data is free from errors, it removes the redundant bits before passing the message to the upper layers.

Error Detection Techniques

There are three main techniques for detecting errors in frames: Parity Check, Checksum and Cyclic Redundancy Check (CRC).

Parity Check

The parity check is done by adding an extra bit, called parity bit to the data to make a number of 1s either even in case of even parity or odd in case of odd parity.

While creating a frame, the sender counts the number of 1s in it and adds the parity bit in the following way

- In case of even parity: If a number of 1s is even then parity bit value is 0. If the number of 1s is odd then parity bit value is 1.
- In case of odd parity: If a number of 1s is odd then parity bit value is 0. If a number of 1s is even then parity bit value is 1.

On receiving a frame, the receiver counts the number of 1s in it. In case of even parity check, if the count of 1s is even, the frame is accepted, otherwise, it is rejected. A similar rule is adopted for odd parity check.

The parity check is suitable for single bit error detection only.

Checksum

In this error detection scheme, the following procedure is applied

- Data is divided into fixed sized frames or segments.
- The sender adds the segments using 1’s complement arithmetic to get the sum. It then complements the sum to get the checksum and sends it along with the data frames.
- The receiver adds the incoming segments along with the checksum using 1’s complement arithmetic to get the sum and then complements it.
- If the result is zero, the received frames are accepted; otherwise, they are discarded.

Cyclic Redundancy Check (CRC)

Cyclic Redundancy Check (CRC) involves binary division of the data bits being sent by a predetermined divisor agreed upon by the communicating system. The divisor is generated using polynomials.

- Here, the sender performs binary division of the data segment by the divisor. It then appends the remainder called CRC bits to the end of the data segment. This makes the resulting data unit exactly divisible by the divisor.
- The receiver divides the incoming data unit by the divisor. If there is no remainder, the data unit is assumed to be correct and is accepted. Otherwise, it is understood that the data is corrupted and is therefore rejected.

Error Correction Techniques

Error correction techniques find out the exact number of bits that have been corrupted and as well as their locations. There are two principle ways

- **Backward Error Correction (Retransmission)** − If the receiver detects an error in the incoming frame, it requests the sender to retransmit the frame. It is a relatively simple technique. But it can be efficiently used only where retransmitting is not expensive as in fiber optics and the time for retransmission is low relative to the requirements of the application.
- **Forward Error Correction** − If the receiver detects some error in the incoming frame, it executes error-correcting code that generates the actual frame. This saves bandwidth required for retransmission. It is inevitable in real-time systems. However, if there are too many errors, the frames need to be retransmitted.

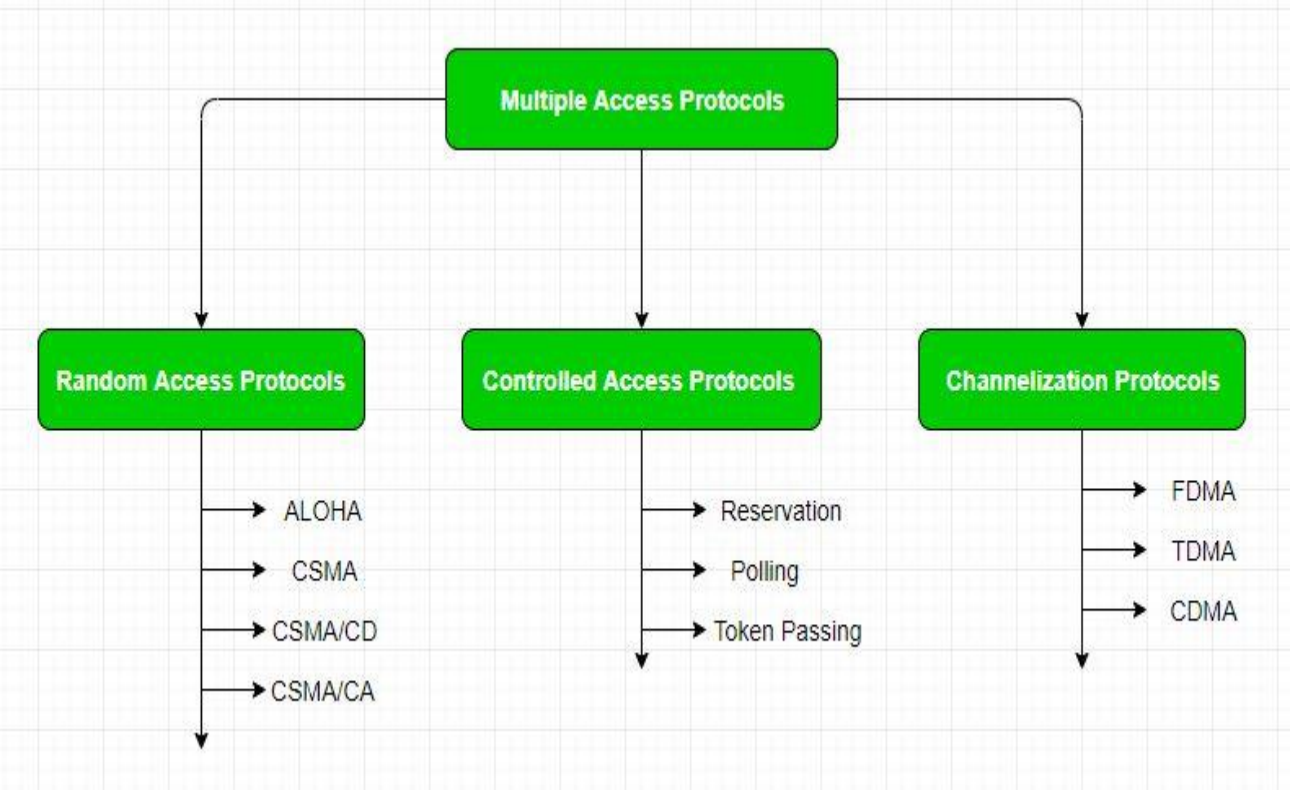
The four main error correction codes are

- Hamming Codes
- Binary Convolution Code
- Reed – Solomon Code
- Low-Density Parity-Check Code

Multiple Access Control –

If there is a dedicated link between the sender and the receiver then data link control layer is sufficient, however if there is no dedicated link present then

multiple stations can access the channel simultaneously. Hence multiple access protocols are required to decrease collision and avoid crosstalk. For example, in a classroom full of students, when a teacher asks a question and all the students (or stations) start answering simultaneously (send data at same time) then a lot of chaos is created(data overlap or data lost) then it is the job of the teacher (multiple access protocols) to manage the students and make them answer one at a time. Thus, protocols are required for sharing data on non dedicated channels. Multiple access protocols can be subdivided further as –



1. Random Access Protocol: In this, all stations have same superiority that is no station has more priority than another station. Any station can send data depending on medium's state(idle or busy). It has two features:

1. There is no fixed time for sending data
2. There is no fixed sequence of stations sending data

The Random access protocols are further subdivided as:

(a) ALOHA – It was designed for wireless LAN but is also applicable for shared medium. In this, multiple stations can transmit data at the same time and can hence lead to collision and data being garbled.

- **Pure Aloha:**
  When a station sends data it waits for an acknowledgement. If the acknowledgement doesn't come within the allotted time then the station waits

for a random amount of time called back-off time (Tb) and re-sends the data. Since different stations wait for different amount of time, the probability of further collision decreases.

- **Slotted Aloha:**
  It is similar to pure aloha, except that we divide time into slots and sending of data is allowed only at the beginning of these slots. If a station misses out the allowed time, it must wait for the next slot. This reduces the probability of collision.

(b) CSMA – Carrier Sense Multiple Access ensures fewer collisions as the station is required to first sense the medium (for idle or busy) before transmitting data. If it is idle then it sends data, otherwise it waits till the channel becomes idle. However there is still chance of collision in CSMA due to propagation delay. For example, if station A wants to send data, it will first sense the medium. If it finds the channel idle, it will start sending data. However, by the time the first bit of data is transmitted (delayed due to propagation delay) from station A, if station B requests to send data and senses the medium it will also find it idle and will also send data. This will result in collision of data from station A and B.

CSMA access modes-

- **1-persistent:** The node senses the channel, if idle it sends the data, otherwise it continuously keeps on checking the medium for being idle and transmits unconditionally(with 1 probability) as soon as the channel gets idle.
- **Non-Persistent:** The node senses the channel, if idle it sends the data, otherwise it checks the medium after a random amount of time (not continuously) and transmits when found idle.
- **P-persistent:** The node senses the medium, if idle it sends the data with p probability. If the data is not transmitted ((1-p) probability) then it waits for some time and checks the medium again, now if it is found idle then it send with p probability. This repeat continues until the frame is sent. It is used in Wifi and packet radio systems.
- **O-persistent:** Superiority of nodes is decided beforehand and transmission occurs in that order. If the medium is idle, node waits for its time slot to send data.

(c) CSMA/CD – Carrier sense multiple access with collision detection. Stations can terminate transmission of data if collision is detected. For more details refer – Efficiency of CSMA/CD

(d) CSMA/CA – Carrier sense multiple access with collision avoidance. The process of collisions detection involves sender receiving acknowledgement signals.

If there is just one signal(its own) then the data is successfully sent but if there are two signals(its own and the one with which it has collided) then it means a collision has occurred. To distinguish between these two cases, collision must have a lot of impact on received signal. However it is not so in wired networks, so CSMA/CA is used in this case.
CSMA/CA avoids collision by:

1. **Interframe space**
2. **Contention Window**
3. **Acknowledgement**

2. Controlled Access:
In this, the data is sent by that station which is approved by all other stations. For further details refer – Controlled Access Protocols
3. Channelization:
In this, the available bandwidth of the link is shared in time, frequency and code to multiple stations to access channel simultaneously.

- **Frequency Division Multiple Access (FDMA)**
- **Time Division Multiple Access (TDMA)**
- **Code Division Multiple Access (CDMA)**

Addressing

Each station on an Ethernet network (such as a PC, workstation, or printer) has its own network interface card (NIC). The NIC fits inside the station and provides the station with a link-layer address. The Ethernet address is 6 bytes (48 bits), normally written in hexadecimal notation, with a colon between the bytes. For example, the following shows an Ethernet MAC address: 4A:30:10:21:10:1A Transmission of Address Bits The way the addresses are sent out online is different from the way they are written in hexadecimal notation. The transmission is left to right, byte by byte; however, for each byte, the least significant bit is sent first and the most significant bit is sent last. This means that the bit that defines an address as unicast or multicast arrives first at the receiver. This helps the receiver to immediately know if the packet is unicast or multicast. Example Show how the address 47:20:IB:2E:08:EE is sent out online. Solution The address is sent left to right, byte by byte; for each byte, it is sent right to left, bit by bit, as shown below: Hexadecimal 47 20 IB 2E 08 EE Binary 01000111 00100000 00011011 00101110 00001000 11101110 Transmitted ~ 11100010 00000100 11011000 01110100 00010000 01110111

Unicast, Multicast, and Broadcast Addresses

A source address is always a unicast address-the frame comes from only one station. The destination address, however, can be unicast, multicast, or broadcast. Below figure shows how to distinguish a unicast address from a multicast address. If the least significant bit of the first byte in a destination address is 0, the address is unicast; otherwise, it is multicast. Note that with the way the bits are transmitted, the unicast/multicast bit is the first bit which is transmitted or received. The broadcast address is a special case of the multicast address: the recipients are all the stations on the LAN. A broadcast destination address is forty-eight 1s. Example Define the type of the following destination addresses: a. 4A:30:10:21:10:1A b. 47:20:IB:2E:08:EE c. FF:FF:FF:FF:FF:FF Solution To find the type of the address, we need to look at the second hexadecimal digit from the left. If it is even, the address is unicast. If it is odd, the address is multicast. If all digits are Fs, the address is broadcast. Therefore, we have the following: a. This is a unicast address because A in binary is 1010 (even). b. This is a multicast address because 7 in binary is 0111 (odd). c. This is a broadcast address because all digits are Fs in hexadecimal Distinguish Between Unicast, Multicast, and Broadcast Transmission Standard Ethernet uses a coaxial cable (bus topology) or a set of twisted-pair cables with a hub (star topology) as shown in Figure. We need to know that transmission in the standard Ethernet is always broadcast, no matter if the intention is unicast, multicast, or broadcast. In the bus topology, when station A sends a frame to station B, all stations will receive it. In the star topology, when station A sends a frame to station B, the hub will receive it. Since the hub is a passive element, it does not check the destination address of the frame; it regenerates the bits (if they have been weakened) and sends them to all stations except station A. In fact, it floods the network with the frame. The question is, then, how the actual unicast, multicast, and broadcast transmissions are distinguished from each other. The answer is in the way the frames are kept or dropped. ¬ In a unicast transmission, all stations will receive the frame, the intended recipient keeps and handles the frame; the rest discard it. ¬ In a multicast transmission, all stations will receive the frame, the stations that are members of the group keep and handle it; the rest discard it. ¬ In a broadcast transmission, all stations (except the sender) will receive the frame and all stations (except the sender) keep and handle it.

Ethernet

The Ethernet LAN was developed in the 1970s by Robert Metcalfe and David Boggs. Since then, it has gone through four generations: Standard Ethernet (10 Mbps), Fast Ethernet (100 Mbps), Gigabit Ethernet (1 Gbps), and 10

Gigabit Ethernet. STANDARD ETHERNET The original Ethernet technology with the data rate of 10 Mbps as the Standard Ethernet is referred. Although most implementations have moved to other technologies in the Ethernet evolution, there are some features of the Standard Ethernet that have not changed during the evolution. Characteristics let us first discuss some characteristics of the Standard Ethernet. Connectionless and Unreliable Service Ethernet provides a connectionless service, which means each frame sent is independent of the previous or next frame. Ethernet has no connection establishment or connection termination phases. The sender sends a frame whenever it has it; the receiver mayor may not be ready for it. The sender may overwhelm the receiver with frames, which may result in dropping frames. If a frame drops, the sender will not know about it. Since IP, which is using the service of Ethernet, is also connectionless, it will not know about it either. If the transport layer is also a connectionless protocol, such as UDP, the frame is lost and salvation may only come from the application layer. However, if the transport layer is TCP, the sender TCP does not receive acknowledgment for its segment and sends it again. Ethernet is also unreliable like IP and UDP. If a frame is corrupted during transmission and the receiver finds out about the corruption, which has a high level of probability of happening because of the CRC-32, the receiver drops the frame silently. It is the duty of high-level protocols to find out about it. Frame Format The Ethernet frame contains seven fields, as shown in below figure. Preamble This field contains 7 bytes (56 bits) of alternating Os and Is that alert the receiving system to the coming frame and enable it to synchronize its clock if it's out of synchronization. The pattern provides only an alert and a timing pulse. The 56-bit pattern allows the stations to miss some bits at the beginning of the frame. The preamble is actually added at the physical layer and is not (formally) part of the frame. Start frame delimiter (SFD) This field (1 byte: 10101011) signals the beginning of the frame. The SFD warns the station or stations that this is the last chance for synchronization. The last 2 bits are (llh and alert the receiver that the next field is the destination address. This field is actually a flag that defines the beginning of the frame. We need to remember that an Ethernet frame is a variable-length frame. It needs a flag to define the beginning of the frame. The SFD field is also added at the physical layer. Destination address (DA) This field is six bytes (48 bits) and contains the link layer address of the destination station or stations to receive the packet. When the receiver sees its own link-layer address, or a multicast address for a group that the receiver is a member of, or a broadcast address, it encapsulates the data from the frame and passes the data to the upper layer protocol defined by the value of the type field. Source address (SA) This field is also six bytes and contains the link-

layer address of the sender of the packet. Type This field defines the upper-layer protocol whose packet is encapsulated in the frame. This protocol can be IP, ARP, OSPF, and so on. In other words, it serves the same purpose as the protocol field in a datagram and the port number in a segment or user datagram. It is used for multiplexing and DE multiplexing. Data This field carries data encapsulated from the upper-layer protocols. It is a minimum of 46 and a maximum of 1500 bytes. We discuss the reason for these minimum and maximum values shortly. If the data coming from the upper layer is more than 1500 bytes, it should be fragmented and encapsulated in more than one frame. If it is less than 46 bytes, it needs to be padded with extra Os. A padded data frame is delivered to the upper-layer protocol as it is (without removing the padding), which means that it is the responsibility of the upper layer to remove or, in the case of the sender, to add the padding. The upper-layer protocol needs to know the length of its data. For example, a datagram has a field that defines the length of the data. CRC The last field contains error detection information, in this case a CRC-32. The CRC is calculated over the addresses, types, and data field. If the receiver calculates the CRC and finds that it is not zero (corruption in transmission), it discards the frame. Frame Length Ethernet has imposed restrictions on both the minimum and maximum lengths of a frame. The minimum length restriction is required for the correct operation of CSMA/CD. An Ethernet frame needs to have a minimum length of 512 bits or 64 bytes. Part of this length is the header and the trailer. If we count 18 bytes of header and trailer (6 bytes of source address, 6 bytes of destination address, 2 bytes of length or type, and 4 bytes of CRC), then the minimum length of data from the upper layer is 64 - 18 = 46 bytes. If the upper-layer packet is less than 46 bytes, padding is added to make up the difference. The standard defines the maximum length of a frame (without preamble and SFD field) as 1518 bytes. If we subtract the 18 bytes of header and trailer, the maximum length of the payload is 1500 bytes. The maximum length restriction has two historical reasons. First, memory was very expensive when Ethernet was designed; a maximum length restriction helped to reduce the size of the buffer. Second, the maximum length restriction prevents one station from monopolizing the shared medium, blocking other stations that have data to send. Minimum frame length: 64 bytes Minimum data length: 46 bytes Maximum frame length: 1518 bytes Maximum data length: 1500 bytes.