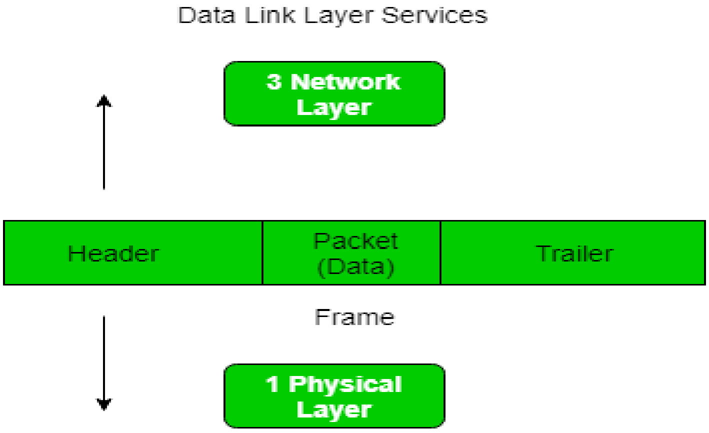
Unit –II

Framing in Data link layer

Framing is a point-to-point connection between two computers or devices consists of a wire in which data is transmitted as a stream of bits. However, these bits must be framed into discernible blocks of information. Framing is a function of the data link layer. It provides a way for a sender to transmit a set of bits that are meaningful to the receiver. Ethernet, token ring, frame relay, and other data link layer technologies have their own frame structures. Frames have headers that contain information such as error-checking codes.



At data link layer, it extracts message from sender and provide it to receiver by providing sender’s and receiver’s address. The advantage of using frames is that data is broken up into recoverable chunks that can easily be checked for corruption.

Problems in Framing –

- **Detecting start of the frame:** When a frame is transmitted, every station must be able to detect it. Station detect frames by looking out for special sequence of bits that marks the beginning of the frame i.e. SFD (Starting Frame Delimeter).
- **How do station detect a frame:** Every station listen to link for SFD pattern through a sequential circuit. If SFD is detected, sequential circuit alerts station. Station checks destination address to accept or reject frame.
- **Detecting end of frame:** When to stop reading the frame.

Types of framing – There are two types of framing:

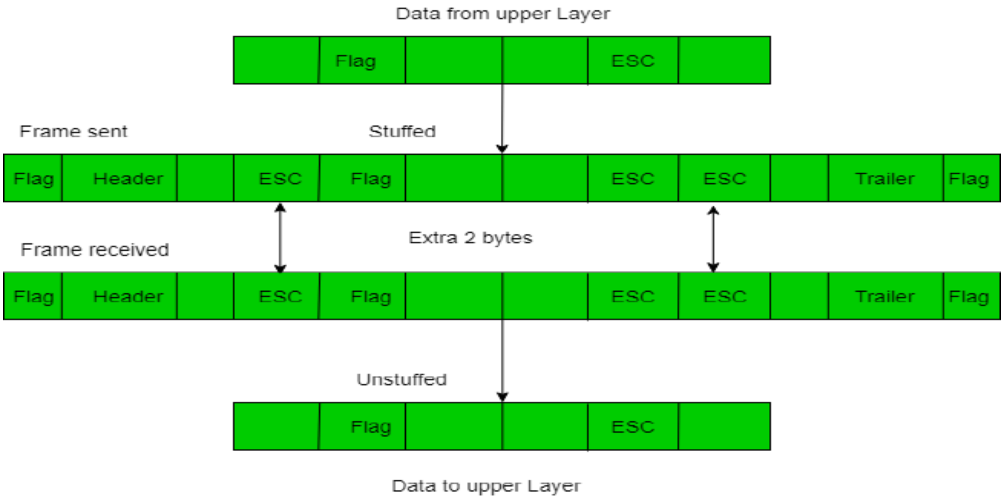
1. Fixed size – The frame is of fixed size and there is no need to provide boundaries to the frame, length of the frame itself acts as delimiter.

- **Drawback:** It suffers from internal fragmentation if data size is less than frame size
- **Solution:** Padding

2. Variable size – In this there is need to define end of frame as well as beginning of next frame to distinguish. This can be done in two ways:

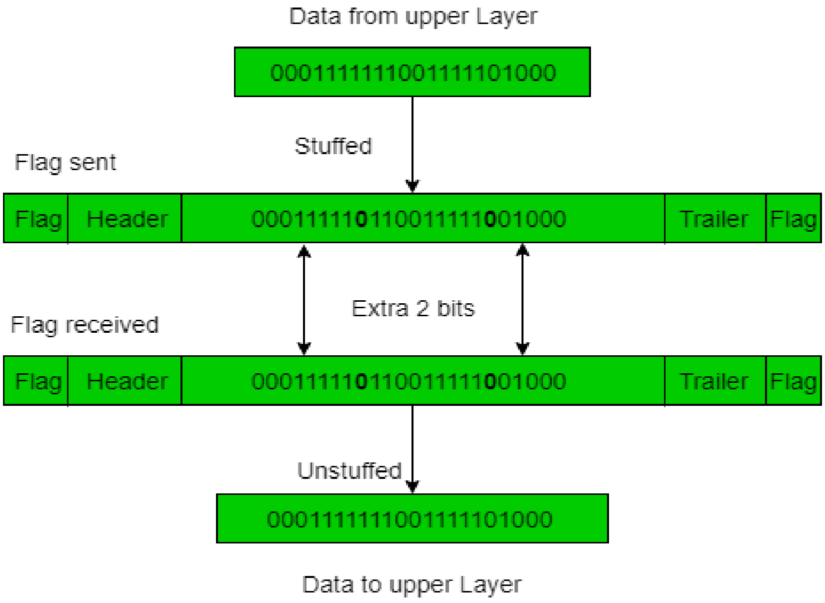
1. **Length field –** We can introduce a length field in the frame to indicate the length of the frame. Used in **Ethernet(802.3)**. The problem with this is that sometimes the length field might get corrupted.
2. **End Delimeter (ED) –** We can introduce an ED(pattern) to indicate the end of the frame. Used in **Token Ring**. The problem with this is that ED can occur in the data. This can be solved by:

1. Character/Byte Stuffing: Used when frames consist of character. If data contains ED then, byte is stuffed into data to diffentiate it from ED.
Let ED = “$” –> if data contains ‘$’ anywhere, it can be escaped using ‘\O’ character.
–> if data contains ‘\O$’ then, use ‘\O\O\O$'($ is escaped using \O and \O is escaped using \O).



Disadvantage – It is very costly and obsolete method.

2. Bit Stuffing: Let ED = 01111 and if data = 01111
–> Sender stuffs a bit to break the pattern i.e. here appends a 0 in data = 0111**0**1.
–> Receiver receives the frame.
–> If data contains 011101, receiver removes the 0 and reads the data.



Examples –

- If Data –> 011100011110 and ED –> 01111 then, find data after bit stuffing ?
  –> 0111**0**000111**0**10
- If Data –> 110001001 and ED –> 1000 then, find data after bit stuffing ?
  –> 1100**1**01001**1**1

3. Error Control

The bit stream transmitted by the physical layer is not guaranteed to be error free. The data link layer is responsible for error detection and correction. The most common error control method is to compute and append some form of a checksum to each outgoing frame at the sender's data link layer and to recompute the checksum and verify it with the received checksum at the receiver's side.

4.Flow Control

Consider a situation in which the sender transmits frames faster than the receiver can accept them. If the sender keeps pumping out frames at high rate, at some point the receiver will be completely swamped and will start losing some frames. This problem may be solved by introducing flow control. Most flow control protocols contain a feedback mechanism to inform the sender when it should transmit the next frame.

Data Link Layer Protocols

An unrestricted simplex protocol

In order to appreciate the step by step development of efficient and complex protocols such as SDLC, HDLC etc., we will begin with a simple but unrealistic protocol. In this protocol:

- Data are transmitted in one direction only
- The transmitting (Tx) and receiving (Rx) hosts are always ready
- Processing time can be ignored
- Infinite buffer space is available
- No errors occur; i.e. no damaged frames and no lost frames (perfect channel)

The protocol consists of two procedures, a sender and receiver as depicted below:

```
/* protocol 1 */

Sender()
{
    forever
    {
        from_host(buffer);
        S.info = buffer;
        sendf(S);
    }
}

Receiver()
{
    forever
    {
        wait(event);
        getf(R);
        to_host(R.info);
    }
}
```

A simplex stop-and-wait protocol
In this protocol we assume that

- Data are transmitted in one direction only
- No errors occur (perfect channel)
- The receiver can only process the received information at a finite rate

These assumptions imply that the transmitter cannot send frames at a rate faster than the receiver can process them.

The problem here is how to prevent the sender from flooding the receiver.

A general solution to this problem is to have the receiver provide some sort of feedback to the sender. The process could be as follows: The receiver send an acknowledge frame back to the sender telling the sender that the last received frame has been processed and passed to the host; permission to send the next frame is granted. The sender, after having sent a frame, must wait for the acknowledge frame from the receiver before sending another frame. This protocol is known as *stop-and-wait*.

The protocol is as follows:

```
/* protocol 2 */

Sender()
{
    forever
    {
        from_host(buffer);
        S.info = buffer;
        sendf(S);
        wait(event);
    }
}

Receiver()
{
    forever
    {
        wait(event);
        getf(R);
        to_host(R.info);
        sendf(S);
    }
}
```

A simplex protocol for a noisy channel

In this protocol the unreal "error free" assumption in protocol 2 is dropped. Frames may be either damaged or lost completely. We assume that transmission errors in the frame are detected by the hardware checksum.

One suggestion is that the sender would send a frame, the receiver would send an ACK frame only if the frame is received correctly. If the frame is in error the receiver simply ignores it; the transmitter would time out and would retransmit it.

One fatal flaw with the above scheme is that if the ACK frame is lost or damaged, duplicate frames are accepted at the receiver without the receiver knowing it.

Imagine a situation where the receiver has just sent an ACK frame back to the sender saying that it correctly received and already passed a frame to its host. However, the ACK frame gets lost completely, the sender times out and retransmits the frame. There is no way for the receiver to tell whether this frame is a retransmitted frame or a new frame, so the receiver accepts this duplicate happily and transfers it to the host. The protocol thus fails in this aspect.

To overcome this problem it is required that the receiver be able to distinguish a frame that it is seeing for the first time from a retransmission. One way to achieve this is to have the sender put a sequence number in the header of each frame it sends. The receiver then can check the sequence number of each arriving frame to see if it is a new frame or a duplicate to be discarded.

The receiver needs to distinguish only 2 possibilities: a new frame or a duplicate; a 1-bit sequence number is sufficient. At any instant the receiver expects a particular sequence number. Any wrong sequence numbered frame arriving at the receiver is rejected as a duplicate. A correctly numbered frame arriving at the receiver is accepted, passed to the host, and the expected sequence number is incremented by 1 (modulo 2).

The protocol is depicted below:

```
/* protocol 3 */

Sender()
{
    NFTS = 0;           /* NFTS = Next Frame To Send */
    from_host(buffer);
    forever
    {
        S.seq = NFTS;
        S.info = buffer;
        sendf(S);
        start_timer(S.seq);
        wait(event);
        if(event == frame_arrival)
        {
            from_host(buffer);
```

```
                ++NFTS;  /* modulo 2 operation */
            }
        }
}

Receiver()
{
        FE = 0;                 /* FE = Frame Expected */
        forever
        {
            wait(event);
            if(event == frame_arrival)
            {
                getf(R);
                if(R.seq == FE)
                {
                    to_host(R.info);
                    ++FE;  /* modulo 2 operation */
                }
                sendf(S);       /* ACK */
            }
        }
}
```

This protocol can handle lost frames by timing out. The timeout interval has to be long enough to prevent premature timeouts which could cause a "deadlock" situation.

Sliding Window Protocols

Piggybacking technique

In most practical situations there is a need for transmitting data in both directions (i.e. between 2 computers). A full duplex circuit is required for the operation.

If protocol 2 or 3 is used in these situations the data frames and ACK (control) frames in the reverse direction have to be interleaved. This method is acceptable but not efficient. An efficient method is to absorb the ACK frame into the header of the data frame going in the same direction. This technique is known as *piggybacking.*

When a data frame arrives at an IMP (receiver or station), instead of immediately sending a separate ACK frame, the IMP restrains itself and waits until the host passes it the next message. The acknowledgement is then attached to the outgoing data frame using the ACK field in the frame header. In effect, the acknowledgement gets a free ride in the next outgoing data frame.

This technique makes better use of the channel bandwidth. The ACK field costs only a few bits, whereas a separate frame would need a header, the acknowledgement, and a checksum.

An issue arising here is the time period that the IMP waits for a message onto which to piggyback the ACK. Obviously the IMP cannot wait forever and there is no way to tell exactly when the next message is available. For these reasons the waiting period is usually a fixed period. If a new host packet arrives quickly the acknowledgement is piggybacked onto it; otherwise, the IMP just sends a separate ACK frame.

Sliding Window

• Sliding window refers to an imaginary boxes that hold the frames on both sender and receiver side.

• It provides the upper limit on the number of frames that can be transmitted before requiring an acknowledgment.

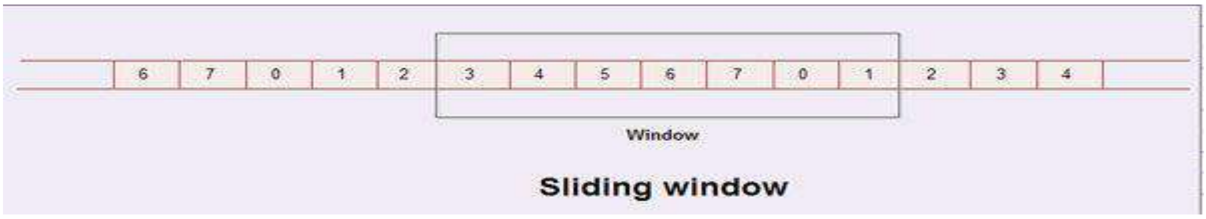
• Frames may be acknowledged by receiver at any point even when window is not full on receiver side.

• Frames may be transmitted by source even when window is not yet full on sender side.

• The windows have a specific size in which the frames are numbered modulo- n, which means they are numbered from 0 to n-l. For e.g. if n = 8, the frames are numbered 0, 1,2,3,4,5,6, 7, 0, 1,2,3,4,5,6, 7, 0, 1, ...

• The size of window is n-1. For e.g. In this case it is 7. Therefore, a maximum of n-l frames may be sent before an acknowledgment.

• When the receiver sends an ACK, it includes the number of next frame it expects to receive. For example in order to acknowledge the group of frames ending in frame 4, the receiver sends an ACK containing the number 5. When sender sees an ACK with number 5, it comes to know that all the frames up to number 4 have been received.



Sliding window

Sliding Window on Sender Side

• At the beginning of a transmission, the sender's window contains n-l frames.

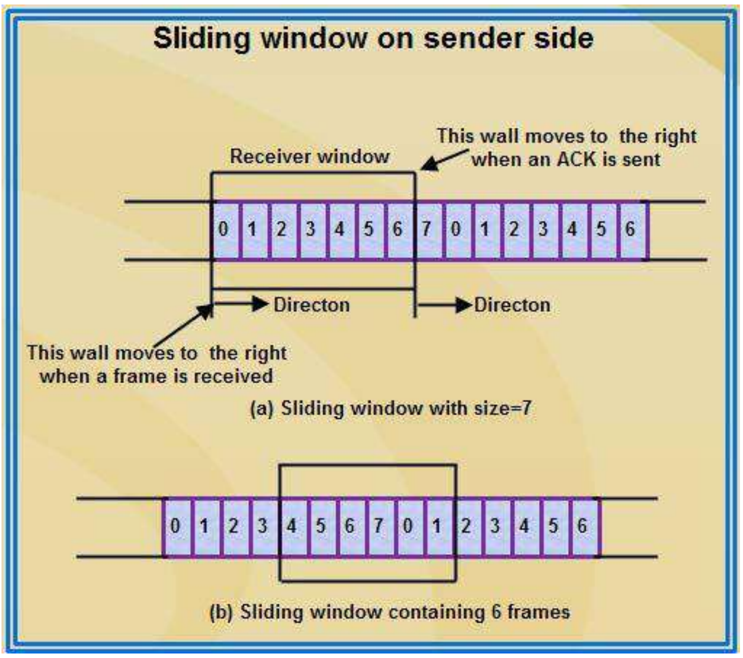
• As the frames are sent by source, the left boundary of the window moves inward, shrinking the size of window. This means if window size is w, if four frames are sent by source after the last acknowledgment, then the number of frames left in window is w-4.

• When the receiver sends an ACK, the source's window expand i.e. (right boundary moves outward) to allow in a number of new frames equal to the number of frames acknowledged by that ACK.

• For example, Let the window size is 7 (see diagram (a)), if frames 0 through 3 have been sent and no acknowledgment has been received, then the sender's window contains three frames - 4,5,6.

• Now, if an ACK numbered 3 is received by source, it means three frames (0, 1, 2) have been received by receiver and are undamaged.

• The sender's window will now expand to include the next three frames in its buffer. At this point the sender's window will contain six frames (4, 5, 6, 7, 0, 1). (See diagram (b)).


Sliding window on sender side
Receiver window
This wall moves to the right when an ACK is sent
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6
Directon
Directon
This wall moves to the right when a frame is received
(a) Sliding window with size=7
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6
(b) Sliding window containing 6 frames

Sliding Window on Receiver Side

- At the beginning of transmission, the receiver's window contains n-1 spaces for frame but not the frames.

- As the new frames come in, the size of window shrinks.

- Therefore the receiver window represents not the number of frames received but the number of frames that may still be received without an acknowledgment ACK must be sent.

- Given a window of size w, if three frames are received without an ACK being returned, the number of spaces in a window is w-3.

- As soon as acknowledgment is sent, window expands to include the number of frames equal to the number of frames acknowledged.

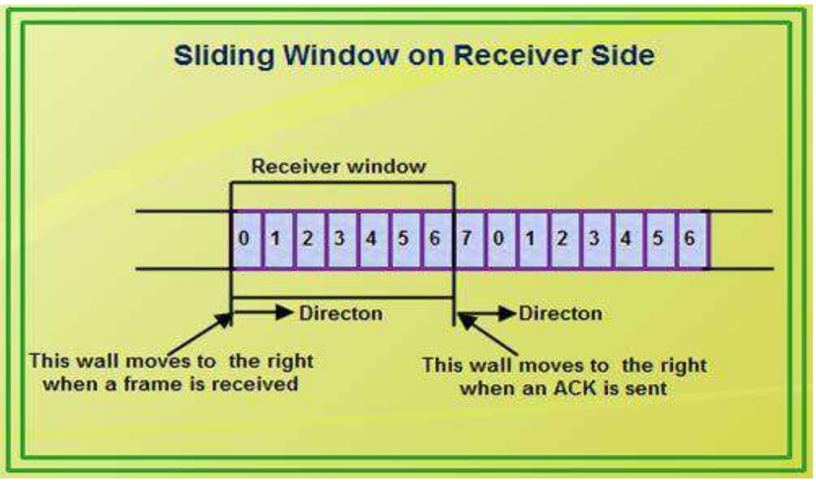
- For example, let the size of receiver's window is 7 as shown in diagram. It means window contains spaces for 7 frames.

- With the arrival of the first frame, the receiving window shrinks, moving the boundary from space 0 to 1. Now, window has shrunk by one, so the receiver may accept six more frame before it is required to send an ACK.

- If frames 0 through 3 have arrived but have DOC been acknowledged, the window will contain three frame spaces.

- As receiver sends an ACK, the window of the receiver expands to include as many new placeholders as newly acknowledged frames.

- The window expands to include a number of new frame spaces equal to the number of the most recently acknowledged frame minus the number of previously acknowledged frame. For *e.g.*, If window size is 7 and if prior ACK was for frame 2 & the current ACK is for frame 5 the window expands by three (5-2).


Sliding Window on Receiver Side
Receiver window
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6
Directon
Directon
This wall moves to the right when a frame is received
This wall moves to the right when an ACK is sent

• Therefore, the sliding window of sender shrinks from left when frames of data are sending. The sliding window of the sender expands to right when acknowledgments are received.

• The sliding window of the receiver shrinks from left when frames of data are received. The sliding window of the receiver expands to the right when acknowledgement is sent.

A one bit sliding window protocol: protocol 4

The sliding window protocol with a maximum window size 1 uses stop-and-wait since the sender transmits a frame and waits for its acknowledgement before sending the next one.

```
/* protocol 4 */

Send_and_receive()
{
    NFTS = 0;
    FE = 0;
    from_host(buffer);
    S.info = buffer;
    S.seq = NFTS;
    S.ack = 1-FE;
    sendf(S);
    start_timer(S.seq);
    forever
    {
        wait(event);
        if(event == frame_arrival)
        {
            getf(R);
            if(R.seq == FE)
            {
                to_host(R.info);
                ++FE;
            }
            if(R.ack == NFTS)
            {
                from_host(buffer);
                ++NFTS;
```

```
                        }
                }
                S.info = buffer;
                S.seq = NFTS;
                S.ack = 1-FE;
                sendf(S);
                start_timer(S.seq);
        }
}
```

Error Detection and Correction

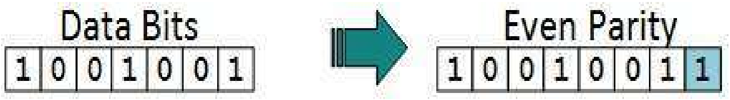
Error Detection

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.

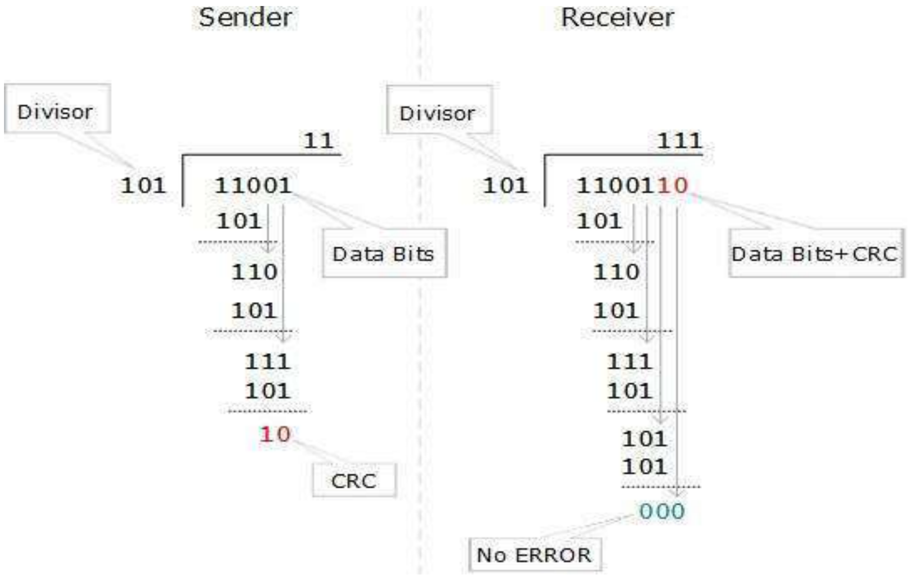


The receiver simply counts the number of 1s in a frame. If the count of 1s is even and even parity is used, the frame is considered to be not-corrupted and is accepted. If the count of 1s is odd and odd parity is used, the frame is still not corrupted.

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.

Cyclic Redundancy Check (CRC)

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 codewords.



At the other end, the receiver performs division operation on codewords 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.

For m data bits, r redundant bits are used. r bits can provide 2r combinations of information. In m+r bit codeword, there is possibility that the r bits themselves may get corrupted. So the number of r bits used must inform about m+r bit locations plus no-error information, i.e. m+r+1.

MAC protocols

Static Channel Allocation

Static Channel Allocation Techniques .Two common static channel allocation techniques are TDMA and FDMA.

Time Division Multiple Access (TDMA) – With TDMA the time axis is divided into time slots of a fixed length. Each user is allocated a fixed set of time slots at which it can transmit. TDMA requires that users be synchronized to a common clock. Typically extra overhead bits are required for synchronization.

Frequency Division Multiple Access (FDMA) – With FDMA the available frequency bandwidth is divided into disjoint frequency bands. A fixed band is allocated to each user. FDMA requires a guard band between user frequency bands to avoid cross-talk.

Another static allocation technique is Code Division Multiple Access (CDMA), this technique is used in many wireless networks.

The performance of static channel allocation depends on:

- The variation in the number of users over time
- The nature of the traffic sent by the user

If the traffic on a shared medium is from a fixed number of sources each transmitting at a fixed rate, static channel allocation can be very efficient. Voice and Video (in their fixed rate forms) have this property and commonly are placed in a shared channel using a static channel allocation. The variation in the number of users over time impacts the performance of a static allocation because some method is needed to allocate the slot to users as they come and go. When the traffic sent by a user is bursty, then, under a static allocation, a user's portion of the channel may be empty when another user could use it. This leads one to think that a dynamic allocation will perform better in such cases.

Dynamic Channel Allocation

The idea of dynamic channel allocation is to redistribute the channel resources among the cells dynamically based on the instantaneous capacity demands.

Ideal channel allocation scheme would have knowledge on the instantaneous load (number of users) in each cell, received signal quality of the users as well as interference coupling among the cells. • This would however, require centralized processing which due to signaling restrictions and delays would be difficult. • Centralized schemes provide knowledge on the performance bounds. • Practical implementation calls for distributed DCA schemes that make the decision based on locally available information

Slow DCA – Resource allocation among the cells is adapted based on long term statistics of the load and interference. – Required signaling load is small, but resources are not always distributed in optimal manner. – Resembles network optimization process

Fast DCA – Resource allocations among the cells are done at the time scale of the calls. – Requires a lot of signaling. Due to signaling restrictions the resource sharing is typically limited to some local neighborhood of cells – e.g. those controlled by the same BSC.

In traffic adaptive DCA, one tries to adapt the allocation of spectral resources among the cells in accordance with the current measured number of active mobiles in each cell. • A purely traffic adaptive DCA uses the same kind of interference analysis than static network planning (FCA) • Instead of splitting the channels into equally sized channel groups, the size of each group is chosen to minimize the number of assignment failures.

MAC Protocols

1.Aloha 2. CSMA 3. CSMA/CD 4. CSMA/CA 5. Ethernet

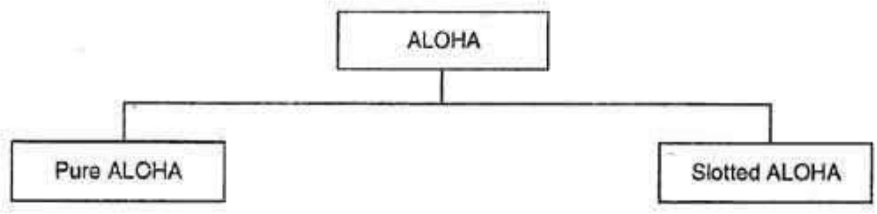
ALOHA Protocols

ALOHA: ALOHA is a system for coordinating and arbitrating access to a shared communication Networks channel. It was developed in the 1970s by Norman Abramson and his colleagues at the University of Hawaii. The original system used for ground based radio broadcasting, but the system has been implemented in satellite communication systems.

A shared communication system like ALOHA requires a method of handling collisions that occur when two or more systems attempt to transmit on the channel at the same time. In the ALOHA system, a node transmits whenever data is available to send. If another node transmits at the same time, a collision occurs, and the frames that were transmitted are lost. However, a node can listen to broadcasts on the medium, even its own, and determine whether the frames were transmitted.

Aloha means "Hello". Aloha is a multiple access protocol at the datalink layer and proposes how multiple terminals access the medium without interference or collision. In 1972 Roberts developed a protocol that would increase the capacity of aloha two fold. The Slotted Aloha protocol involves dividing the time interval into discrete slots and each slot interval corresponds to the time period of one frame. This method requires synchronization between the sending nodes to prevent collisions.

There are two different versions of ALOHA



Types of ALOHA

Pure ALOHA

• **In** pure ALOHA, the stations transmit frames whenever they have data to send.

- When two or more stations transmit simultaneously, there is collision and the frames are destroyed.

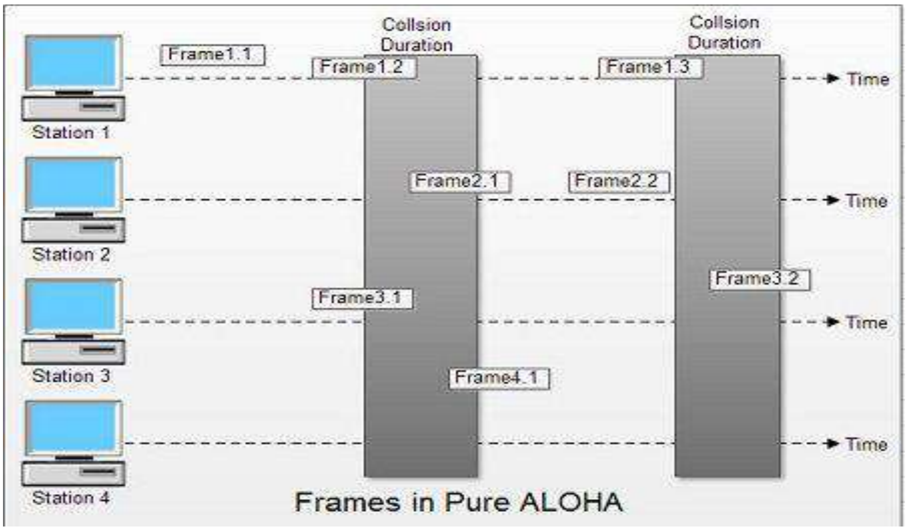
- In pure ALOHA, whenever any station transmits a frame, it expects the acknowledgement from the receiver.

- If acknowledgement is not received within specified time, the station assumes that the frame (or acknowledgement) has been destroyed.

- If the frame is destroyed because of collision the station waits for a random amount of time and sends it again. This waiting time must be random otherwise same frames will collide again and again.

- Therefore pure ALOHA dictates that when time-out period passes, each station must wait for a random amount of time before resending its frame. This randomness will help avoid more collisions.

- Figure shows an example of frame collisions in pure ALOHA.



- In fig there are four stations that .contended with one another for access to shared channel. All these stations are transmitting frames. Some of these frames collide because multiple frames are in contention for the shared channel. Only two frames, frame 1.1 and frame 2.2 survive. All other frames are destroyed.

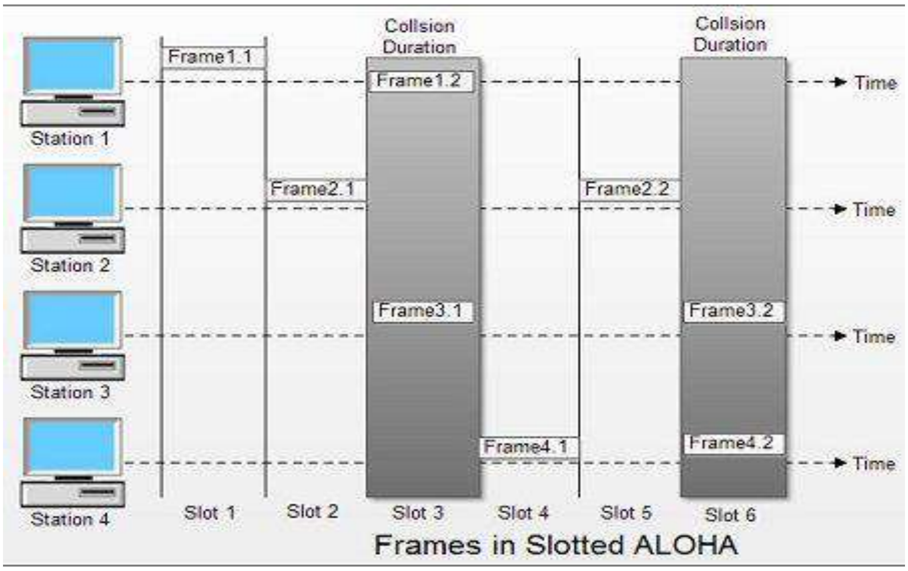
- Whenever two frames try to occupy the channel at the same time, there will be a collision and both will be damaged. If first bit of a new frame overlaps with just the last bit of a frame almost finished, both frames will be totally destroyed and both will have to be retransmitted.

Slotted ALOHA

- Slotted ALOHA was invented to improve the efficiency of pure ALOHA as chances of collision in pure ALOHA are very high.

- In slotted ALOHA, the time of the shared channel is divided into discrete intervals called slots.

- The stations can send a frame only at the beginning of the slot and only one frame is sent in each slot.



- In slotted ALOHA, if any station is not able to place the frame onto the channel at the beginning of the slot *i.e.* it misses the time slot then the station has to wait until the beginning of the next time slot.

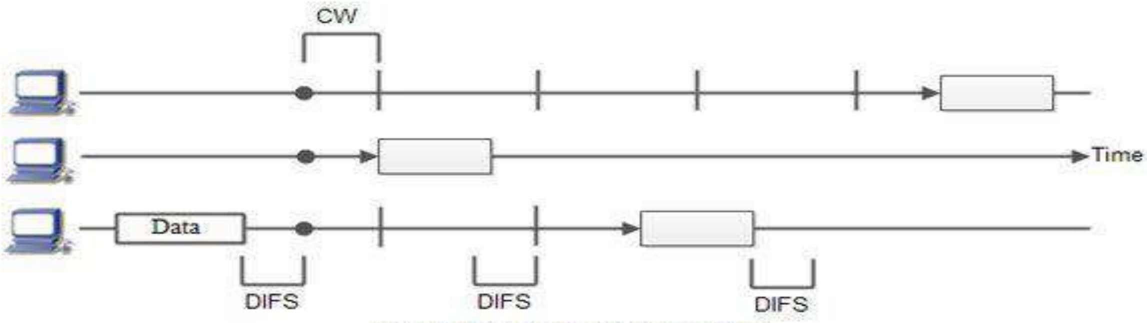
- In slotted ALOHA, there is still a possibility of collision if two stations try to send at the beginning of the same time slot as shown in fig.

- Slotted ALOHA still has an edge over pure ALOHA as chances of collision are reduced to one-half.

CSMA Protocol

Carrier Sensed Multiple Access (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.

In other words, a station that wants to communicate "listen" first on the media communication and awaits a "silence" of a preset time (called the Distributed Inter Frame Space or DIFS). After this compulsory period, the station starts a countdown for a random period considered. The maximum duration of this countdown is called the collision window (Window Collision, CW). If no equipment speaks before the end of the countdown, the station simply deliver its package. However, if it is overtaken by another station, it stops immediately its countdown and waits for the next silence. She then continued his account countdown where it left off. This is summarized in Figure. The waiting time random has the advantage of allowing a statistically equitable distribution of speaking time between the various network equipment, while making little unlikely (but not impossible) that both devices speak exactly the same time. The countdown system prevents a station waiting too long before issuing its package. It's a bit what place in a meeting room when no master session (and all the World's polite) expected a silence, then a few moments before speaking, to allow time for someone else to speak. The time is and randomly assigned, that is to say, more or less equally.



Sharing time with CSMA method.

Again, this is what we do naturally in a meeting room if many people speak exactly the same time, they are realizing account immediately (as they listen at the same time they speak), and they interrupt without completing their sentence. After a while, one of them speaks again. If a new collision occurs, the two are interrupted again and tend to wait a little longer before speaking again.

CSMA protocol was developed to overcome the problem found in ALOHA i.e. to minimize the chances of collision, so as to improve the performance. CSMA protocol is based on the principle of 'carrier sense'. The station senses the carrier or channel before transmitting a frame. It means the station checks the state of channel, whether it is idle or busy.

Even though devices attempt to sense whether the network is in use, there is a good chance that two stations will attempt to access it at the same time. On large networks, the transmission time between one end of the cable and another is enough that one station may access the cable even though another has already just accessed it.

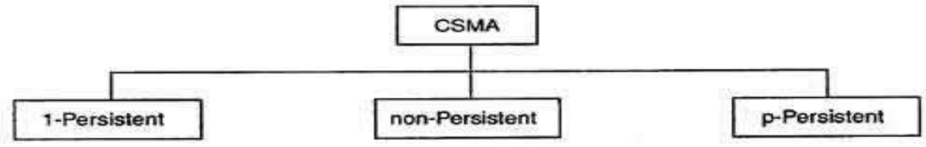
The chances of collision still exist because of propagation delay. The frame transmitted by one station takes some time to reach other stations. In the meantime, other stations may sense the channel to be idle and transmit their frames. This results in the collision.

There Are Three Different Type of CSMA Protocols

(I) I-persistent CSMA

(ii) Non- Persistent CSMA

(iii) p-persistent CSMA



Types of CSMA

(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.

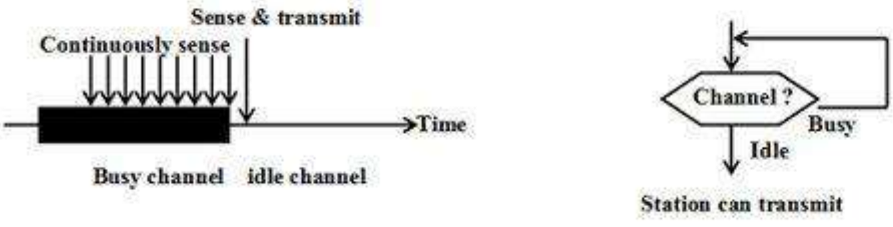
• When the station detects an idle-channel, it immediately transmits the frame with probability 1. Hence it is called I-persistent CSMA.

• 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.

Drawback of I-persistent

• The propagation delay time greatly affects this protocol. Let us suppose, just after the station I begins its transmission, station 2 also became ready to send its data and senses the channel. If the station I signal has not yet reached station 2, station 2 will sense the channel to be idle and will begin its transmission. This will result in collision.



1-persistent CSMA

Even if propagation delay time is zero, collision will still occur. If two stations became .ready in the middle of third station's transmission, both stations will wait until the transmission of first station ends and then both will begin their transmission exactly simultaneously. This will also result in collision.

(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 oftime.
• 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.

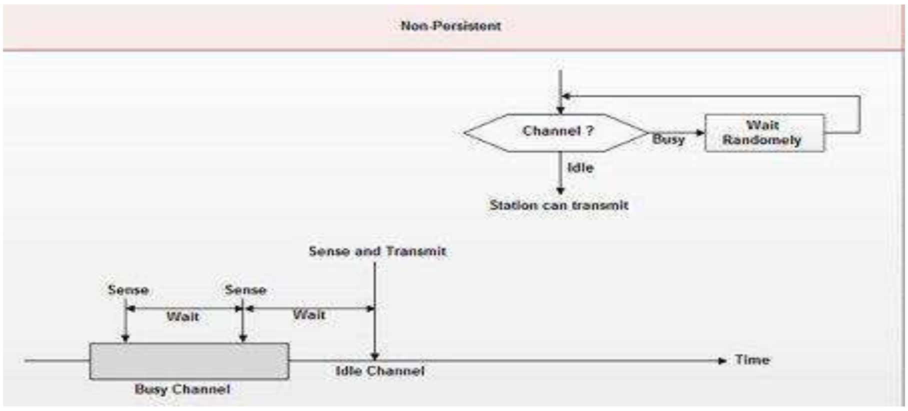
• In non-persistent CSMA the station does not continuously sense the channel for the purpose of capturing it when it detects the end of previous transmission.

Advantage of non-persistent

• It reduces the chance of collision because the stations wait a random amount of time. It is unlikely that two or more stations will wait for same amount of time and will retransmit at the same time.

Disadvantage of non-persistent

• It reduces the efficiency of network because the channel remains idle when there may be stations with frames to send. This is due to the fact that the stations wait a random amount of time after the collision.



(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.

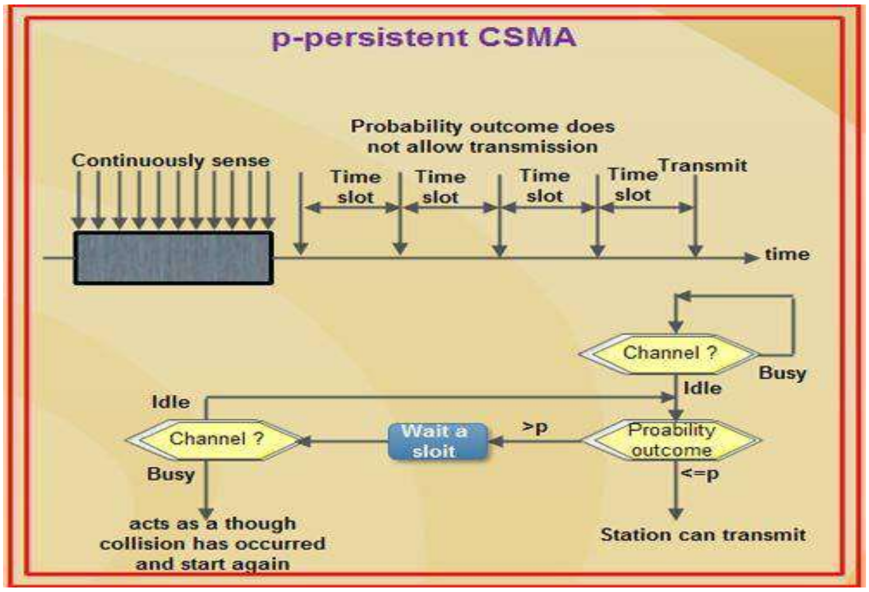
• If channel is idle, it transmits with a probability p.

• 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.



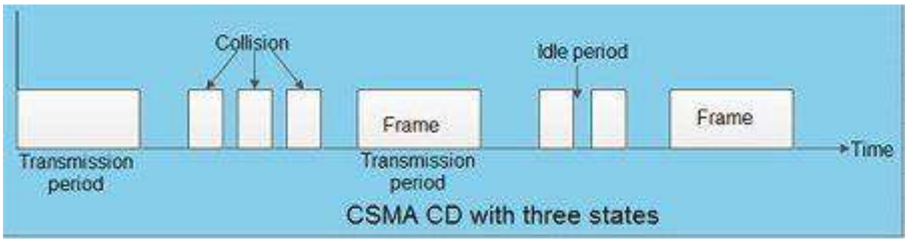
Advantage of p-persistent

• It reduces the chance of collision and improves the efficiency of the network.

CSMA/CD

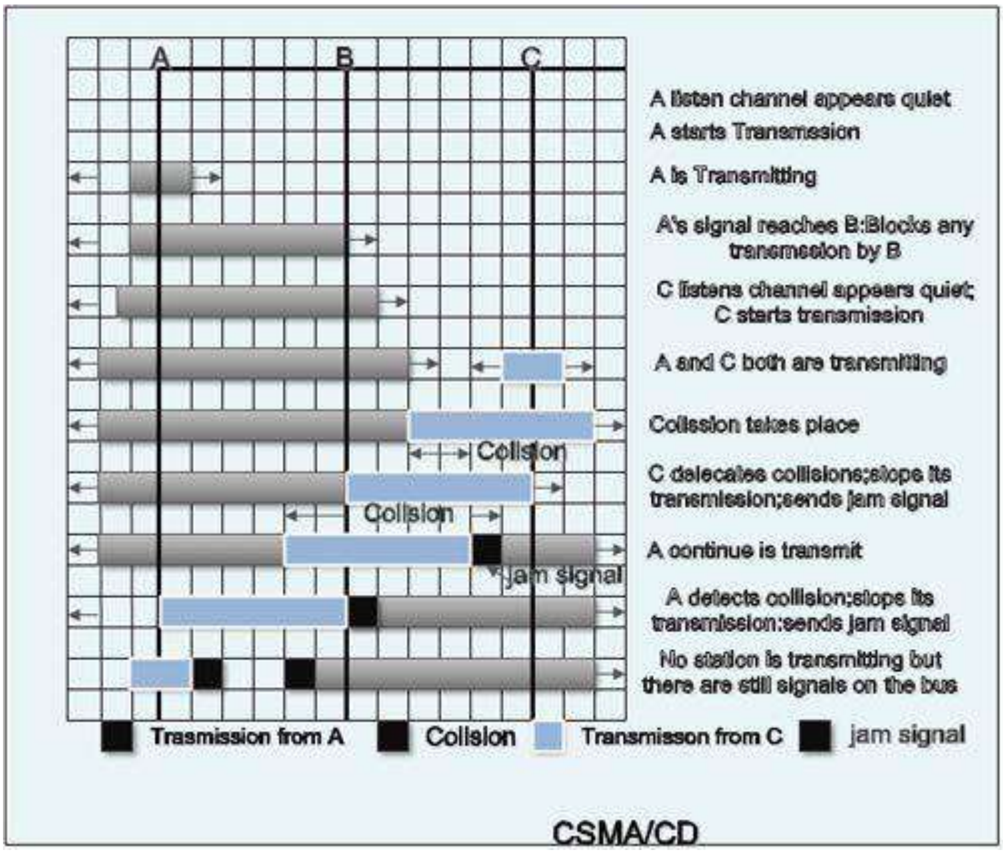
To reduce the impact of collisions on the network performance, Ethernet uses an algorithm called CSMA with Collision Detection (CSMA / CD): CSMA/CD is a protocol in which the station senses the carrier or channel before transmitting frame just as in persistent and non-persistent CSMA. If the channel is busy, the station waits. it listens at the same time on communication media to ensure that there is no collision with a packet sent by another station. In a collision, the issuer immediately cancel the sending of the package. This allows to limit the duration of collisions: we do not waste time to send a packet complete if it detects a collision. After a collision, the transmitter waits again silence and again, he continued his hold for a random number; but this time the random number is nearly double the previous one: it is this called back-off (that is to say, the "decline") exponential. In fact, the window collision is simply doubled (unless it has already reached a maximum). From a packet is transmitted successfully, the window will return to its original size.

Again, this is what we do naturally in a meeting room if many people speak exactly the same time, they are realizing account immediately (as they listen at the same time they speak), and they interrupt without completing their sentence. After a while, one of them speaks again. If a new collision occurs, the two are interrupted again and tend to wait a little longer before speaking again.



CSMA CD with three states

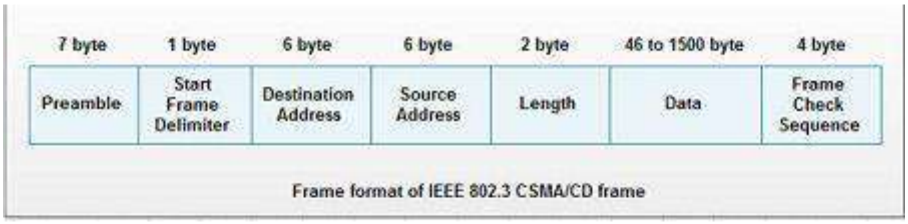
The entire scheme of CSMA/CD is depicted in the fig.


A
B
C
A listen channel appears quiet
A starts Transmssion
A is Transmitting
A's signal reaches B:Blocks any transmssion by B
C listens channel appears quiet; C starts transmission
A and C both are transmitting
Collssion takes place
Collsion
C delecates collisions;slops its transmission;sends jam signal
Collsion
A continue is transmit
jam signal
A detects collision;slops its transmission:sends jam signal
No station is transmitting but there are still signals on the bus
Trasmission from A
Collsion
Transmisson from C
jam signal

CSMA/CD

Frame format of CSMA/CD

The frame format specified by IEEE 802.3 standard contains following fields.



7 byte	1 byte	6 byte	6 byte	2 byte	46 to 1500 byte	4 byte
Preamble	Start Frame Delimiter	Destination Address	Source Address	Length	Data	Frame Check Sequence

Frame format of IEEE 802.3 CSMA/CD frame

1. **Preamble**: It is seven bytes (56 bits) that provides bit synchronization. It consists of alternating Os and 1s. The purpose is to provide alert and timing pulse.

2. **Start Frame Delimiter (SFD)**: It is one byte field with unique pattern: 10 10 1011. It marks the beginning of frame.

3. **Destination Address (DA)**: It is six byte field that contains physical address of packet's destination.

4. **Source Address (SA)**: It is also a six byte field and contains the physical address of source or last device to forward the packet (most recent router to receiver).

5. **Length**: This two byte field specifies the length or number of bytes in data field.

6. **Data**: It can be of 46 to 1500 bytes, depending upon the type of frame and the length of the informationfield.

7. **Frame Check Sequence (FCS)**: This for byte field contains CRC for error detection.

Explanation:

• The station that has a ready frame sets the back off parameter to zero.

• Then it senses the line using one of the persistent strategies.

• If then sends the frame. If there is no collision for a period corresponding to one complete frame, then the transmission is successful.

• Otherwise the station sends the jam signal to inform the other stations about the collision.

• The station then increments the back off time and waits for a random back off time and sends the frame again.

• If the back off has reached its limit then the station aborts the transmission.

• *CSMA/CD* is used for the traditional Ethernet.

• *CSMA/CD* is an important protocol. IEEE 802.3 (Ethernet) is an example of *CSMNCD*. It is an international standard.

• The MAC sublayer protocol does not guarantee reliable delivery. Even in absence of collision the receiver may not have copied the frame correctly.

CSMA/CA

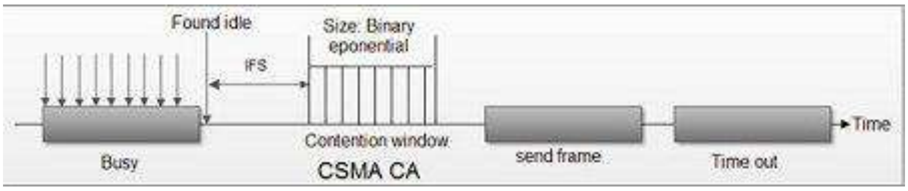
• CSMA/CA protocol is used in wireless networks because they cannot detect the collision so the only solution is collision avoidance.

• CSMA/CA avoids the collisions using three basic techniques.

(i) Interframe space

(ii) Contention window

(iii) Acknowledgements



1. Interframe Space (IFS)

• Whenever the channel is found idle, the station does not transmit immediately. It waits for a period of time called interframe space (IFS).

• When channel is sensed to be idle, it may be possible that same distant station may have already started transmitting and the signal of that distant station has not yet reached other stations.

• Therefore the purpose of IFS time is to allow this transmitted signal to reach other stations.

• If after this IFS time, the channel is still idle, the station can send, but it still needs to wait a time equal to contention time.

• IFS variable can also be used to define the priority of a station or a frame.

2. Contention Window

• Contention window is an amount of time divided into slots.

• A station that is ready to send chooses a random number of slots as its wait time.

• The number of slots in the window changes according to the binary exponential back-off strategy. It means that it is set of one slot the first time and then doubles each time the station cannot detect an idle channel after the IFS time.

• This is very similar to the p-persistent method except that a random outcome defines the number of slots taken by the waiting station.

• In contention window the station needs to sense the channel after each time slot.

• If the station finds the channel busy, it does not restart the process. It just stops the timer & restarts it when the channel is sensed as idle.

3. Acknowledgement

• Despite all the precautions, collisions may occur and destroy the data.

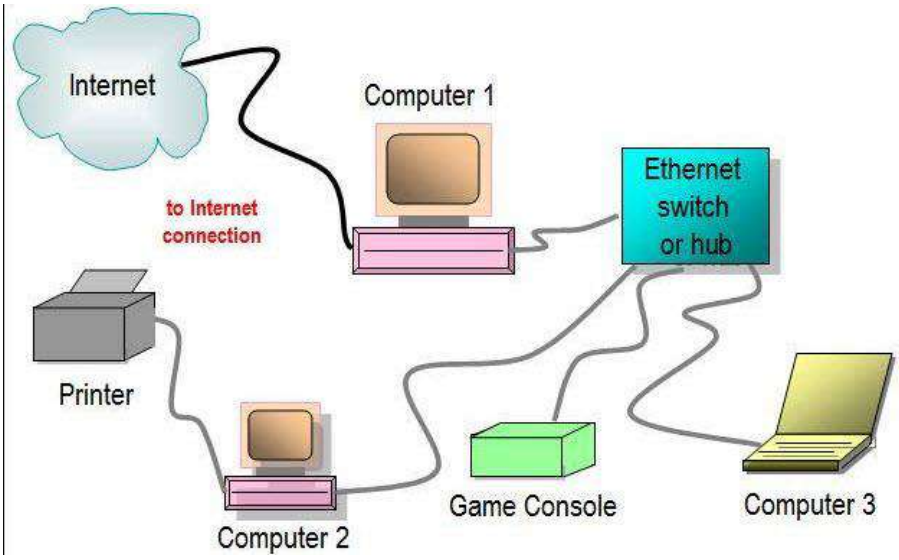
• The positive acknowledgment and the time-out timer can help guarantee that receiver has received the frame.

Ethernet

Ethernet is a type of network cabling and signaling specifications developed by Xerox in the late 1970. While Internet is a global network, Ethernet is a local area network (LAN).Ethernet is a standard communication protocol embedded in software and hardware devices. It is used for building a local area network. The local area network is a computer network that interconnects a group of computers and shares the information through cables or wires.

Wired Ethernet Network

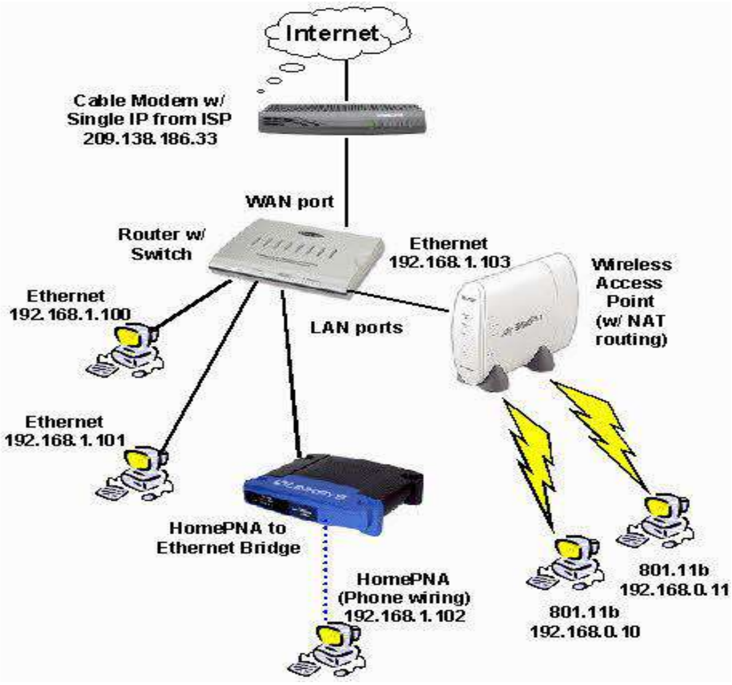
The Ethernet technology mainly works with the fiber optic cables that connect devices within a distance of 10 km. The Ethernet supports 10 Mbps.



Ethernet communication

A computer network interface card (NIC) is installed in each computer, and is assigned to a unique address. An Ethernet cable runs from each NIC to the central switch or hub. The switch and hub act as a relay though they have significant differences in the manner in which they handle network traffic – receiving and directing packets of data across the LAN. Thus, Ethernet networking creates a communications system that allows sharing of data and resources including printers, fax machines and scanners.

Wireless Ethernet



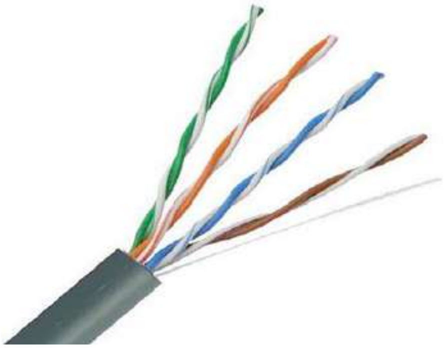
Wireless Network

Ethernet networks can also be wireless. Rather than using Ethernet cable to connect the computers, wireless NICs use radio waves for two-way communication with a wireless switch or hub. It consists of Ethernet ports, wireless NICs, switches and hubs. Wireless network technology can be more flexible to use, but also require extra care in configuring security.

Types of Ethernet Networks

There are several types of Ethernet networks, such as Fast Ethernet, Gigabit Ethernet, and Switch Ethernet. A network is a group of two or more computer systems connected together.

1. Fast Ethernet



Twisted pair cable

The fast Ethernet is a type of Ethernet network that can transfer data at a rate of 100 Mbps using a twisted-pair cable or a fiber-optic cable. The older 10 Mbps Ethernet is still used, but such networks do not provide necessary bandwidth for some network-based video applications.

Fast Ethernet is based on the proven CSMA/CD Media Access Control (MAC) protocol, and uses existing 10BaseT cabling. Data can move from 10 Mbps to 100 Mbps without any protocol translation or changes to the application and networking software.

What is Ethernet Port Speed?

When compare to a 10 mb port, a 100 Mb port is theoretically 10 times faster than the standard port. Therefore, with a 100 Mb port more information can stream to and from your server. This will be of great help to you if you really need to explore very high speed, but not if you are under DDOS attack because you will find yourself running out of traffic allocation very fast.

100Mbit/s Ethernet port

If you are doing standard web hosting, the bigger 100 Mbps pipe will not offer true benefit to you because you may not even use more than 1 mbps at any given time. If you are hosting games or streaming media, then the bigger pipe of 100 Mbps would indeed be helpful to you.

With a 10 mbps pipe, you can transfer up to 1.25 Mbps, while a 100 mbps pipe, would allow you to transfer up to 12.5 Mbps.

However, if you leave your server unattended and running at full steam, a 10 Mbps pipe can consume about 3,240 GB a month and a 100 Mbps pipe can consume up to 32,400 GB a month. It would be really disgusting when you receive your bill.

2. Gigabit Ethernet

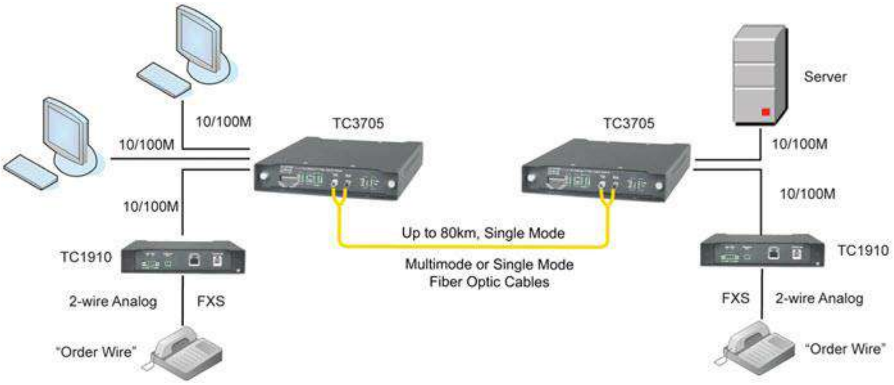
Optic fiber cable

The Gigabit Ethernet is a type of Ethernet network capable of transferring data at a rate of 1000 Mbps based on a twisted-pair or fiber optic cable, and it is very popular. The type of twisted-pair cables that support Gigabit Ethernet is Cat 5e cable, where all the four pairs of twisted wires of the cable are used to achieve high data transfer rates. The 10 Gigabit Ethernet is a latest generation Ethernet capable of transferring data at a rate of 10 Gbps using twisted-pair or fiber optic cable.

3. Switch Ethernet

Multiple network devices in a LAN require network equipments such as a network switch or hub. When using a network switch, a regular network cable is used instead of a crossover cable. The crossover cable consists of a transmission pair at one end and a receiving pair at the other end.

The main function of a network switch is to forward data from one device to another device on the same network. Thus a network switch performs this task efficiently as the data is transferred from one device to another without affecting other devices on the same network.



Switch Ethernet

The network switch normally supports different data transfer rates. The most common data transfer rates include 10 Mbps – 100 Mbps for fast Ethernet, and 1000 Mbps – 10 Gbps for the latest Ethernet.

Switch Ethernet uses star topology, which is organized around a switch. The switch in a network uses a filtering and switching mechanism similar to the one used by the gateways, in which these techniques have been in use for a long time.

Alternate Technologies of Ethernet

The Ethernet supports different types of networks or topologies such a bus topology, ring topology, star topology, tree topology, and so on. These topologies can be used for transferring and receiving data using different types of cables like coax, twisted pair, fiber optic, etc.

Alternate technologies of Ethernet include the “Token Ring” protocol designed by IBM, and the robust Asynchronous Transfer Mode (ATM) technology. ATM allows devices to be connected over very long distances to create WANs (Wide Area Networks) that behave like LANs. However, for an inexpensive network located in a single building, Ethernet is a well-established standard with a solid record, boasting over three decades of providing reliable networking environments.

The formal designation for standardization of Ethernet protocol is referred to as IEEE 802.3. A third subcommittee works on a flavor essentially identical to Ethernet, though there are insignificant variances. Consequently, generic use of the term “Ethernet” might refer to IEEE 802.3 or DIX Ethernet.

Different Types of Ethernet Cables

Different variants of Ethernet technologies are designated according to the type and diameter of the cables used as given below:

- 10Base2: The cable used is a thin coaxial cable: thin Ethernet.
- 10Base5: The cable used is a thick coaxial cable: thick Ethernet.
- 10Base-T: The cable used is a twisted-pair (T means twisted pair) and the speed achieved is around 10 Mbps.
- 100Base-FX: Makes it possible to achieve a speed of 100 Mbps by using multimode fiber optic (F stands for Fiber).
- 100Base-TX: Similar to 10Base-T, but with a speed 10 times greater (100 Mbps).
- 1000Base-T: Uses a double-twisted pair of category 5 cables and allows a speed up to one Gigabit per second.

- 1000Base-SX: Based on multimode fiber optic uses a short wavelength signal (S stands for short) of 850 nanometers (770 to 860 nm).
- 1000Base-LX: Based on multimode fiber optic uses a long wavelength signal (L stands for long) of 1350 nm (1270 to 1355 nm). Ethernet is a widely used network technology because the cost of such a network is not very high.

Top Features of Ethernet controller

1. Includes 1st round “hop” to a Tier 1 provider
2. Provides wholesale pricing for all types of businesses
3. Connects directly to the carrier’s backbone
4. Offers Service Level Agreements with every connection
5. Provides low-cost bandwidth
6. Provides higher rates of data transfer
7. Offers ‘Plug and Play’ provisioning

Wireless LAN

Introduction

Wireless local area networks (WLANs) are the same as the traditional LAN but they have a wireless interface. With the introduction of small portable devices such as PDAs (personal digital assistants), the WLAN technology is becoming very popular. WLANs provide high speed data communication in small areas such as a building or an office. It allows users to move around in a confined area while they are still connected to the network. Examples of wireless LAN that are available today are NCR's waveLAN and Motorola's ALTAIR. In this article, the transmission technology used in WLANs is considered. We will also discuss some of the technical standards for WLANs developed by the IEEE Project 802.11.

Figure 1 : The Motorola Envoy (PDA) [2]

Transmission Technology

There are three main ways by which WLANs transmit information : microwave, spread spectrum and infrared.

Microwave Transmission

Motorola's WLAN product (ALTAIR) transmits data by using low powered microwave radio signals. It operates at the 18GHz frequency band.

With this transmission technology, there are two methods used by wireless LAN products : frequency hopping and direct sequence modulation.

- **Frequency hopping**
  The signal jumps from one frequency to another within a given frequency range. The transmitter device "listens" to a channel, if it detects an idle time (i.e. no signal is transmitted), it transmits the data using the full channel bandwidth. If the channel is full, it "hops" to another channel and repeats the process. The transmitter and the receiver "jump" in the same manner.
- **Direct SequenceModulation**
  This method uses a wide frequency band together with Code Division Multiple Access (CDMA). Signals from different units are transmitted at a given frequency range. The power levels of these signals are very low (just above background noise). A code is transmitted with each signal so that the receiver can identify the appropriate signal transmitted by the sender unit.
  The frequency at which such signals are transmitted is called the ISM (industrial, scientific and medical) band. This frequency band is reserved for ISM devices. The ISM band has three frequency ranges : 902-928, 2400-2483.5 and 5725-5850 MHz. An exception to this is Motorola's ALTAIR which operates at 18GHz.
  Spread spectrum transmission technology is used by many wireless LAN manufacturers such as NCR for waveLAN product and SpectraLink for the 2000 PCS.

Infrared Transmission

This method uses infrared light to carry information. There are three types of infrared transmission : diffused, directed and directed point-to-point.

- **Diffused**
  The infrared light transmitted by the sender unit fills the area (e.g. office). Therefore the receiver unit located anywhere in that area can receive the signal.
- **Directed**
  The infrared light is focused before transmitting the signal. This method increases the transmission speed.
- **Directed point-to-point**
  Directed point-to-point infrared transmission provides the highest transmission speed. Here the receiver is aligned with the sender unit. The infrared light is then transmitted directly to the receiver.

The light source used in infrared transmission depends on the environmemt. Light emitting diode (LED) is used in indoor areas, while lasers are used in outdoor areas.

Infrared radiation (IR) has major biological effects. It greatly affects the eyes and skin. Microwave signals are also dangerous to health. But with proper design of systems, these effects are reduced considerably.

Technical Standards

Technical standards are one of the main concerns of users of wireless LAN products. Users would like to be able to buy wireless products from different manufacturers and be able to use them on one network. The IEEE Project 802.11 has set up universal standards for wireless LAN. In this section we will consider some of these standards.

Requirements

In March 1992 the IEEE Project 802.11 established a set of requirements for wireless LAN. The minimum bandwidth needed for operations such as file transfer and program loading is 1Mbps. Operations which need real-time data transmission such as digital voice and process control, need support from time bounded services.

Types of Wireless LAN

The Project 802.11 committee distinguished between two types of wireless LAN : "ad-hoc" and"infrastructred" networks.

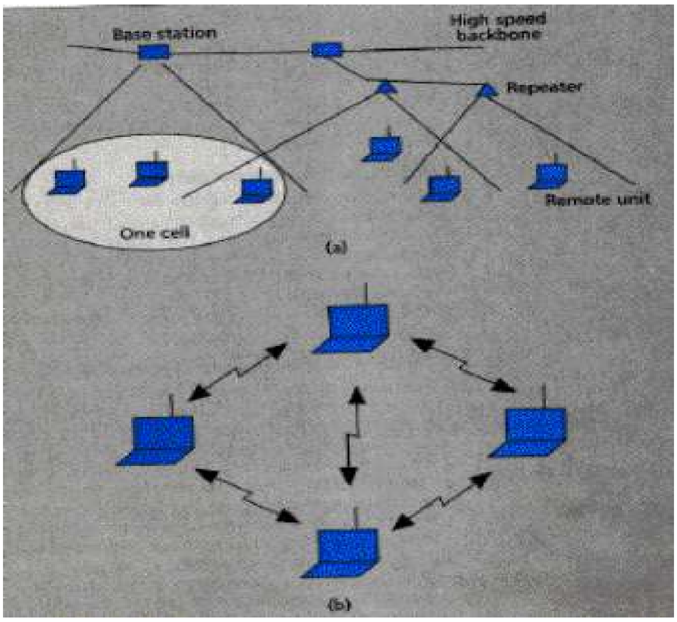


Figure 2 : (a) Infrastructred Wireless LAN; (b) Ad-hoc Wireless LAN. [3]

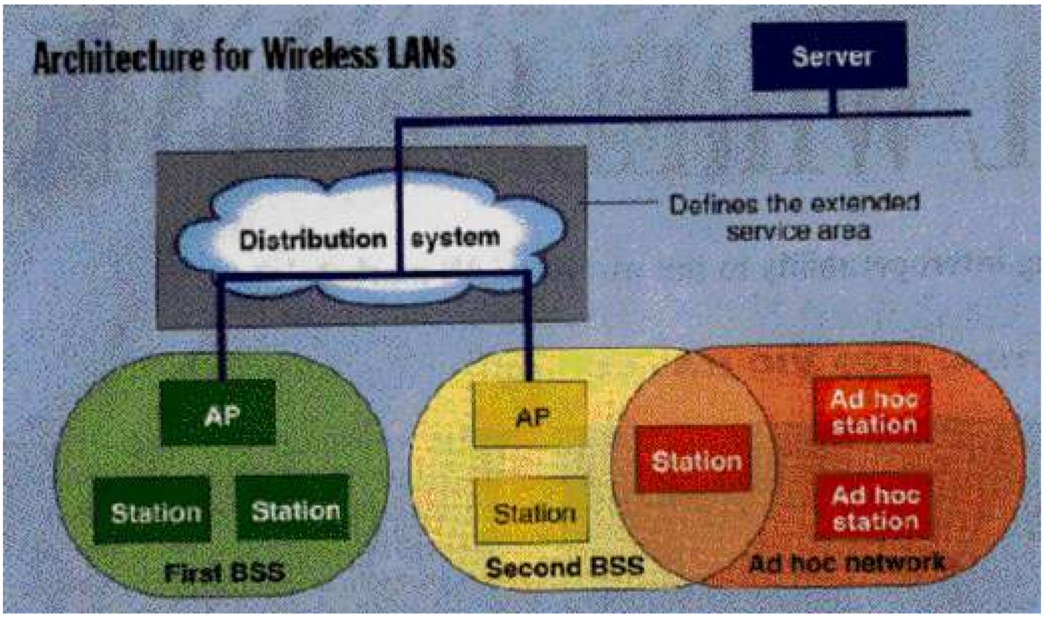
Ad-hoc Networks

Figure 2b shows an ad-hoc network. This network can be set up by a number mobile users meeting in a small room. It does not need any support from a wired/wireless backbone. There are two ways to implement this network.

- **Broadcasting/Flooding**
  Suppose that a mobile user A wants to send data to another user B in the same area. When the packets containing the data are ready, user A broadcasts the packets. On receiving the packets, the receiver checks the identification on the packet. If that receiver was not the correct destination, then it rebroadcasts the packets. This process is repeated until user B gets the data.
- **Temporary Infrastructure**
  In this method, the mobile users set up a temporary infrastructure. But this method is complicated and it introduces overheads. It is useful only when there is a small number of mobile users.

Infrastructure Networks

Figure 2a shows an infrastructure-based network. This type of network allows users to move in a building while they are connected to computer resources. The IEEE Project 802.11 specified the components in a wireless LAN architecture. In an infrastructure network, a cell is also known as a Basic Service Area (BSA). It contains a number of wireless stations. The size of a BSA depends on the power of the transmitter and receiver units, it also depends on the environment. A number of BSAs are connected to each other and to a distribution system by Access Points (APs). A group of stations belonging to an AP is called a Basic Service Set (BSS). Figure 3 shows the basic architecture for wireless LANs.



Bluetooth

Bluetooth wireless technology is a short range communications technology intended to replace the cables connecting portable unit and maintaining high levels of security. Bluetooth technology is based on **Ad-hoc technology** also known as **Ad-hoc Pico nets**, which is a local area network with a very limited coverage.

History of Bluetooth

WLAN technology enables device connectivity to infrastructure based services through a wireless carrier provider. The need for personal devices to communicate wirelessly with one another without an established infrastructure has led to the emergence of **Personal Area Networks (PANs)**.

- Ericsson's Bluetooth project in 1994 defines the standard for PANs to enable communication between mobile phones using low power and low cost radio interfaces.
- In May 1988, Companies such as IBM, Intel, Nokia and Toshiba joined Ericsson to form the Bluetooth Special Interest Group (SIG) whose aim was to develop a defacto standard for PANs.
- IEEE has approved a Bluetooth based standard named IEEE 802.15.1 for Wireless Personal Area Networks (WPANs). IEEE standard covers MAC and Physical layer applications.

Bluetooth specification details the entire protocol stack. Bluetooth employs Radio Frequency (RF) for communication. It makes use of **frequency modulation** to generate radio waves in the **ISM** band.

Symbol of Bluetooth

An example of a Bluetooth device

The usage of Bluetooth has widely increased for its special features.

- Bluetooth offers a uniform structure for a wide range of devices to connect and communicate with each other.
- Bluetooth technology has achieved global acceptance such that any Bluetooth enabled device, almost everywhere in the world, can be connected with Bluetooth enabled devices.
- Low power consumption of Bluetooth technology and an offered range of up to ten meters has paved the way for several usage models.
- Bluetooth offers interactive conference by establishing an adhoc network of laptops.

- Bluetooth usage model includes cordless computer, intercom, cordless phone and mobile phones.

Piconets and Scatternets

Bluetooth enabled electronic devices connect and communicate wirelessly through shortrange devices known as **Piconets**. Bluetooth devices exist in small ad-hoc configurations with the ability to act either as master or slave the specification allows a mechanism for **master** and **slave** to switch their roles. Point to point configuration with one master and one slave is the simplest configuration.

When more than two Bluetooth devices communicate with one another, this is called a **PICONET**. A Piconet can contain up to seven slaves clustered around a single master. The device that initializes establishment of the Piconet becomes the **master**.

The master is responsible for transmission control by dividing the network into a series of time slots amongst the network members, as a part of **time division multiplexing** scheme which is shown below.

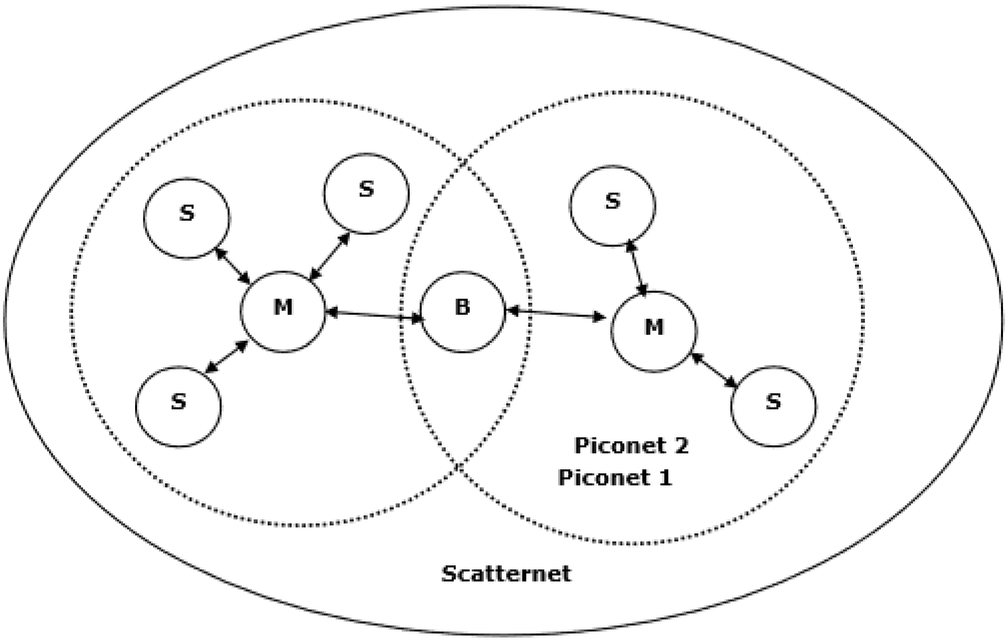


Figure: Piconets and Scatternets

The features of Piconets are as follows −

- Within a Piconet, the timing of various devices and the frequency hopping sequence of individual devices is determined by the clock and unique **48-bit address** of master.
- Each device can communicate simultaneously with up to seven other devices within a single Piconet.
- Each device can communicate with several piconets simultaneously.

- Piconets are established dynamically and automatically as Bluetooth enabled devices enter and leave piconets.
- There is no direct connection between the slaves and all the connections are essentially master-to-slave or slave-to-master.
- Slaves are allowed to transmit once these have been polled by the master.
- Transmission starts in the slave-to-master time slot immediately following a polling packet from the master.
- A device can be a member of two or more piconets, jumping from one piconet to another by adjusting the transmission regime-timing and frequency hopping sequence dictated by the master device of the second piconet.
- It can be a slave in one piconet and master in another. It however cannot be a master in more than once piconet.
- Devices resident in adjacent piconets provide a bridge to support inner-piconet connections, allowing assemblies of linked piconets to form a physically extensible communication infrastructure known as **Scatternet**.

Spectrum

Bluetooth technology operates in the unlicensed industrial, scientific and medical (ISM) band at 2.4 to 2.485 GHZ, using a spread spectrum hopping, full-duplex signal at a nominal rate of 1600 hops/sec. the 2.4 GHZ ISM band is available and unlicensed in most countries.

Range

Bluetooth operating range depends on the device Class 3 radios have a range of up to 1 meter or 3 feet Class 2 radios are most commonly found in mobile devices have a range of 10 meters or 30 feet Class 1 radios are used primarily in industrial use cases have a range of 100 meters or 300 feet.

Data rate

Bluetooth supports 1Mbps data rate for version 1.2 and 3Mbps data rate for Version 2.0 combined with Error Data Rate.

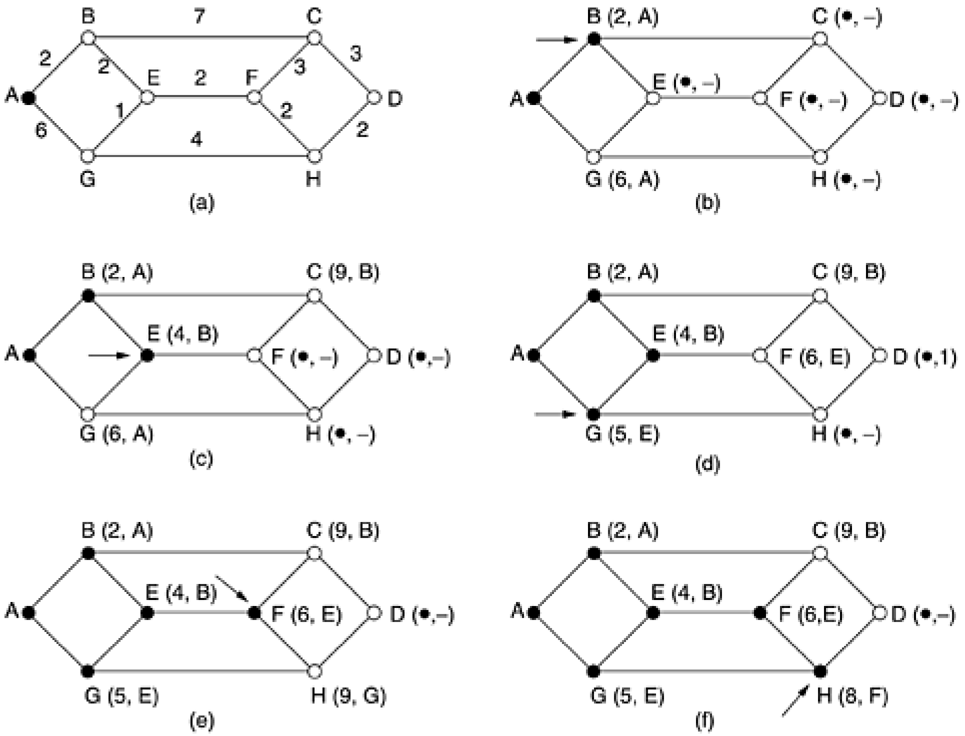
Network Layer

Routing Algorithms

Shortest path routing

The concept of a **shortest path** deserves some explanation. One way of measuring path length is the number of hops. Using this metric, the paths *ABC* and *ABE* in Fig. 5-7 are equally long. Another metric is the geographic distance in kilometers, in which case *ABC* is clearly much longer than *ABE* (assuming the figure is drawn to scale).

Figure 5-7. The first five steps used in computing the shortest path from* A *to* D*. The arrows indicate the working node.



However, many other metrics besides hops and physical distance are also possible. For example, each arc could be labeled with the mean queueing and transmission delay for some standard test packet as determined by hourly test runs. With this graph labeling, the shortest path is the fastest path rather than the path with the fewest arcs or kilometers.

In the general case, the labels on the arcs could be computed as a function of the distance, bandwidth, average traffic, communication cost, mean queue length, measured delay, and other factors. By changing the weighting function, the algorithm would then compute the "shortest" path measured according to any one of a number of criteria or to a combination of criteria.

Several algorithms for computing the shortest path between two nodes of a graph are known. This one is due to Dijkstra (1959). Each node is labeled (in parentheses) with its distance from the source node along the best known path. Initially, no paths are known, so all nodes are labeled with infinity. As the algorithm proceeds and paths are found, the labels may change, reflecting better paths. A label may be either tentative or permanent. Initially, all labels are tentative. When it is discovered that a label represents the shortest possible path from the source to that node, it is made permanent and never changed thereafter.

To illustrate how the labeling algorithm works, look at the weighted, undirected graph of Fig. 5-7(a), where the weights represent, for example, distance. We want to find the shortest path from *A* to *D*. We start out by marking node *A* as permanent, indicated by a filled-in circle. Then we examine, in turn, each of the nodes adjacent to *A* (the working node), relabeling each one with the distance to *A*. Whenever a node is relabeled, we also label it with the node from which the probe was made so that we can reconstruct the final path later. Having examined each of the nodes adjacent to *A*, we examine all the tentatively labeled nodes in the whole graph and make the one with the smallest label permanent, as shown in Fig. 5-7(b). This one becomes the new working node.

We now start at *B* and examine all nodes adjacent to it. If the sum of the label on *B* and the distance from *B* to the node being considered is less than the label on that node, we have a shorter path, so the node is relabeled.

After all the nodes adjacent to the working node have been inspected and the tentative labels changed if possible, the entire graph is searched for the tentatively-labeled node with the smallest value. This node is made permanent and becomes the working node for the next round. Figure 5-7 shows the first five steps of the algorithm.

To see why the algorithm works, look at Fig. 5-7(c). At that point we have just made *E* permanent. Suppose that there were a shorter path than *ABE*, say *AXYZE*. There are two possibilities: either node *Z* has already been made permanent, or it has not been. If it has, then *E* has already been probed (on the round following the one when *Z* was made permanent), so the *AXYZE* path has not escaped our attention and thus cannot be a shorter path.

Now consider the case where *Z* is still tentatively labeled. Either the label at *Z* is greater than or equal to that at *E*, in which case *AXYZE* cannot be a shorter path than *ABE*, or it is less than that of *E*, in which case *Z* and not *E* will become permanent first, allowing *E* to be probed from *Z*.

Distance Vector routing

Modern computer networks generally use dynamic routing algorithms rather than the static ones described above because static algorithms do not take the current network load into account. Two dynamic algorithms in particular, distance vector routing and link state routing, are the most popular. In this section we will look at the former algorithm. In the following section we will study the latter algorithm.

Distance vector routing algorithms operate by having each router maintain a table (i.e, a vector) giving the best known distance to each destination and which line to use to get there. These tables are updated by exchanging information with the neighbors.

The distance vector routing algorithm is sometimes called by other names, most commonly the distributed Bellman-Ford routing algorithm and the Ford-Fulkerson algorithm, after the researchers who developed it (Bellman, 1957; and Ford and Fulkerson, 1962). It was the original ARPANET routing algorithm and was also used in the Internet under the name RIP.

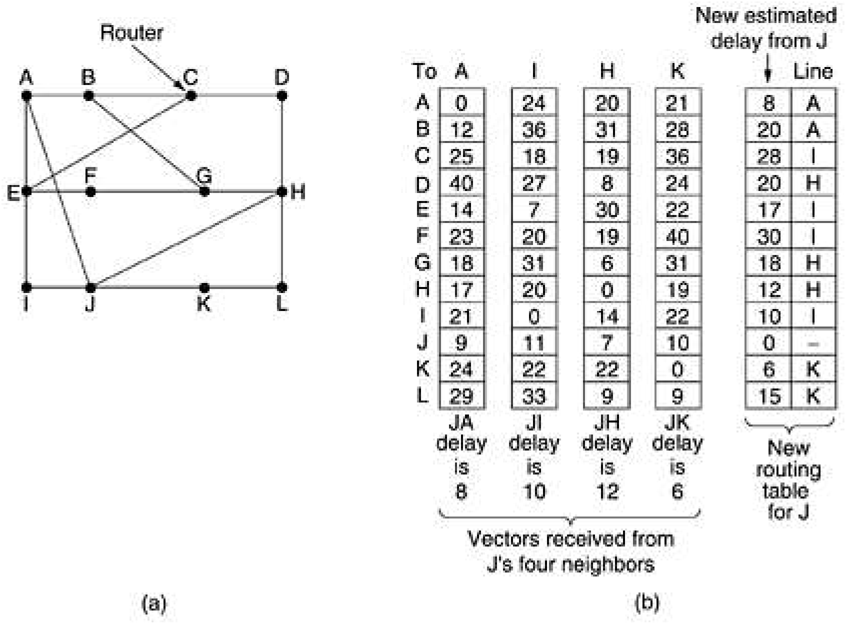
In distance vector routing, each router maintains a routing table indexed by, and containing one entry for, each router in the subnet. This entry contains two parts: the preferred outgoing line to use for that destination and an estimate of the time or distance to that destination. The metric used might be number of hops, time delay in milliseconds, total number of packets queued along the path, or something similar.

The router is assumed to know the "distance" to each of its neighbors. If the metric is hops, the distance is just one hop. If the metric is queue length, the router simply examines each queue. If the metric is delay, the router can measure it directly with special ECHO packets that the receiver just timestamps and sends back as fast as it can.

As an example, assume that delay is used as a metric and that the router knows the delay to each of its neighbors. Once every T msec each router sends to each neighbor a list of its estimated delays to each destination. It also receives a similar list from each neighbor. Imagine that one of these tables has just come in from neighbor X, with Xi being X's estimate of how long it takes to get to router i. If the router knows that the delay to X is m msec, it also knows that it can reach router i via X in Xi + m msec. By performing this calculation for each neighbor, a router can find out which estimate seems the best and use that estimate and the corresponding line in its new routing table. Note that the old routing table is not used in the calculation.

This updating process is illustrated in Fig. 5-9. Part (a) shows a subnet. The first four columns of part (b) show the delay vectors received from the neighbors of router J. A claims to have a 12-msec delay to B, a 25-msec delay to C, a 40-msec delay to D, etc. Suppose that J has measured or estimated its delay to its neighbors, A, I, H, and K as 8, 10, 12, and 6 msec, respectively.

Figure 5-9. (a) A subnet. (b) Input from A, I, H, K, and the new routing table for J.



To	A	I	H	K	New estimated delay from J	Line
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	–
K	24	22	22	0	6	K
L	29	33	9	9	15	K
	JA delay is 8	JI delay is 10	JH delay is 12	JK delay is 6	New routing table for J	

Vectors received from J's four neighbors

Consider how J computes its new route to router G. It knows that it can get to A in 8 msec, and A claims to be able to get to G in 18 msec, so J knows it can count on a delay of 26 msec to G if it forwards packets bound for G to A. Similarly, it computes the delay to G via I, H, and K as 41 (31 + 10), 18 (6 + 12), and 37 (31 + 6) msec, respectively. The best of these values is 18, so it makes an entry in its routing table that the delay to G is 18 msec and that the route to use is via H. The same calculation is performed for all the other destinations, with the new routing table shown in the last column of the figure.

Link state routing

Distance vector routing was used in the ARPANET until 1979, when it was replaced by link state routing. Two primary problems caused its demise. First, since the delay metric was queue length, it did not take line bandwidth into account when choosing routes. Initially, all the lines were 56 kbps, so line bandwidth was not an issue, but after some lines had been upgraded to 230 kbps and others to 1.544 Mbps, not taking bandwidth into account was a major problem. Of course, it would have been possible to change the delay metric to factor in line bandwidth, but a second problem also existed, namely, the algorithm often took too long to converge (the count-to-infinity problem). For these reasons, it was replaced by an entirely new algorithm, now called **link state routing**. Variants of link state routing are now widely used.

The idea behind link state routing is simple and can be stated as five parts. Each router must do the following:

1. Discover its neighbors and learn their network addresses.
2. Measure the delay or cost to each of its neighbors.
3. Construct a packet telling all it has just learned.
4. Send this packet to all other routers.
5. Compute the shortest path to every other router.

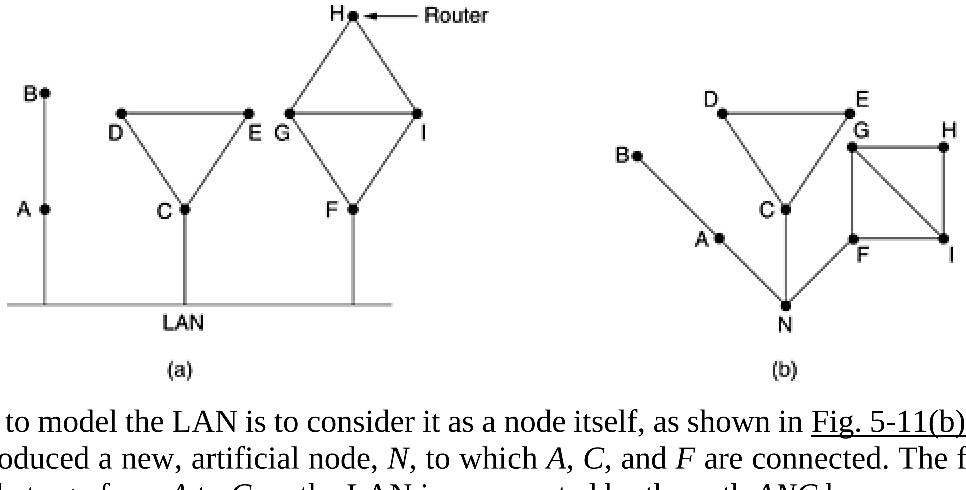
In effect, the complete topology and all delays are experimentally measured and distributed to every router. Then Dijkstra's algorithm can be run to find the shortest path to every other router. Below we will consider each of these five steps in more detail.

Learning about the Neighbors

When a router is booted, its first task is to learn who its neighbors are. It accomplishes this goal by sending a special HELLO packet on each point-to-point line. The router on the other end is expected to send back a reply telling who it is. These names must be globally unique because when a distant router later hears that three routers are all connected to *F*, it is essential that it can determine whether all three mean the same *F*.

When two or more routers are connected by a LAN, the situation is slightly more complicated. Fig. 5-11(a) illustrates a LAN to which three routers, *A*, *C*, and *F*, are directly connected. Each of these routers is connected to one or more additional routers, as shown.

Figure 5-11. (a) Nine routers and a LAN. (b) A graph model of (a).



One way to model the LAN is to consider it as a node itself, as shown in Fig. 5-11(b). Here we have introduced a new, artificial node, *N*, to which *A*, *C*, and *F* are connected. The fact that it is possible to go from *A* to *C* on the LAN is represented by the path *ANC* here.

Measuring Line Cost

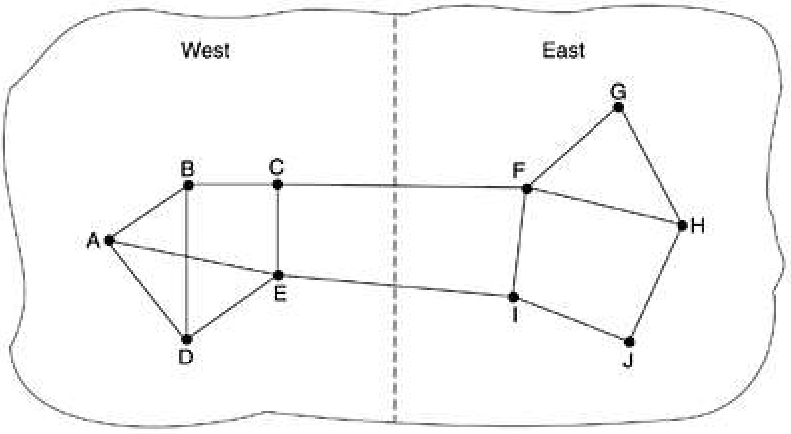
The link state routing algorithm requires each router to know, or at least have a reasonable estimate of, the delay to each of its neighbors. The most direct way to determine this delay is to send over the line a special ECHO packet that the other side is required to send back immediately. By measuring the round-trip time and dividing it by two, the sending router can get a reasonable estimate of the delay. For even better results, the test can be conducted several times, and the average used. Of course, this method implicitly assumes the delays are symmetric, which may not always be the case.

An interesting issue is whether to take the load into account when measuring the delay. To factor the load in, the round-trip timer must be started when the ECHO packet is queued. To ignore the load, the timer should be started when the ECHO packet reaches the front of the queue.

Arguments can be made both ways. Including traffic-induced delays in the measurements means that when a router has a choice between two lines with the same bandwidth, one of which is heavily loaded all the time and one of which is not, the router will regard the route over the unloaded line as a shorter path. This choice will result in better performance.

Unfortunately, there is also an argument against including the load in the delay calculation. Consider the subnet of Fig. 5-12, which is divided into two parts, East and West, connected by two lines, *CF* and *EI*.

Figure 5-12. A subnet in which the East and West parts are connected by two lines.

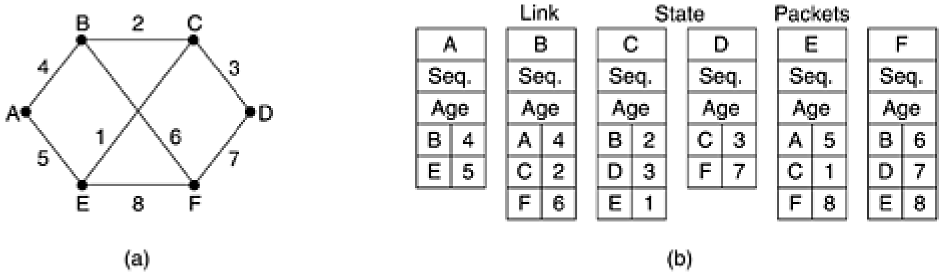


Suppose that most of the traffic between East and West is using line *CF*, and as a result, this line is heavily loaded with long delays. Including queueing delay in the shortest path calculation will make *EI* more attractive. After the new routing tables have been installed, most of the East-West traffic will now go over *EI*, overloading this line. Consequently, in the next update, *CF* will appear to be the shortest path. As a result, the routing tables may oscillate wildly, leading to erratic routing and many potential problems. If load is ignored and only bandwidth is considered, this problem does not occur. Alternatively, the load can be spread over both lines, but this solution does not fully utilize the best path. Nevertheless, to avoid oscillations in the choice of best path, it may be wise to distribute the load over multiple lines, with some known fraction going over each line.

Building Link State Packets

Once the information needed for the exchange has been collected, the next step is for each router to build a packet containing all the data. The packet starts with the identity of the sender, followed by a sequence number and age (to be described later), and a list of neighbors. For each neighbor, the delay to that neighbor is given. An example subnet is given in Fig. 5-13(a) with delays shown as labels on the lines. The corresponding link state packets for all six routers are shown in Fig. 5-13(b).

Figure 5-13. (a) A subnet. (b) The link state packets for this subnet.



Link State Packets

A	B	C	D	E	F
Seq.	Seq.	Seq.	Seq.	Seq.	Seq.
Age	Age	Age	Age	Age	Age
B 4	A 4	B 2	C 3	A 5	B 6
E 5	C 2	D 3	F 7	C 1	D 7
	F 6	E 1		F 8	E 8

Building the link state packets is easy. The hard part is determining when to build them. One possibility is to build them periodically, that is, at regular intervals. Another possibility is to build them when some significant event occurs, such as a line or neighbor going down or coming back up again or changing its properties appreciably.

Distributing the Link State Packets

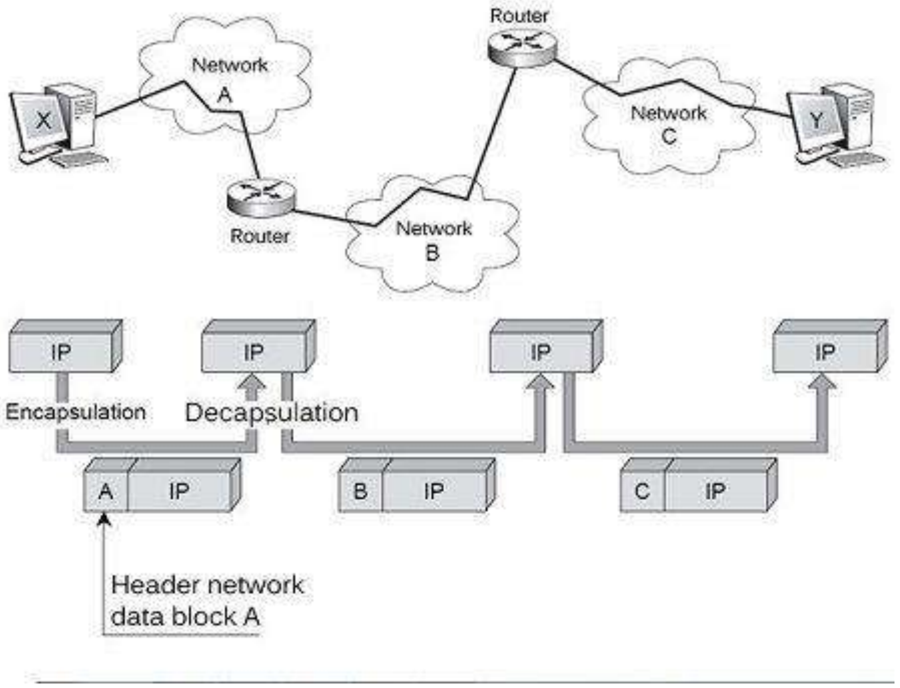
The trickiest part of the algorithm is distributing the link state packets reliably. As the packets are distributed and installed, the routers getting the first ones will change their routes. Consequently, the different routers may be using different versions of the topology, which can lead to inconsistencies, loops, unreachable machines, and other problems.

IP

The Internet's basic protocol called IP for Internet Protocol. The objective of starting this protocol is assigned to interconnect networks do not have the same frame-level protocols or package level. The internet acronym comes from inter-networking and corresponds to an interconnection fashion: each independent network must transport in the weft or in the data area of the packet an IP packet, as shown in Figure.

There are two generations of IP packets, called IPv4 (IP version 4) and IPv6 (IP version 6). IPv4 has been dominant so far. The transition to IPv6 could accelerate due to its adoption in many Asian countries. The transition is however difficult and will last many years.

- Internet Protocol (IP) of network layer contains addressing information and some control information that enables the packets to be routed.



• IP has two primary responsibilities:

1. Providing connectionless, best effort delivery of datagrams through a internetwork. The term best effort delivery means that IP does not provides any error control or flow control. The term connectionless means that each datagram is handled independently, and each datagram can follow different route to the destination. This implies that datagrams sent by the same source to the same destination could arrive out of order.

2. Providing fragmentation and reassembly of datagrams to support data links with different maximum transmission unit (MTU) sizes.

IP packet format

• Packets in the network layer are called datagrams.

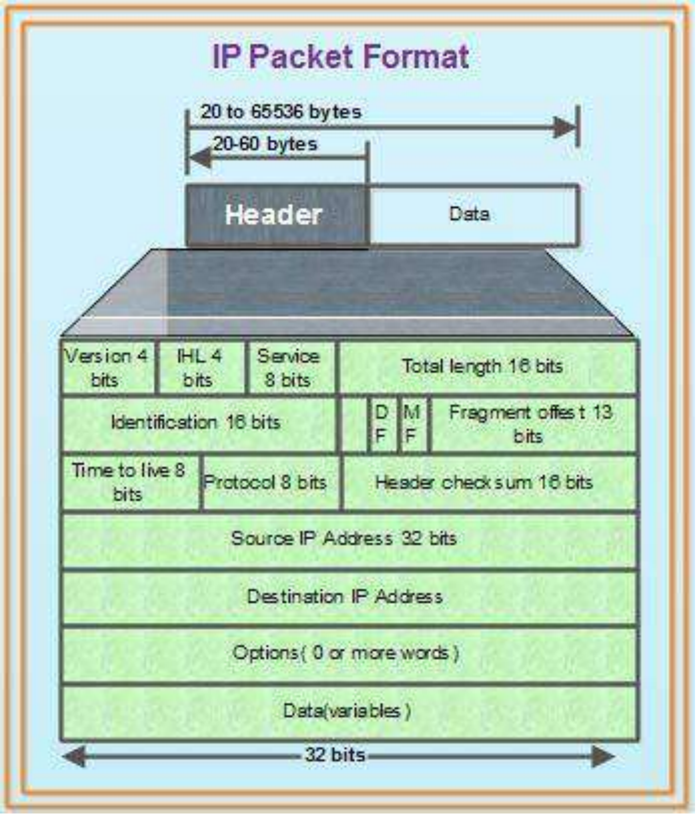
A datagram is a variable length packet consisting of two parts: header and data.

• The header is 20 to 60 bytes in length and contains information essential to routing and delivery.

• The various fields in IP header are:

1. **Version**: It is a 4-bit field that specifies the version of IP currently being used. Two different versions of protocols are IPV4 (Internet Protocol Version 4) and IPV6 (Internet Protocol Version 6).

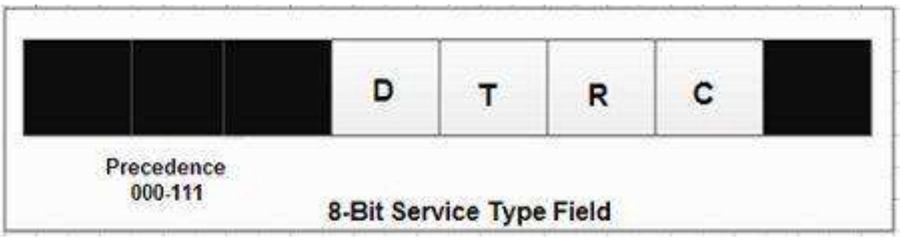
2. **IP Header Length (IHL):** This 4-bit field indicates the datagram header length in 32 bit word. The header length i8 not constant in IP. It may vary from 20 to 60 bytes. When there are no options, the header length is 20 bytes, and the value of this field is 5. When the option field is at its maximum size, the value of this field is 15.



3. **Services**: This 8 hit field was previously called services type but is now called differentiated services.

The various bits in service type are:

• A 3-bit precedence field that defines the priority of datagram in issues such as congestion. This 3-bit subfield ranges from 0 (000 in binary) to 7 (111 in binary).



• After 3-bit precedence there are four flag bits. These bits can be either 0 or 1 and only one of the bits can have value of 1 in each datagram.

The various flag bits are:

D : Minimize delay

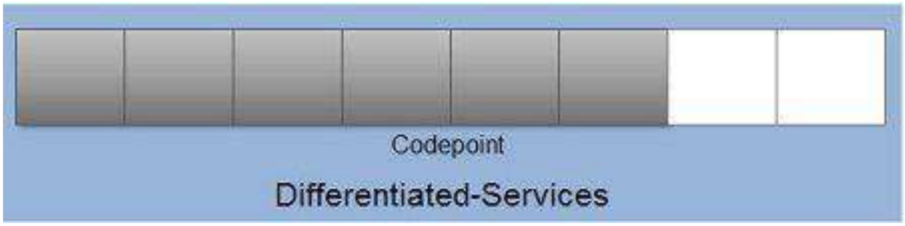
T : Maximize throughout

R : Maximize reliability

C : Minimize Cost

The various bits in differentiated services are:

• The first 6 bits defined a *codepoint* and last two bits are not used. If the 3 rightmost bits are 0s, the 3 leftmost bits are interpreted the same as the precedence bits in the service type interpretation.



4. **Total length**: This 16 bit field specifies the total length of entire IP datagram including data and header in bytes. As there are 16 bits, the total length of IP datagram is limited to 65,535 (2^{16} - 1) bytes.

5. **Identification**: This 16 bit field is used in fragmentation. A datagram when passing through different networks may be divided into fragments to match the network frame size. Therefore, this field contains an integer that identifies the current datagram. This field is used to help piece together datagram fragments.

6. **Flags**: Consists' of a 3 bit field of which the two low order bit DF, MF control fragmentation. DF stands for Don't Fragment. DF specifies whether the packet can be fragmented MF stands for more fragments. MF specifies whether the packet is the last fragment in a series of fragmented packets. The third or high order but is not used.

7. **Fragment Offset**: This 13 bit field indicates the position of the fragment's data relative to the beginning of the data in the original datagram, which allows the destination IP process to properly reconstruct the original datagram.

8. **Time to Live**: It is 8 bit field that maintain a counter that gradually decrements down to zero, at which point the datagram is discarded. This keeps the packet from looping endlessly.

9. **Protocol**: This 8 bit field indicates which upper layer protocol receives incoming packets after IP processing is complete.

10. **Header Checksum**: This 16 bit field contains a checksum that covers only the header and not the data.

11. **Source IP address**: These 32-bit field contains the IP address of source machine.

12. **Destination IP address**: This 32-bit field contains the IP address of destination machine.

13. **Options**: This field allows IP to support various options such as security, routing, timing management and alignment.

14. **Data**: It contains upper layer information.

IP Address

IP address is a short form of "Internet Protocol Address." It is a unique number provided to every device connected to the internet network, such as Android phone, laptop, Mac, etc. An IP address is represented in an integer number separated by a dot (.), for example, 192.167.12.46.

Types of IP Address

An IP address is categorized into two different types based on the number of IP address it contains. These are:

- IPv4 (Internet Protocol version 4)
- IPv6 (Internet Protocol version 6)

What is IPv4?

IPv4 is version 4 of IP. It is a current version and the most commonly used IP address. It is a 32-bit address written in four numbers separated by a dot (.), i.e., periods. This address is unique for each device. For example, 66.94.29.13

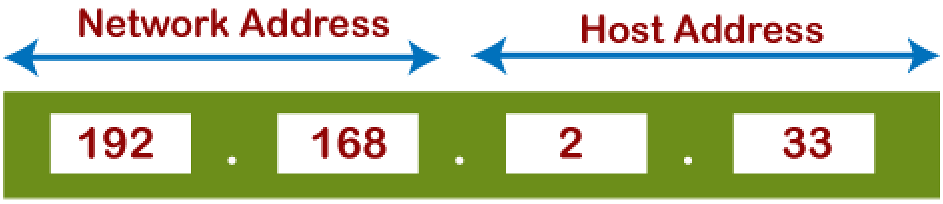
What is IPv6?

IPv4 produces 4 billion addresses, and the developers think that these addresses are enough, but they were wrong. IPv6 is the next generation of IP addresses. The main difference between IPv4 and IPv6 is the address size of IP addresses. The IPv4 is a 32-bit address, whereas IPv6 is a 128-bit hexadecimal address. IPv6 provides a large address space, and it contains a simple header as compared to IPv4.

To know more about the difference between IPv4 and IPv6, look at our article ipv4 vs. ipv6.

IP Address Format

Originally IP addresses were divided into five different categories called **classes**. These divided IP classes are class A, class B, class C, class D, and class E. Out of these, classes A, B, and C are most important. Each address class defines a different number of bits for its **network prefix (network address)** and **host number (host address)**. The starting address bits decide from which class an address belongs.

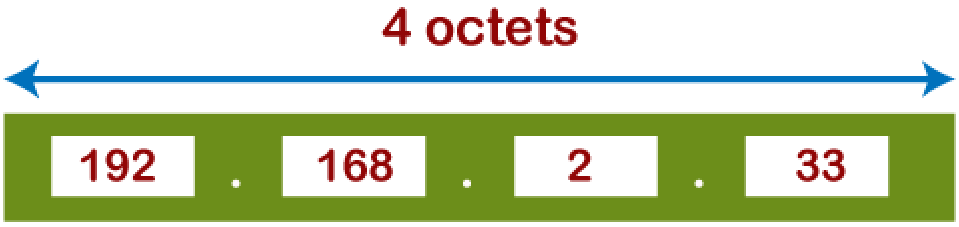


Network Address: The network address specifies the unique number which is assigned to your network. In the above figure, the network address takes two bytes of IP address.

Host Address: A host address is a specific address number assigned to each host machine. With the help of the host address, each machine is identified in your network. The network address will be the same for each host in a network, but they must vary in host address.

Address Format IPv4

The address format of IPv4 is represented into **4-octets** (32-bit), which is divided into three different classes, namely class A, class B, and class C.



The above diagram shows the address format of IPv4. An IPv4 is a 32-bit decimal address. It contains four octets or fields separated by 'dot,' and each field is 8-bit in size. The number that each field contains should be in the range of 0-255.

Class A

Class A address uses only first higher order octet (byte) to identify the network prefix, and remaining three octets (bytes) are used to define the individual host addresses. The class A address ranges between 0.0.0.0 to 127.255.255.255. The first bit of the first octet is always set to 0 (zero), and next 7 bits determine network address, and the remaining 24 bits determine host address. So the first octet ranges from 0 to 127 (00000000 to 01111111).

Class B

Class B addresses use the initial two octets (two bytes) to identify the network prefix, and the remaining two octets (two bytes) define host addresses. The class B addresses are range between 128.0.0.0 to 191.255.255.255. The first two bits of the first higher octet is always set to 10 (one and zero bit), and next 14 bits determines the network address and remaining 16 bits determines the host address. So the first octet ranges from 128 to 191 (10000000 to 10111111).

Class C

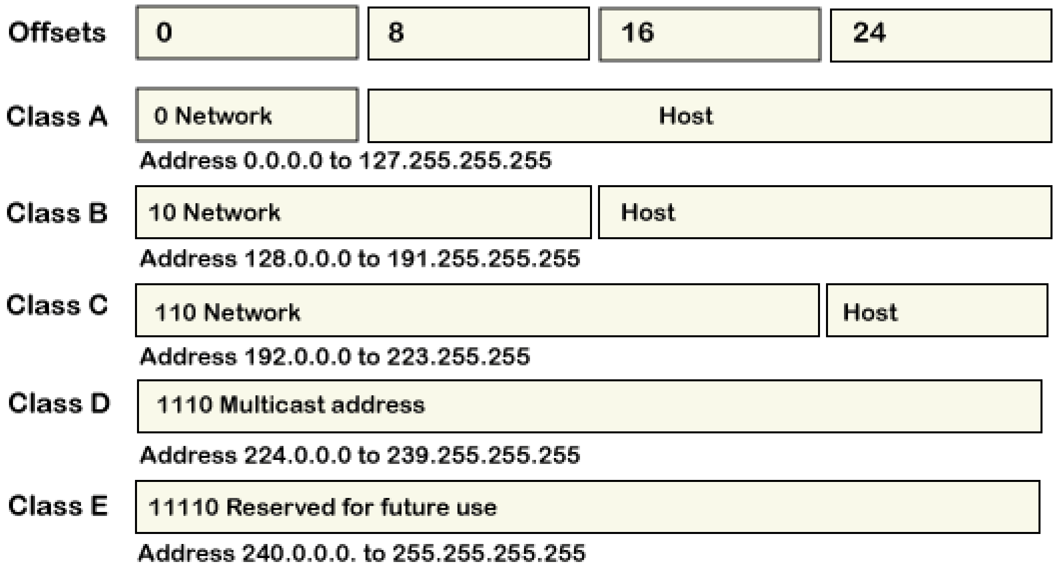
Class C addresses use the first three octets (three bytes) to identify the network prefix, and the remaining last octet (one byte) defines the host address. The class C address ranges between 192.0.0.0 to 223.255.255.255. The first three bit of the first octet is always set to 110, and next 21 bits specify network address and remaining 8 bits specify the host address. Its first octet ranges from 192 to 223 (11000000 to 11011111).

Class D

Class D IP address is reserved for multicast addresses. Its first four bits of the first octet are always set to 1110, and the remaining bits determine the host address in any IP address. The first higher octet bits are always set to 1110, and the remaining bits specify the host address. The class D address ranges between 224.0.0.0 to 239.255.255.255. In multicasting, data is not assigned to any particular host machine, so it is not require to find the host address from the IP address, and also, there is no subnet mask present in class D.

Class E

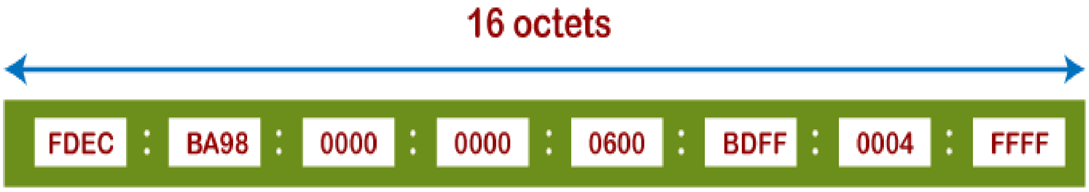
Class E IP address is reserved for experimental purposes and future use. It does not contain any subnet mask in it. The first higher octet bits are always set to 1111, and next remaining bits specify the host address. Class E address ranges between 240.0.0.0 to 255.255.255.255.


Offsets
0
8
16
24
Class A
0 Network
Host
Address 0.0.0.0 to 127.255.255.255
Class B
10 Network
Host
Address 128.0.0.0 to 191.255.255.255
Class C
110 Network
Host
Address 192.0.0.0 to 223.255.255
Class D
1110 Multicast address
Address 224.0.0.0 to 239.255.255.255
Class E
11110 Reserved for future use
Address 240.0.0.0. to 255.255.255.255

In every IP address class, all host-number bits are specified by a power of 2 that indicates the total numbers of the host's address that can create for a particular network address. Class A address can contain the maximum number of 2^{24} (16,777,216) host numbers. Class B addresses contain the maximum number of 2^{16} (65, 536) host numbers. And class C contains a maximum number of 2^{8} (256) host numbers.

IP Address Format IPv6

All IPv6 addresses are 128-bit hexadecimal addresses, written in 8 separate sections having each of them have 16 bits. As the IPv6 addresses are represented in a hexadecimal format, their sections range from 0 to FFFF. Each section is separated by colons (:). It also allows to removes the starting zeros (0) of each 16-bit section. If two or more consecutive sections 16-bit contains all zeros (0 : 0), they can be compressed using double colons (::).



IPv6 addresses are consist of 8 different sections, each section has a 16-bit hexadecimal values separated by colon (:). IPv6 addresses are represented as following format:

```
xxxx : xxxx : xxxx : xxxx : xxxx : xxxx : xxxx : xxxx
```

Each "xxxx" group contains a 16-bit hexadecimal value, and each "x" is a 4-bit hexadecimal value. For example:

```
FDEC : BA98 : 0000 : 0000 : 0600 : BDFF : 0004 : FFFF
```

You can also remove the starting zeros (0) of each 16-bit section. For example, the above IPv6 can be rewritten by omitting starting zeros (0) as follow:

```
FDEC : BA98 : 0 : 0 : 600 : BDFF : 4 : FFFF
```

You can also compress the consecutive sections 16-bit zeros (0 : 0) using double colons (::). But keep in mind that you can do it only once per IP address.

```
FDEC : BA98 : : 600 : BDFF : 4 : FFFF
```

Congestion Control Techniques and algorithms

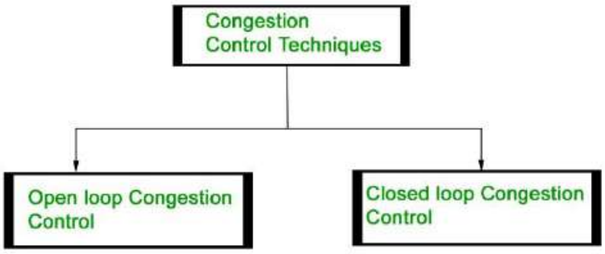
A state occurring in network layer when the message traffic is so heavy that it slows down network response time.

Effects of Congestion

- As delay increases, performance decreases.
- If delay increases, retransmission occurs, making situation worse.

Techniques:

Congestion control refers to the techniques used to control or prevent congestion. Congestion control techniques can be broadly classified into two categories:


Congestion Control Techniques
Open loop Congestion Control
Closed loop Congestion Control

Open Loop Congestion Control

Open loop congestion control policies are applied to prevent congestion before it happens. The congestion control is handled either by the source or the destination.

Policies adopted by open loop congestion control –

1. **Retransmission Policy :**
   It is the policy in which retransmission of the packets are taken care. If the sender feels that a sent packet is lost or corrupted, the packet needs to be retransmitted. This transmission may increase the congestion in the network.
   To prevent congestion, retransmission timers must be designed to prevent congestion and also able to optimize efficiency.
2. **Window Policy :**
   The type of window at the sender side may also affect the congestion. Several packets in the Go-back-n window are resent, although some packets may be received successfully at the receiver side. This duplication may increase the congestion in the network and making it worse.
   Therefore, Selective repeat window should be adopted as it sends the specific packet that may have been lost.
3. **Discarding Policy :**
   A good discarding policy adopted by the routers is that the routers may prevent congestion and at the same time partially discards the corrupted or less sensitive package and also able to maintain the quality of a message.
   In case of audio file transmission, routers can discard less sensitive packets to prevent congestion and also maintain the quality of the audio file.
4. **Acknowledgment Policy :**
   Since acknowledgement are also the part of the load in network, the acknowledgment policy imposed by the receiver may also affect congestion. Several approaches can be used to prevent congestion related to acknowledgment.

The receiver should send acknowledgement for N packets rather than sending acknowledgement for a single packet. The receiver should send a acknowledgment only if it has to sent a packet or a timer expires.

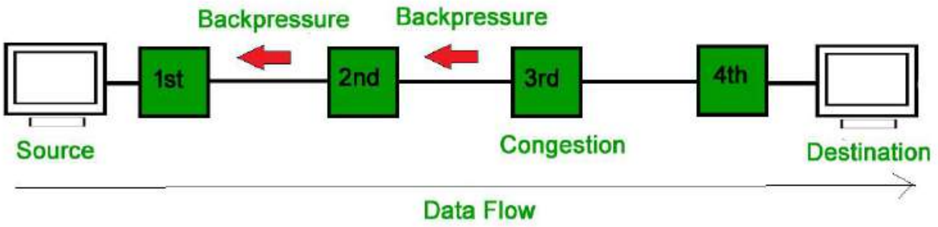
5. **Admission Policy :**
   In admission policy a mechanism should be used to prevent congestion. Switches in a flow should first check the resource requirement of a network flow before transmitting it further. If there is a chance of a congestion or there is a congestion in the network, router should deny establishing a virtual network connection to prevent further congestion.

All the above policies are adopted to prevent congestion before it happens in the network.

Closed Loop Congestion Control

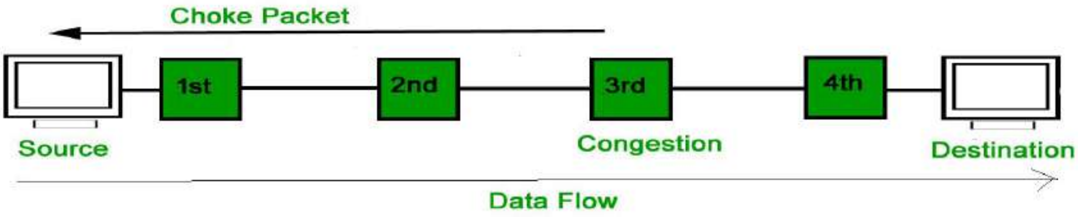
Closed loop congestion control technique is used to treat or alleviate congestion after it happens. Several techniques are used by different protocols; some of them are:

1. **Backpressure :**
   Backpressure is a technique in which a congested node stop receiving packet from upstream node. This may cause the upstream node or nodes to become congested and rejects receiving data from above nodes. Backpressure is a node-to-node congestion control technique that propagate in the opposite direction of data flow. The backpressure technique can be applied only to virtual circuit where each node has information of its above upstream node.



   In above diagram the 3rd node is congested and stops receiving packets as a result 2nd node may be get congested due to slowing down of the output data flow. Similarly 1st node may get congested and informs the source to slow down.

2. **Choke Packet Technique :**
   Choke packet technique is applicable to both virtual networks as well as datagram subnets. A choke packet is a packet sent by a node to the source to inform it of congestion. Each router monitor its resources and the utilization at each of its output lines. whenever the resource utilization exceeds the threshold value which is set by the administrator, the router directly sends a choke packet to the source giving it a feedback to reduce the traffic. The intermediate nodes through which the packets hastraveled are not warned about congestion.



1. **Implicit Signaling :**
   In implicit signaling, there is no communication between the congested nodes and the

source. The source guesses that there is congestion in a network. For example when sender sends several packets and there is no acknowledgment for a while, one assumption is that there is a congestion.

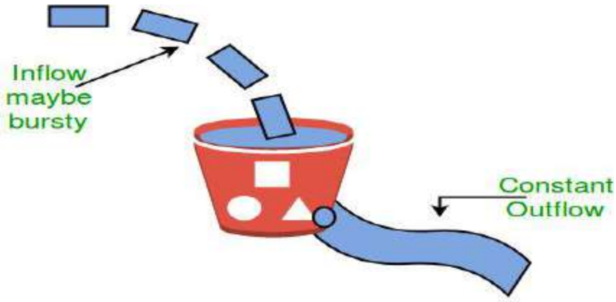
2. **Explicit Signaling :**
   In explicit signaling, if a node experiences congestion it can explicitly sends a packet to the source or destination to inform about congestion. The difference between choke packet and explicit signaling is that the signal is included in the packets that carry data rather than creating different packet as in case of choke packet technique. Explicit signaling can occur in either forward or backward direction.
   - **Forward Signaling :** In forward signaling signal is sent in the direction of the congestion. The destination is warned about congestion. The reciever in this case adopt policies to prevent further congestion.
   - **Backward Signaling :** In backward signaling signal is sent in the opposite direction of the congestion. The source is warned about congestion and it needs to slow down.

Congestion control algorithms

- **Leaky Bucket Algorithm**

Let us consider an example to understand

Imagine a bucket with a small hole in the bottom.No matter at what rate water enters the bucket, the outflow is at constant rate.When the bucket is full with water additional water entering spills over the sides and is lost.



Similarly, each network interface contains a leaky bucket and the following **steps** are involved in leaky bucket algorithm:

1. When host wants to send packet, packet is thrown into the bucket.
2. The bucket leaks at a constant rate, meaning the network interface transmits packets at a constant rate.
3. Bursty traffic is converted to a uniform traffic by the leaky bucket.
4. In practice the bucket is a finite queue that outputs at a finite rate.

Token bucket Algorithm

The leaky bucket algorithm enforces output pattern at the average rate, no matter how bursty the traffic is. So in order to deal with the bursty traffic we need a flexible algorithm so that the data is not lost. One such algorithm is token bucket algorithm.

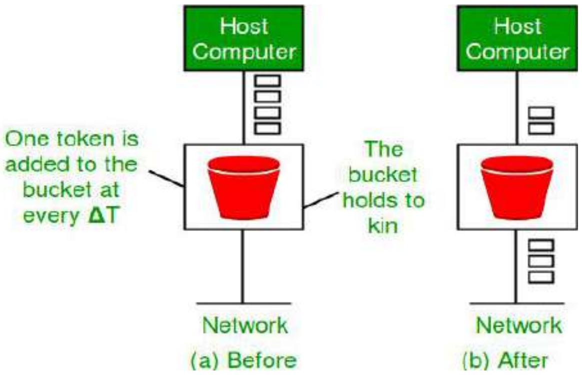
Steps of this algorithm can be described as follows:

1. In regular intervals tokens are thrown into the bucket. ƒ
2. The bucket has a maximum capacity. ƒ
3. If there is a ready packet, a token is removed from the bucket, and the packet is send.
4. If there is no token in the bucket, the packet cannot be send.

Let's understand with an example,

In figure (A) we see a bucket holding three tokens, with five packets waiting to be transmitted.For a packet to be transmitted, it must capture and destroy one token. In figure (B) We see that three of the five packets have gotten through, but the other two are stuck waiting for more tokens to be generated.

Let's understand with an example,

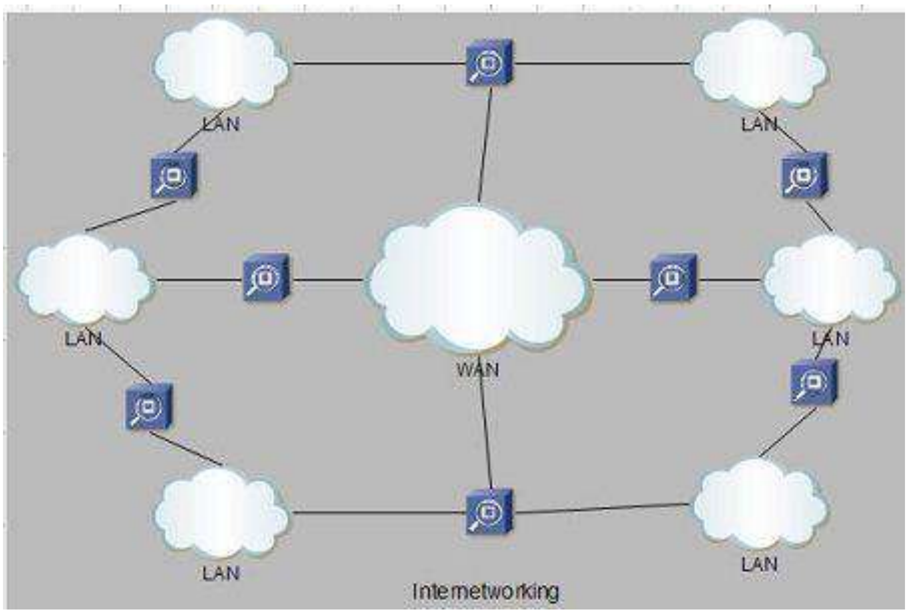


Internetworking

Internetworking started as a way to connect disparate types of computer networking technology. Computer network term is used to describe two or more computers that are linked to each other. When two or more computer LANs or WANs or computer network segments are connected using devices such as a *router* and configure by logical addressing scheme with a protocol such as IP, then it is called as **computer internetworking**.

Internetworking is a term used by Cisco. Any interconnection among or between public, private, commercial, industrial, or governmental computer networks may also be defined as an internetwork or "**Internetworking**".

In modern practice, the interconnected computer networks or **Internetworking** use the Internet Protocol. Two architectural models are commonly used to describe the protocols and methods used in **internetworking**. The standard reference model for **internetworking** is Open Systems Interconnection (**OSI)**.



Type of Internetworking

Internetworking is implemented in Layer 3 (Network Layer) of this model The most notable example of internetworking is the Internet (capitalized). There are three variants of internetwork or **Internetworking**, depending on who administers and who participates in them :

- Extranet
- Intranet
- Internet

Intranets and extranets may or may not have connections to the Internet. If connected to the Internet, the intranet or extranet is normally protected from being accessed from the Internet without proper authorization. The Internet is not considered to be a part of the intranet or extranet, although it may serve as a portal for access to portions of an extranet.

Extranet

An extranet is a **network of internetwork or Internetworking** that is limited in scope to a **single organisation or entity** but which also has **limited connections** to the networks of one or more other usually, but not necessarily, trusted organizations or entities .Technically, an **extranet may also be categorized as a MAN, WAN**, or other type of network, although, by definition, an extranet cannot consist of a single LAN; it must have at least one connection with an external network.

Intranet

An intranet is a set of **interconnected networks or Internetworking**, using the I**nternet Protocol** and uses **IP-based tools** such as **web browsers** and **ftp tools**, that is under the control of a **single administrative entity.** That administrative entity closes the intranet to the rest of the world, and allows only specific users. Most commonly, an intranet is the internal network of a company or other enterprise. A large intranet will typically have its **own web server** to provide users with browseable information.

Internet

A specific **Internetworking**, consisting of a **worldwide interconnection** of governmental, academic, public, and private networks based upon the Advanced Research Projects Agency Network (**ARPANET)** developed by ARPA of the U.S. **Department of Defense** also **home** to the **World Wide Web (WWW)** and referred to as the '**Internet**' with a capital 'I' to distinguish it from other generic internetworks. Participants in the Internet, or their service providers, use IP Addresses obtained from address registries that control assignments.

ICMP

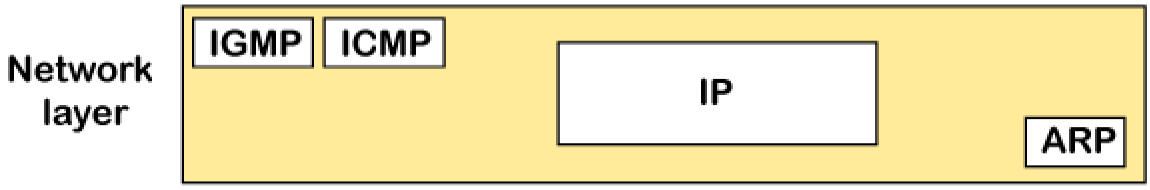
The ICMP stands for Internet Control Message Protocol. It is a network layer protocol. It is used for error handling in the network layer, and it is primarily used on network devices such as routers. As different types of errors can exist in the network layer, so ICMP can be used to report these errors and to debug those errors.

For example, some sender wants to send the message to some destination, but the router couldn't send the message to the destination. In this case, the router sends the message to the sender that I could not send the message to that destination.

The IP protocol does not have any error-reporting or error-correcting mechanism, so it uses a message to convey the information. For example, if someone sends the message to the destination, the message is somehow stolen between the sender and the destination. If no one reports the error, then the sender might think that the message has reached the destination. If someone in-between reports the error, then the sender will resend the message very quickly.

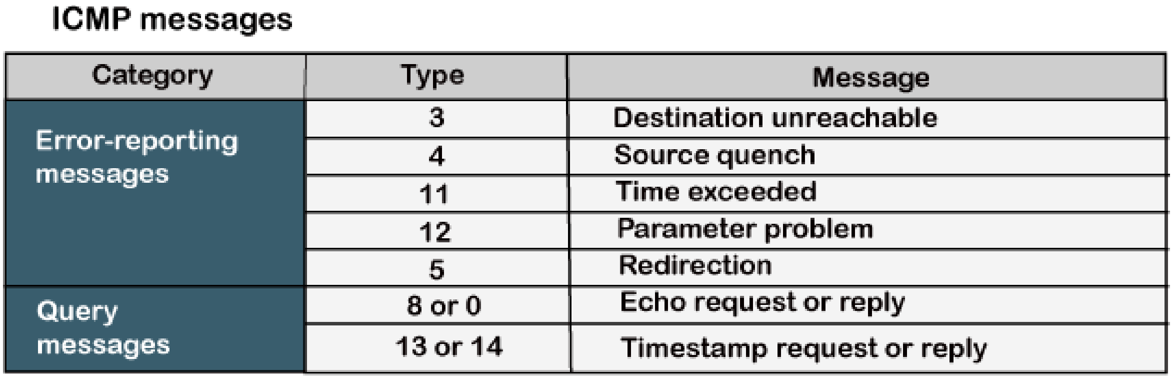
Position of ICMP in the network layer

The ICMP resides in the IP layer, as shown in the below diagram.



Messages

The ICMP messages are usually divided into two categories:



ICMP messages

Category	Type	Message
Error-reporting messages	3	Destination unreachable
	4	Source quench
	11	Time exceeded
	12	Parameter problem
	5	Redirection
Query messages	8 or 0	Echo request or reply
	13 or 14	Timestamp request or reply

- **Error-reporting messages**

The error-reporting message means that the router encounters a problem when it processes an IP packet then it reports a message.

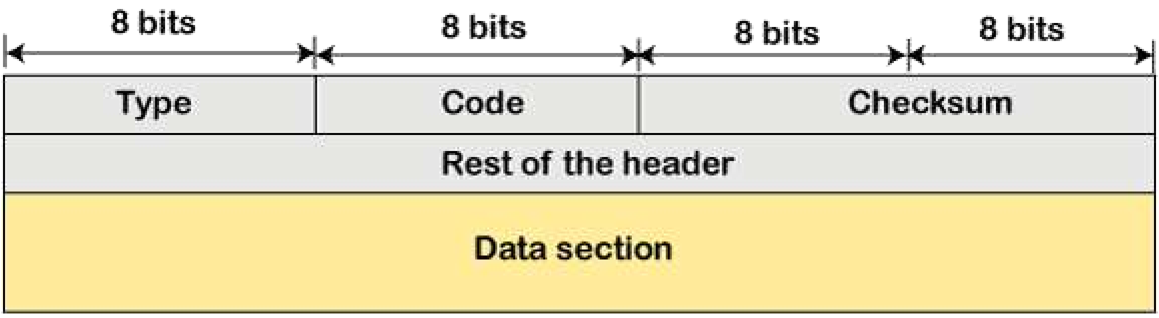
- **Query messages**

The query messages are those messages that help the host to get the specific information of another host. For example, suppose there are a client and a server, and the client wants to know whether the server is live or not, then it sends the ICMP message to the server.

ICMP Message Format

The message format has two things; one is a category that tells us which type of message it is. If the message is of error type, the error message contains the type and the code. The type defines the type of message while the code defines the subtype of the message.

The ICMP message contains the following fields:



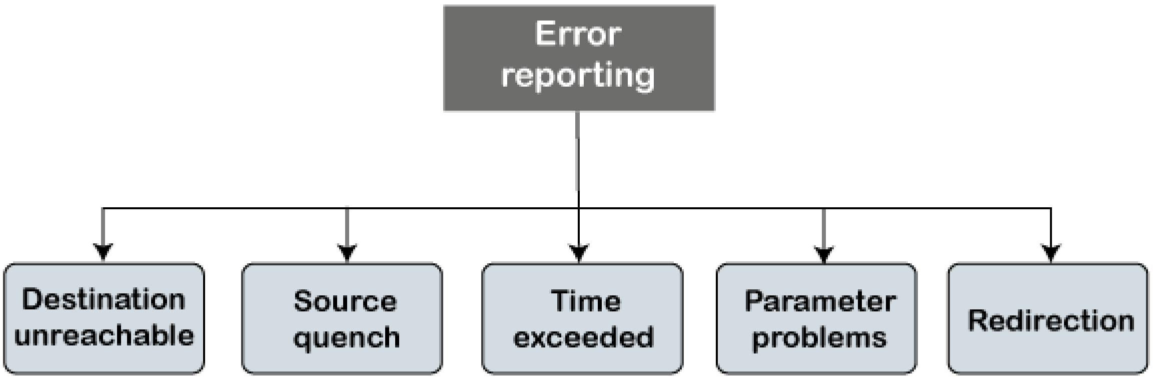
Type	Code	Checksum
Rest of the header		
Data section		

- **Type:** It is an 8-bit field. It defines the ICMP message type. The values range from 0 to 127 are defined for ICMPv6, and the values from 128 to 255 are the informational messages.
- **Code:** It is an 8-bit field that defines the subtype of the ICMP message
- **Checksum:** It is a 16-bit field to detect whether the error exists in the message or not.

> Note: The ICMP protocol always reports the error messages to the original source. For example, when the sender sends the message, if any error occurs in the message then the router reports to the sender rather than the receiver as the sender is sending the message.

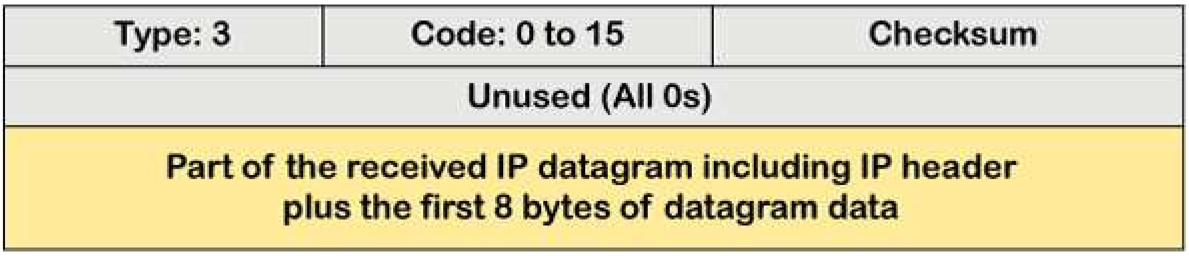
Types of Error Reporting messages

The error reporting messages are broadly classified into the following categories:



- **Destination unreachable**

The destination unreachable error occurs when the packet does not reach the destination. Suppose the sender sends the message, but the message does not reach the destination, then the intermediate router reports to the sender that the destination is unreachable.



Type: 3	Code: 0 to 15	Checksum
Unused (All 0s)		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

The above diagram shows the message format of the destination unreachable message. In the message format:

Type: It defines the type of message. The number 3 specifies that the destination is unreachable.

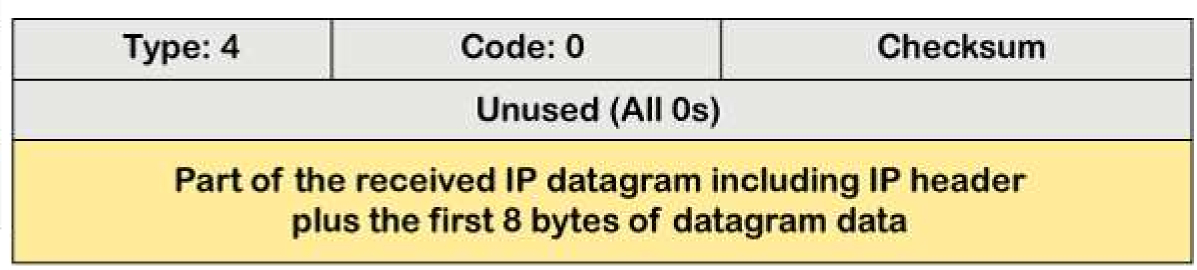
Code (0 to 15): It is a 4-bit number which identifies whether the message comes from some intermediate router or the destination itself.

Note: If the destination creates the destination unreachable message then the code could be either 2 or 3.

Sometimes the destination does not want to process the request, so it sends the destination unreachable message to the source. A router does not detect all the problems that prevent the delivery of a packet.

- **Source quench**

There is no flow control or congestion control mechanism in the network layer or the IP protocol. The sender is concerned with only sending the packets, and the sender does not think whether the receiver is ready to receive those packets or is there any congestion occurs in the network layer so that the sender can send a lesser number of packets, so there is no flow control or congestion control mechanism. In this case, ICMP provides feedback, i.e., source quench. Suppose the sender resends the packet at a higher rate, and the router is not able to handle the high data rate. To overcome such a situation, the router sends a source quench message to tell the sender to send the packet at a lower rate.



Type: 4	Code: 0	Checksum
Unused (All 0s)		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

The above diagram shows the message format of the source quench message. It is a type 4 message, and code is zero.

Note: A source quench message informs the sender that the datagram has been discarded due to the congestion occurs in the network layer.

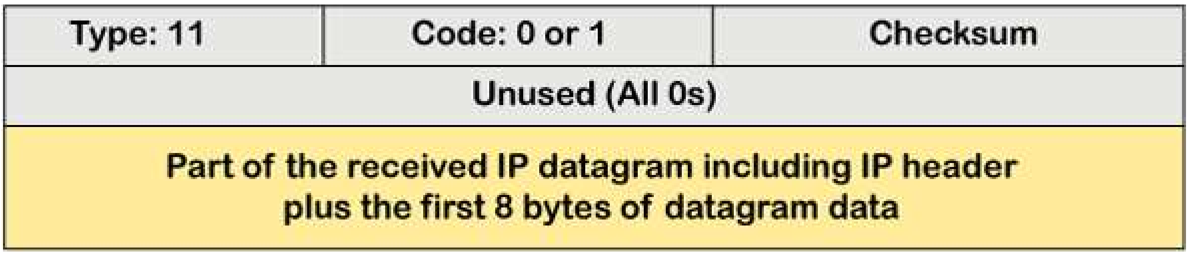
So, the sender must either stop or slow down the sending of datagrams until the congestion is reduced. The router sends one source-quench message for each datagram that is discarded due to the congestion in the network layer.

- **Time exceeded**

Sometimes the situation arises when there are many routers that exist between the sender and the receiver. When the sender sends the packet, then it moves in a routing loop. The time exceeded is based on the time-to-live value. When the packet traverses through the router, then each router decreases the value of TTL by one. Whenever a router decreases a datagram with a time-to-live value to zero, then the router discards a datagram and sends the time exceeded message to the original source.

Each of the MAC layers has different data units. For example, some layers can handle upto 1500 data units, and some can handle upto 300 units. When the packet is sent from a layer having 1500 units to the layer having 300 units, then the packet is divided into fragments; this process is known as fragmentation. These 1500 units are divided into 5 fragments, i.e., f1, f2, f3, f4, f5, and these fragments reach the destination in a sequence. If all the fragments are not reached to the destination in a set time, they discard all the received fragments and send a time-exceeded message to the original source.

In the case of fragmentation, the code will be different as compared to TTL. Let's observe the message format of time exceeded.



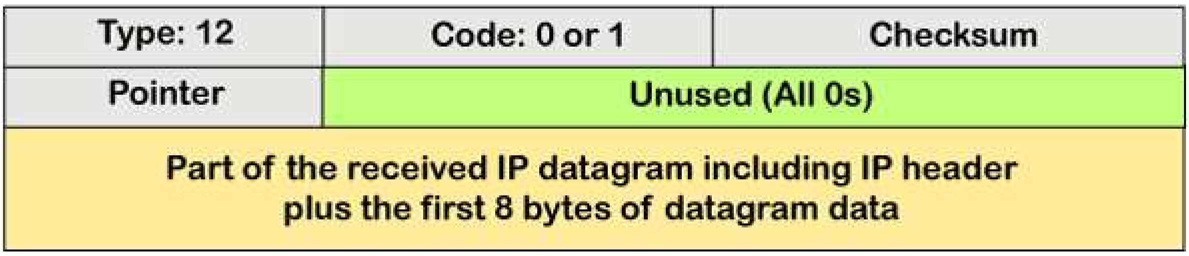
Type: 11	Code: 0 or 1	Checksum
Unused (All 0s)		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

The above message format shows that the type of time-exceeded is 11, and the code can be either 0 or 1. The code 0 represents TTL, while code 1 represents fragmentation. In a time-exceeded message, the code 0 is used by the routers to show that the time-to-live value is reached to zero.

The code 1 is used by the destination to show that all the fragments do not reach within a set time.

Parameter problems

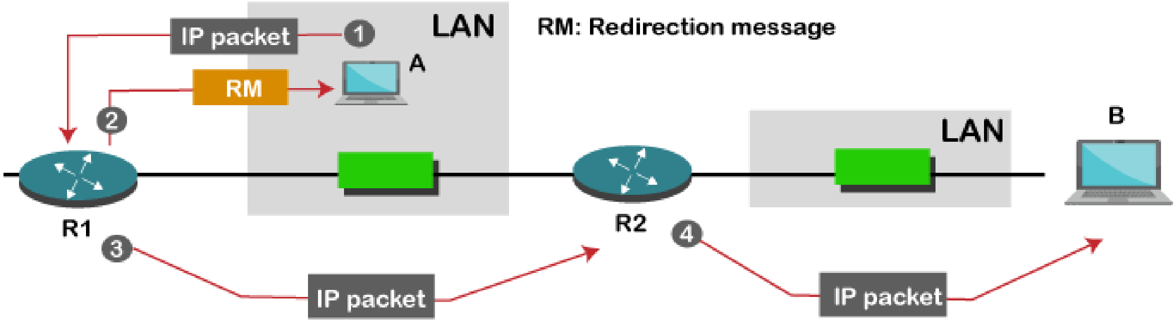
The router and the destination host can send a parameter problem message. This message conveys that some parameters are not properly set.



Type: 12	Code: 0 or 1	Checksum
Pointer	Unused (All 0s)	
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

The above diagram shows the message format of the parameter problem. The type of message is 12, and the code can be 0 or 1.

Redirection



When the packet is sent, then the routing table is gradually augmented and updated. The tool used to achieve this is the redirection message. For example, A wants to send the packet to B, and there are two routers exist between A and B. First, A sends the data to the router 1. The router 1 sends the IP packet to router 2 and redirection message to A so that A can update its routing table.

Note: A redirection message is sent from the router to the host on the same network.

ICMP Query Messages

The ICMP Query message is used for error handling or debugging the internet. This message is commonly used to ping a message.

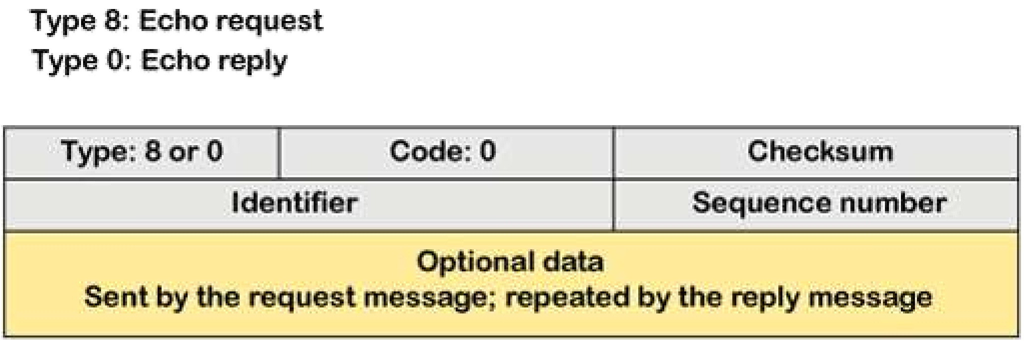
Echo-request and echo-reply message

A router or a host can send an echo-request message. It is used to ping a message to another host that "Are you alive". If the other host is alive, then it sends the echo-reply message. An echo-reply message is sent by the router or the host that receives an echo-request message.

Key points of Query messages

1. The echo-request message and echo-reply message can be used by the network managers to check the operation of the IP protocol. Suppose two hosts, i.e., A and B, exist, and A wants to communicate with host B. The A host can communicate to host B if the link is not broken between A and B, and B is still alive.
2. The echo-request message and echo-reply message check the host's reachability, and it can be done by invoking the ping command.

The message format of echo-request and echo-reply message



Type 8: Echo request
Type 0: Echo reply

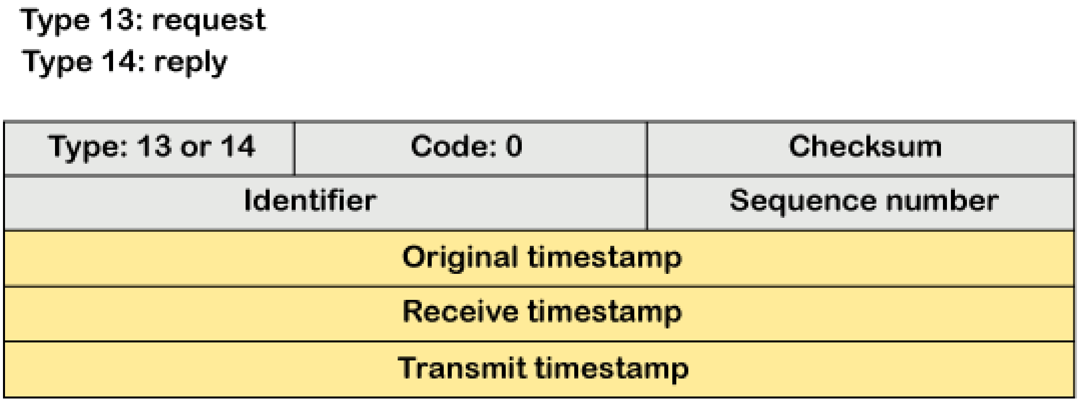
Type: 8 or 0	Code: 0	Checksum
Identifier		Sequence number
Optional data Sent by the request message; repeated by the reply message		

The above diagram shows the message format of the echo-request and echo-reply message. The type of echo-request is 8, and the request of echo-reply is 0. The code of this message is 0.

Timestamp-request and timestamp-reply message

The timestamp-request and timestamp-reply messages are also a type of query messages. Suppose the computer A wants to know the time on computer B, so it sends the timestamp-request message to computer B. The computer B responds with a timestamp-reply message.

Message format of timestamp-request and timestamp-reply



Type 13: request
Type 14: reply

Type: 13 or 14	Code: 0	Checksum
Identifier		Sequence number
Original timestamp		
Receive timestamp		
Transmit timestamp		

The type of timestamp-request is 13, and the type of timestamp-reply is 14. The code of this type of message is 0.

Key points related to timestamp-request and timestamp-reply message

- It can be used to calculate the round-trip time between the source and the destination, even if the clocks are not synchronized.
- It can also be used to synchronize the clocks in two different machines if the exact transit time is known.

If the sender knows the exact transit time, then it can synchronize the clock. The sender asks the time on the receiver's clock, and then it adds the time and propagation delay. Suppose the time is 1:00 clock and propagation delay is 100 ms, then time would be 1:00 clock plus 100 ms.