LAKSHMI NARAIN COLLEGE OF TECHNOLOGY & SCIENCE, BHOPAL

CS-602
COMPUTER NETWORK

Vision of the Department

To be recognized for keeping innovation, research and excellence abreast of learning in the field of computer science & engineering to cater the global society.

Mission of the Department

M1: To provide an exceptional learning environment with academic excellence in the field of computer science and engineering.

M2: To facilitate the students for research and innovation in the field of software, hardware and computer applications and nurturing to cater the global society.

M3: To establish professional relationships with industrial and research organisations to enable the students to be updated of the recent technological advancements.

M4: To groom the learners for being the software professionals catering the needs of modern society with ethics, moral values and full of patriotism.

Program Educational Objectives (PEO’s)

PEO1: The graduate will have the knowledge and skills of major domains of computer science and engineering in providing solution to real world problems most efficiently.

PEO2: The graduate will be able to create and use the modern tools and procedures followed in the software industry in the relevant domain.

PEO3: The graduate will be following the ethical practices of the software industry and contributing to the society as a responsible citizen.

PEO4: The graduate will have the innovative mindset of learning and implementing the latest developments and research outcomes in the computer hardware and software to keep pace with the fast changing socio economic world.

LAKSHMI NARAIN COLLEGE OF TECHNOLOGY & SCIENCE, BHOPAL

CS-602
COMPUTER NETWORK

COURSE OUTCOMES

CO1: Compare computer network protocol hierarchy of OSI and TCP/IP models.

CO2: Demonstrate mechanisms of data link layer and related protocols to avoid collision and congestion.

CO3: Compare various data transmission protocol.

CO4: Evaluate efficiency of various routing algorithms.

CO5: Differentiate IPv4 and IPv6 internet protocol.

LIST OF EXPERIMENTS

1. Study of Different Type of LAN& Network Equipments.
2. Study and Verification of standard Network topologies i.e. Star, Bus, Ring etc.
3. LAN installations and Configurations.
4. Write a program to implement various types of error correcting techniques.
5. Write a program to implement various types of framing methods.
6. Study of Tool Command Language (TCL).
7. Implement & Simulate various types of routing algorithm.
8. Study & Simulation of MAC Protocols like Aloha, CSMA, CSMA/CD and CSMA/CA using Standard Network Simulators.
9. Study of Application layer protocols-DNS, HTTP, HTTPS, FTP and TelNet.
10. Configure 802.11 WLAN.

CS-602

COMPUTER NETWORK

EXPERIMENT -1

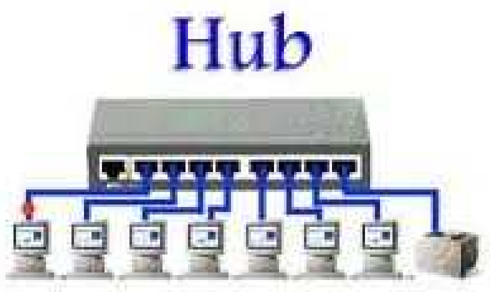
Aim: Study of Different Type of LAN& Network Equipments.

1. Repeater
2. Hub
3. Switch
4. Bridge
5. Router
6. Gate Way

Theory

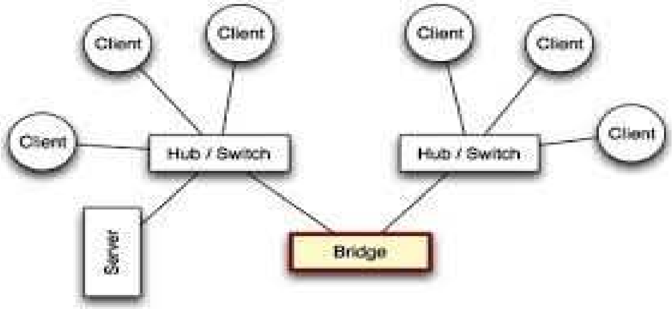
1. **Repeater:** Functioning at Physical Layer. A repeater is an electronic device that receives a signal and retransmits it at a higher level and/or higher power, or onto the other side of an obstruction, so that the signal can cover longer distances. Repeaters have two ports, so cannot be used to connect for more than two devices.

2. **Hub:** An Ethernet hub, active hub, network hub, repeater hub, hub or concentrator is a device for connecting multiple twisted pair or fiber optic Ethernet devices together and making them act as a single network segment. Hubs work at the physical layer of the OSI model. The device is a form of multiport repeater. Repeater hubs also participate in collision detection, forwarding a jam signal to all ports if it detects a collision.

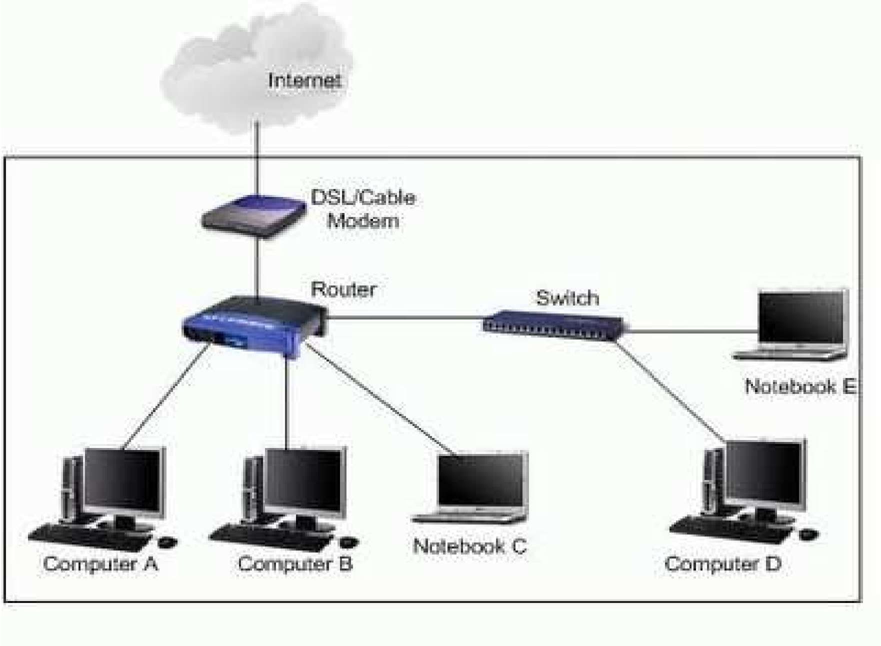


Switch: A network switch or switching hub is a computer networking device that connects network segments. The term commonly refers to a network bridge that processes and routes data at the data link layer of the OSI model. Switch that additionally process data at the network layer are often referred to as layer 3 switches or multilayer switches.

Bridge: A network bridge connects multiple network segments at the data link layer of the OSI model. In Ethernet networks, the term bridge formally means a device that behaves according to the IEEE 802.1 D standards. A bridge and switch are very much alike; a switch being a bridge with numerous ports. Switch of layer 2 switch is often used interchangeably with bridge. Bridges can analyze incoming data packets to determine if the bridge is able to send the given packet to another segment of the network.

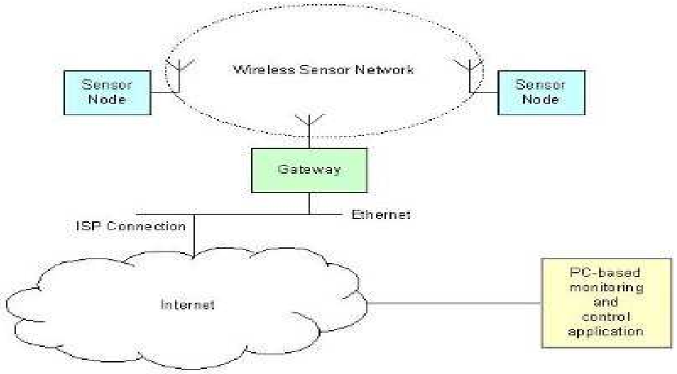


Router: A router is an electronic device that interconnects two or more computer networks, and selectively interchanges packets of data between them. Each data packet contains address information that a router can use to determine if the source and destination are on the same network, or if the data packet must be transferred from one network to another. Where multiple routers are used in a large collection of interconnected networks, the routers exchange information about target system addresses, so that each router can build up a table showing the preferred paths between any two systems on the interconnected networks.



Gate way: In a communications network, a network node equipped for interfacing with another network that uses different protocols.A gateway may contain devices such as protocol translators, impedance matching devices, rate converters, fault isolators, or signal translators as necessary to provide system interoperability. It alsorequires the establishment of mutually acceptable administrative procedures between both networks.

A protocol translation/mapping gateway interconnects networks with different network protocol technologies by performing the required protocol conversions.



Conclusions

Different Network Devices have been studied in detail.

EXPERIMENT -2

Aim: Study and Verification of standard Network topologies i.e. Star, Bus, Ring etc.

Theory

Network topology is the arrangement of the various elements (links, nodes, etc.) of a computer or biological network. Essentially, it is the topological structure of a network, and may be depicted physically or logically. *Physical* topology refers to the placement of the network's various components, including device location and cable installation, while *logical* topology shows how data flows within a network, regardless of its physical design. Distances between nodes, physical interconnections, transmission rates, and/or signal types may differ between two networks, yet their topologies may be identical.

A good example is a local area network (LAN): Any given node in the LAN has one or more physical links to other devices in the network; graphically mapping these links results in a geometric shape that can be used to describe the physical topology of the network. Conversely, mapping the data flow between the components determines the logical topology of the network.

Topology

There are two basic categories of network topologies:

1. Physical topologies
2. Logical topologies

The shape of the cabling layout used to link devices is called the physical topology of the network. This refers to the layout of cabling, the locations of nodes, and the interconnections between the nodes and the cabling The physical topology of a network is determined by the capabilities of the network access devices and media, the level of control or fault tolerance desired, and the cost associated with cabling or telecommunications circuits.

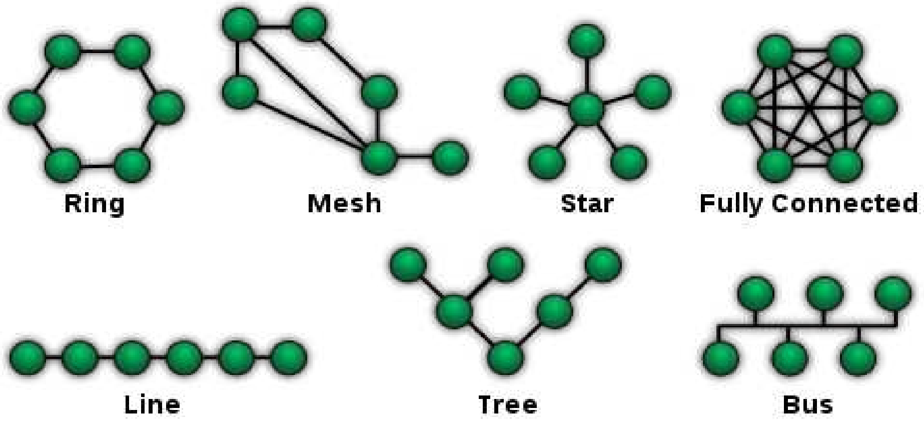
The logical topology, in contrast, is the way that the signals act on the network media, or the way that the data passes through the network from one device to the next without regard to the physical interconnection of the devices. A network's logical topology is not necessarily the same as its physical topology. For example, the original twisted pair Ethernet using repeater hubs was a logical bus topology with a physical star topology layout. Token Ring is a logical ring topology, but is wired a physical star from the Media Access Unit.

The logical classification of network topologies generally follows the same classifications as those in the physical classifications of network topologies but describes the path that the *data* takes between nodes being used as opposed to the actual *physical* connections between nodes. The logical topologies are generally determined by network protocols as opposed to being determined by the physical layout of cables, wires, and network devices or by the flow of the electrical signals, although in many cases the paths that the electrical signals take between nodes may closely match the logical flow of data, hence the convention of using the terms *logical topology* and *signal topology* interchangeably.

Logical topologies are often closely associated with Media Access Control methods and protocols. Logical topologies are able to be dynamically reconfigured by special types of equipment such as routers and switches.

The study of network topology recognizes eight basic topologies:

- Point-to-point
- Bus
- Star
- Ring or circular
- Mesh
- Tree
- Hybrid
- Daisy chain



EXPERIMENT -3

Aim: LAN installations and their Configurations.

Apparatus Required

Theory

Performance:

Perform the following steps as directed

Step 1: To make a Direct Cable connection

1. Click **Start**, click **Control Panel**, and then double-click **Network Connections**.

2. Under **Network Tasks**, click **Create a new connection**, and then click **next**.

3. Click **Set up an advanced connection**, and then click **next**.

4. Click **Connect directly to another computer**, and click **next**.

5. Choose the role this machine will play in the communication. If this computer has the information to which you need to gain access, click **Host**. If this computer will access information from the other computer, click **Guest**.

Step 2: To Set Up the Host Computer

1. Click the connection device that you want to use for this connection (a parallel or serial port, or an infrared port), and then click **Next**.

2. Grant access to the users who are allowed to connect by selecting the appropriate check boxes, and then click **Next**.

3. Click **Finish** to end the configuration process.

Step 3: to Set up the Guest Computer

1. Type a name to identify this connection, and then click **Next**.

2. Click the connection device that you want to use for this connection (a parallel or serial port, or an infrared port), and then click **Next**.

3. Decide whether this connection will be available for all users (click **Anyone's use**), or only for you (click **my use only**), and then click **Next**.

4. Click **Finish** to end the setup process

Step 4: To create Windows Workgroup

1. In Windows XP, right click on **My Computer**, select **System Properties**.

2. Select the **Computer Name** tab, click on **Change**.

3. Enter the appropriate **Computer name** and **Workgroup**.

4. Make sure that every computer on your home network references the same workgroup.

Step 5:

To Configure TCP/IP

To assign IP address, gateway, subnet mask, DNS

Step 6:

To create domain Bring all the PC of Lab under a network using workgroup or domain.

Create client and server

Result/ Conclusions

Windows workgroup is established and used for sharing and transferring data between physically connected PCs.

EXPERIMENT NO. 4

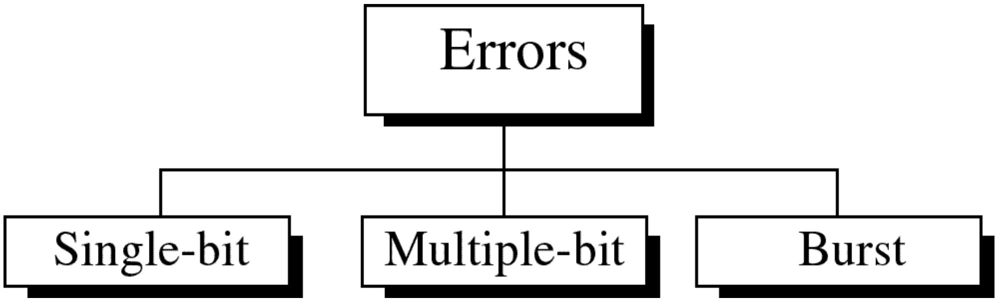
Aim: To implement various types of error correcting techniques.

Apparatus Required

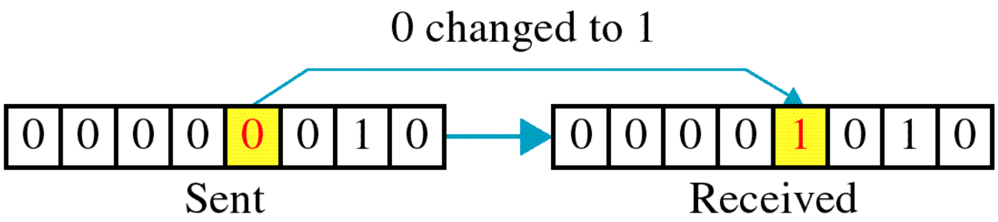
Theory

Error detection and correction are implemented either at the data link layer or the transport layer of the OSI model.

Types of Errors



Single-bit error

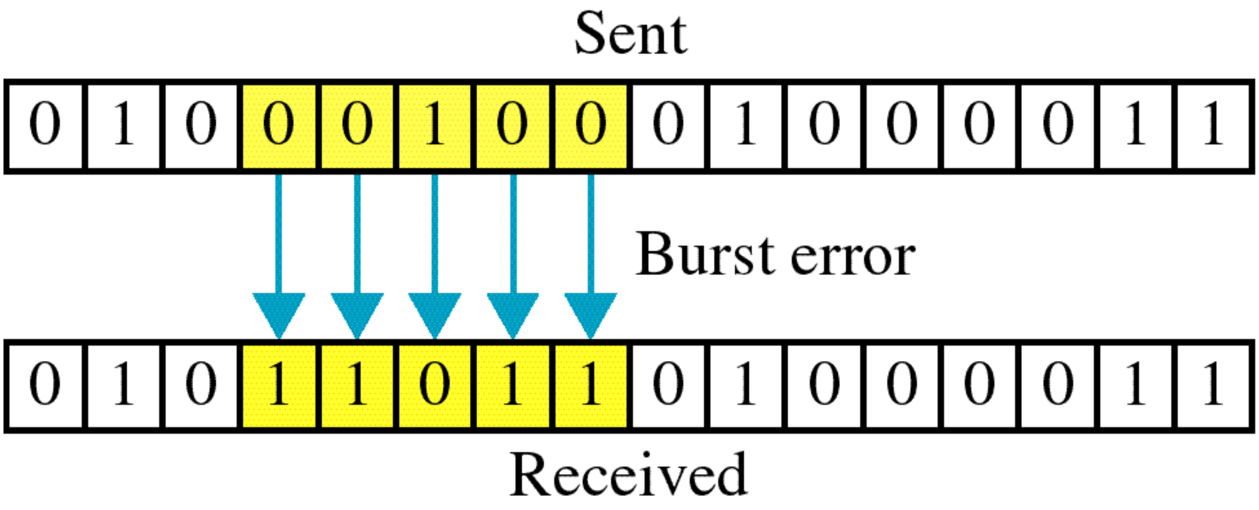


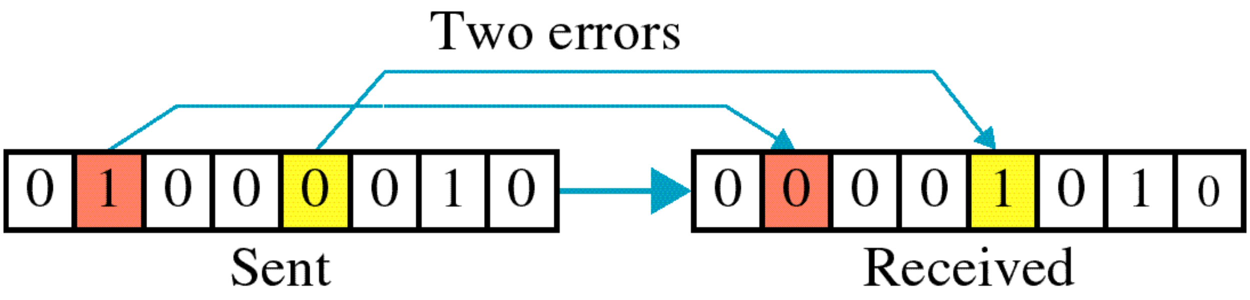
Single bit errors are the **least likely** type of errors in serial data transmission because the noise must have a very short duration which is very rare. However this kind of errors can happen in parallel transmission.

Example:

- If data is sent at 1Mbps then each bit lasts only 1/1,000,000 sec. or 1 µs.
- For a single-bit error to occur, the noise must have a duration of only 1 µs, which is very rare.

Burst error





The term **burst error** means that two or more bits in the data unit have changed from 1 to 0 or from 0 to 1.

Burst errors do not necessarily **mean that the errors occur in consecutive bits**, the length of the burst is measured from the first corrupted bit to the last corrupted bit. Some bits in between may not have been corrupted.

Burst error is most likely to happen in serial transmission since the duration of noise is normally longer than the duration of a bit.

The number of bits affected depends on the data rate and duration of noise.

Example:

If data is sent at rate = 1Kbps then a noise of 1/100 sec can affect 10 bits.(1/100*1000)

If same data is sent at rate = 1Mbps then a noise of 1/100 sec can affect 10,000 bits.(1/100*10^6)

CYCLIC REDUNDANCY CHECK (CRC)

LOGIC:

1. Let r be the degree of G(x).Append r zero bits to the low-order end of the frame. So it now contains m+r bits and corresponds to the polynomial x2 m(x).
2. Divide the bit string corresponding to G(x) into the bit string corresponding to x2 m(x) using modulo-2 division.
3. Subtract the remainder from the bit string corresponding to x2 m(x) using modulo-2 sub. The result is the check summed frame to be transmitted. We call it as a polynomial.

<u>EXPERIMENT -5</u>

Write a program to implement various types of framing methods.

a) CHARACTER COUNT

b) BIT STUFFING AND DE STUFFING

c) CHARACTER STUFFING AND DE STUFFING

CHARACTER COUNT

Implementation of data link framing methods for counting characters in a given frame

LOGIC: The header in the given frame is by default first frame size including the first field data bits are counted and considered as the first frame and the next field contains the next frame size and so on. **PSEUDO CODE**:

1. At the sender side the user is asked to enter the number of frames he want to transmit.
2. Depending upon the input, that much number of frames are taken as input from the user and stored in a 2 by 2 matrix.
3. The length of each frame is calculated and stored in a new array.

4. While out putting the frame, the length of each frame is added to the each frame and finally all the frames are appended and sent as a single string.
5. At the receiver side, the first number is treated as the length of the first frame and the string is extracted and displayed.
6. The next number is treated as the length of the next frame and so on.

At Sender:

INPUT:

```
Enter the number of frames you want to send: 2

Enter the frame: 1234

Enter the frame: 678
```

OUTPUT

```
The transmitted frame is: 512344678
```

At receiver:

INPUT:

```
Enter data 512344678
```

OUTPUT:

```
Frame sizes are: 5 4

Frames are:

Frame 1: 1234

Frame 2: 678
```

BIT STUFFING

Implementation of the data link framing methods for the bit stuffing in a frame.

LOGIC: Stuffing a 0 bit in a data frame in order to differentiate the header, trailer and data.

PSEUDO CODE: /* For given data */

```
1. a flag "01111110" is embedded at the starting and the ending of the data. /* stuffing of data */
2. if data bit is 1 increment count else count is zero.
3. If count is five store a zero bit after the five 1‟s in the data array.
4. Repeat step 3 till the end of data. /* De stuffing of data */
5. If the data bit is 1 increment count else count is zero.
6. If the count is five and the next bit is zero then store the next bit after zero in the data array. /* transmit the data */
7. De stuffed data is transmitted without flags.
```

At sender:
INPUT:
Enter the string 1111110101
OUTPUT:
Transmitted data is: 01111110111111010101111110
Stuffed data is: 011111101111101010101111110

At Receiver:
INPUT:
Enter the string 1111110101
OUTPUT:
De stuffed data is: 11111110101

CHARACTER STUFFING

AIM

Implementation of data link framing methods for character stuffing in a frame.

LOGIC: In a character data frame if a DLE is encountered between the data it is doubled and transmitted at the receiver side it is de stuffed and original data is obtained.

PSEUDO CODE: /* Defining DLE characters */

1. As the DLE characters are non-printable characters. The ASCII values of the printable characters like *, #, $ are assigned to DLE, STX, ETX.

/*Stuffing the data */

2. If the ASCII value that is assigned to DLE occurs in the data array another DLE character is stuffed and stored in the array and transmitted along with starting and ending flags /* DE stuffing data */

3. If the ASCII value of DLE occurs in the data array, the next bit is stored in to the array and transmitted without the flags.

4. Here whenever the program encounters characters like * the string DLE is added to the original string.

At Sender:

```
INPUT:
Enter Data r*gm
OUTPUT:
Stuffed data DLESTXrDLEDLEgmDLEETX
```

At receiver:

```
OUTPUT:
The message: r*gm
```

EXPERIMENT NO. 6

Aim - Study of Tool Command Language (Tcl)

Tcl (originally from "Tool Command Language", but conventionally spelled "Tcl" rather than "TCL"; pronounced as "tickle" or "tee-see-ell") is a scripting language created by John Ouster out.Originally "born out of frustration”, according to the author, with programmers devising their own languages intended to be embedded into applications, Tcl gained acceptance on its own. It is commonly used for rapid prototyping, scripted applications, GUIs and testing. Tcl is used on embedded systems platforms, both in its full form and in several other small-footprint versions.

The combination of Tcl and the TkGUI toolkit is referred to as **Tcl/Tk**.

Features

Tcl's features include

- All operations are commands, including language structures. They are written in prefix notation.
- Commands are commonly variadic.
- Everything can be dynamically redefined and overridden.
- All data types can be manipulated as strings, including source code.
- Event-driven interface to sockets and files. Time-based and user-defined events are also possible.
- Variable visibility restricted to lexical (static) scope by default, but uplevel and upvar allowing process to interact with the enclosing functions' scopes.
- All commands defined by Tcl itself generate error messages on incorrect usage.
- Extensibility, via C, C++, Java, and Tcl.
- Interpreted language using byte code
- Full Unicode (3.1) support, first released 1999.
- Cross-platform: Windows API; UNIX, Linux, Macintosh, etc.

- Close integration with windowing (GUI) interface Tk.
- Multiple distribution mechanisms exist:
    - Full development version (e.g., Active State Tcl)
    - tclkit (kind of single-file runtime, only about 1 megabyte in size)
    - starpack (single-file executable of a script/program, derived from the tclkit technology)
    - FreewrapTCLSH turns TCL scripts into single-file binary executable programs.
    - BSD licenses, freely distributable source.

Tcl did not originally have object oriented (OO) syntax (8.6 provides an OO system in Tcl core), so OO functionality was provided by extension packages, such as incr Tcl and XOTcl. Even purely scripted OO packages exist, such as Snit and STOOOP (simple Tcl-only object-oriented programming).

Safe-Tcl is a subset of TCL that has restricted features. File system access is limited and arbitrary system commands are prevented from execution. It uses a dual interpreter model with the "untrusted interpreter" running code in an untrusted script. It was designed by Nathaniel Bornstein and Marshall Rose to include active messages in e-mail. Safe-Tcl can be included in e-mail when the *application/safe-Tcl* and *multipart/enabled-mail* are supported. The functionality of Safe-Tcl has since been incorporated as part of the standard Tcl/Tk releases.

Syntax and fundamental semantics

A Tcl script consists of several command invocations. A command invocation is a list of words separated by whitespace and terminated by a newline or semicolon.

word0 word1 word2 ... wordN

The first word is the name of a command, which is not built into the language, but which is in the library. The following words are arguments. So we have:

commandName argument1 argument2 ... argumentN

- Practical example, using the puts command which outputs a string, adding a trailing newline, by default to the stdout channel:
- puts"Hello, world!"

- Variables and the results of other commands can be substituted inside strings too, such as in this example where we use set and expr to store a calculation result in a variable, and puts to print the result together with some explanatory text:
- # Good style would put the expression (1+2+3+4+5, in this case) inside {curly braces}
- set sum [expr1+2+3+4+5]
- puts"The sum of the numbers 1..5 is $sum."
-
- #expr function will be evaluated faster if curly braces are added on the equation.
-
- set sum [expr{1+2+3+4+5}]
- puts"The sum of the numbers 1..5 is $sum."
- There is one basic construct (the command) and a set of simple substitution rules.

Formally, words are either written as-is, with double-quotes around them (allowing whitespace characters to be embedded), or with curly-brace characters around them, which suppresses all substitutions inside (except for backslash-newline elimination). In bare and double-quoted words, three types of substitution occur (once, in a single left-to-right scan through the word):

- **Command substitution** replaces the contents of balanced square brackets with the result of evaluating the script contained inside. For example, "**[expr 1+2+3]**" is replaced with the result of evaluating the contained expression (i.e. 6) since that's what the expr command does.
- **Variable substitution** replaces a dollar-sign followed by the name of a variable with the contents of the variable. For example, "**$foo**" is replaced with the contents of the variable called "foo". The variable name may be surrounded in curly braces so as to delimit what is and isn't the variable name in otherwise ambiguous cases.
- **Backslash substitution** replaces a backslash followed by a letter with another character. For example, "**\n**" is replaced with a newline.

From Tcl 8.5 onwards, any word may be prefixed by "**{*}**" to cause that word to be split apart into its constituent sub-words for the purposes of building the command invocation (similar to the "**,@**" sequence of Lisp's quasiquote feature).

As a consequence of these rules, the result of any command may be used as an argument to any other command. Also, there is no operator or command for string concatenation, as the language concatenates directly. Note that, unlike in Unix command shells, Tcl does not reparse any string unless explicitly directed to do so, which makes interactive use more cumbersome but scripted use more predictable (e.g. the presence of spaces in filenames does not cause difficulties).

The single equality sign (=) for example is not used at all, and the double equality sign (==) is the test for equality, and even then only in expression contexts such as the expr command or the first argument to if. (Both of those commands are just part of the standard library; they have no particularly special place in the library and can be replaced if so desired.)

The majority of Tcl commands, especially in the standard library, are variadic, and the proc (the constructor for scripted command procedures) allows one to define default values for unspecified arguments and a catch-all argument to allow the code to process arbitrary numbers of arguments.

Tcl is not statically typed: each variable may contain integers, floats, strings, lists, command names, dictionaries, or any other value; values are reinterpreted (subject to syntactic constraints) as other types on demand. However, values are immutable and operations that appear to change them actually just return a new value instead.

Experiment-7

Implement and simulate various types of routing algorithm.

THEORY:

Routing is the act of moving information across an inter-network from a source to a destination. Along the way, at least one intermediate node typically is encountered. It’s also referred to as the process of choosing a path over which to send the packets. Routing is often contrasted with bridging, which might seem to accomplish precisely the same thing to the casual observer. Routing protocols use metrics to evaluate what path will be the best for a packet to travel. A metric is a standard of measurement; such as path bandwidth, reliability, delay, current load on that path etc; that is used by routing algorithms to determine the optimal path to a destination. To aid the process of path determination, routing algorithms initialize and maintain routing tables, which contain route information. Route information varies depending on the routing algorithm used.

Routing algorithm:

For a packet to travel from source to destination it has to pass through multiple paths or sometimes a single path. So when a packet finds multiple paths to reach the destination, it has no judging methods available to find a right path. A router with the help of certain algorithms calculates the best path for the packet to reach the destination. These algorithms are called routing algorithms. This is the way in which a router finds the correct entry in its routing table. There are several algorithms available to find this best path but here I am going to discuss only the two basic types of algorithms.

Two basic routing algorithms are,

1. Distance-vector algorithm.
2. Link state routing algorithm.

1. Distance vector algorithm:

As from the name suggests it uses distance and direction to find the best path to reach the destination. The distance here is the number of hops a packet crosses to reach the destination. Each hop refers to a router across the path. The word vector refers to the direction of the packet to reach the destination. It has lesser convergence time and knowledge about the whole network when compared to link state routing algorithm. Working of this distance vector algorithm can be explained in three steps. The steps are as follows,

Step 1 : In this algorithm, the information about every router connected directly and routing updates will be gathered by every single router. This information about the whole network will be sent periodically to all the neighboring routers connected to it. In this way every router updates the information in its routing table.

Step 2 : All the information collected by a single router about the whole network will be sent only to its neighbors and not to all other routers in the routing table. If there is any change in the hopcount or disabled paths it will updated only to its neighbors which in turn after a period passes to its neighbors.

Step 3 : The above explained sharing of information will take place in a period of 30 seconds. If there is a change in the network like if a network fails or additionally a router is added to the network, the changed information will be updated only after that time period.

2. Link state routing algorithm:

This is the most popular routing algorithm used in the real time networks. It uses three tables for the calculation of the routing table entries. It is also called as 'Shortest path first algorithm'. It has several advantages over distance vector algorithm. Some of them include its faster convergence time, ability to handle very large networks, reliable path prediction. It uses link state advertisements to find the information about the router. Here in steps working of link state algorithm can be analyzed. The steps are,

Step 1: As from its name 'Shortest path first algorithm' it uses several calculations to find the shortest path to reach the destination. This algorithm uses link state packets or advertisements to collect the information about the neighboring routers. Only links that are connected directly are considered as neighbors. In contrast to distance vector it sends info only about neighbors.

Step 2: In this algorithm instead of sending the routing table info only to the neighbors it sends to all the routers in the network. In this algorithm totally three tables are maintained. One is for collecting info about neighbors, one has info about the entire topology, final one is the actual routing table.

Step 3: In this algorithm there is no periodic updates involved. A router in the network will send updates to all the routers and only if there is a change in the network. That is why it is

called as event triggered updates. This event triggered updates will help the router to find its path immediately without any errors.

EXPERIMENT NO. 8

Study of MAC Protocols like Aloha, CSMA, CSMA/CD and CSMA/CA .

Multiple Access Protocols in Computer Network

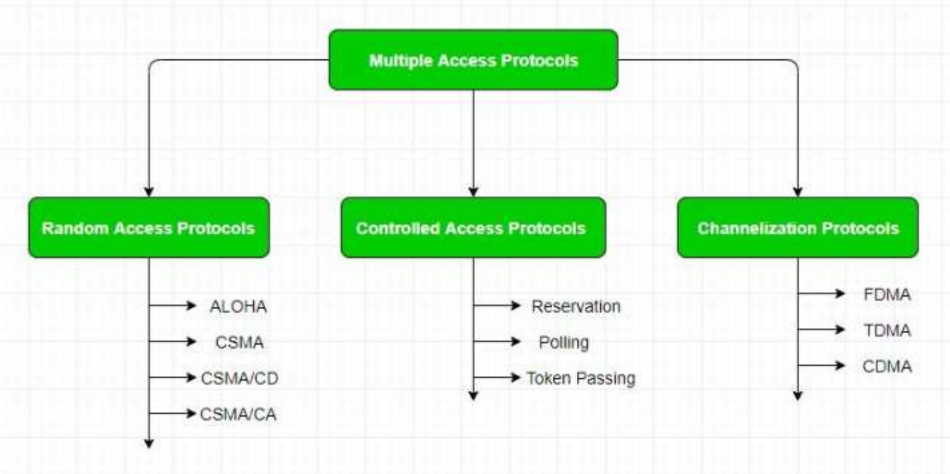
The Data Link Layer is responsible for transmission of data between two nodes. Its main functions are-

- Data Link Control
- Multiple Access Control

Data Link Control
Multiple Access Resolution

Data Link control –
The data link control is responsible for reliable transmission of message over transmission channel by using techniques like framing, error control and flow control.

Multiple Access Control –
If there is a dedicated link between the sender and the receiver then data link control layer is sufficient, however if there is no dedicated link present then multiple stations can access the channel simultaneously. Hence multiple access protocols are required to decrease collision and avoid crosstalk. For example, in a classroom full of students, when a teacher asks a question and all the students (or stations) start answering simultaneously (send data at same time) then a lot of chaos is created(data overlap or data lost) then it is the job of the teacher (multiple access protocols) to manage the students and make them answer one at a time.



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

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

The Random access protocols are further subdivided as:

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

- **Pure Aloha:**

When a station sends data it waits for an acknowledgement. If the acknowledgement doesn't come within the allotted time then the station waits for a random amount of time called back-off time (Tb) and re-sends the data. Since different stations wait for different amount of time, the probability of further collision decreases.

Vulnerable Time = 2* Frame transmission time

Throughput = G exp{-2*G}

Maximum throughput = 0.184 for G=0.5

- **Slotted Aloha:**

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

Vulnerable Time = Frame transmission time

Throughput = G exp{-*G}

Maximum throughput = 0.368 for G=1

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

CSMA access modes-

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

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

(d) **CSMA/CA** – Carrier sense multiple access with collision avoidance. The process of collisions detection involves sender receiving acknowledgement signals. If there is just one signal(its own) then the data is successfully sent but if there are two signals(its own and the one with which it has collided) then it means a collision has occurred. To distinguish between these two cases, collision must have a lot of impact on received signal. However it is not so in wired networks, so CSMA/CA is used in this case.

CSMA/CA avoids collision by:

1. **Interframe space** – Station waits for medium to become idle and if found idle it does not immediately send data (to avoid collision due to propagation delay) rather it waits for a period of time called Interframe space or IFS. After this time it again checks the medium for being idle. The IFS duration depends on the priority of station.

2. **Contention Window** – It is the amount of time divided into slots. If the sender is ready to send data, it chooses a random number of slots as wait time which doubles every time medium is not found idle. If the medium is found busy it does not restart the entire process, rather it restarts the timer when the channel is found idle again.

3. **Acknowledgement** – The sender re-transmits the data if acknowledgement is not received before time-out.

2. Controlled Access:

In this, the data is sent by that station which is approved by all other stations.

3. Channelization:

In this, the available bandwidth of the link is shared in time, frequency and code to multiple stations to access channel simultaneously.

- **Frequency Division Multiple Access (FDMA)** – The available bandwidth is divided into equal bands so that each station can be allocated its own band. Guard bands are also added so that no two bands overlap to avoid crosstalk and noise.

- **Time Division Multiple Access (TDMA)** – In this, the bandwidth is shared between multiple stations. To avoid collision time is divided into slots and stations are allotted these slots to transmit data. However there is a overhead of synchronization as each station needs to know its time slot. This is resolved by adding synchronization bits to each slot. Another issue with TDMA is propagation delay which is resolved by addition of guard bands.

- **Code Division Multiple Access (CDMA)** – One channel carries all transmissions simultaneously. There is neither division of bandwidth nor division of time. For example, if there are many people in a room all speaking at the same time, then also perfect reception of data is possible if only two person speak the same language. Similarly, data from different stations can be transmitted simultaneously in different code languages.
- **Orthogonal Frequency Division Multiple Access (OFDMA)** – In OFDMA the available bandwidth is divided into small subcarriers in order to increase the overall performance, Now the data is transmitted through these small subcarriers. it is widely used in the 5G technology.

Advantages:

- Increase in efficiency
- High data rates
- Good for multimedia traffic

Disadvantages:

- Complex to implement
- High peak to power ratio

Spatial Division Multiple Access (SDMA) – SDMA uses multiple antennas at the transmitter and receiver to separate the signals of multiple users that are located in different spatial directions. This technique is commonly used in MIMO (Multiple-Input, Multiple-Output) wireless communication systems.

Advantages :

- Frequency band uses effectively
- The overall signal quality will be improved

- The overall data rate will be increased

Disadvantages :

- It is complex to implement
- It require the accurate information about the channel

EXPERIMENT NO. 9

Study of Application layer protocols-DNS, HTTP, HTTPS, FTP and TelNet.

Application layer protocols:

Protocols in each layer of the network model provide a mechanism for devices to identify and connect. They also contain formatting rules specifying how data is packaged when the messages are sent and received.

There are several protocols in the application layer used for different services like email services, file transfers, etc. We will look at each one of them one by one.

1. DNS

A service that is used to translate domain names (google.com) to their corresponding IP addresses (8.8.8.8).

DNS stands for "domain name system". It is used for an effective translation of internet domain names into internet protocol addresses. What does this mean ? As humans, we work with a name to identify a particular website. However, that is not how computer networks understand. For viable communication between humans and systems, we need DNS.

For example, the public IP address 1.1.1.1 is used through which the computer locates our desired website i.e. cloudflare.com. An IP address is a 32-bit number similar in structure to 227.82.157.177.

The following are some characteristics of DNS :

The port number used is number 53.

The domain name is usually contained in a URL.

The domain name system follows a hierarchy which is an inverted tree-like structure to manage its distributed database system.

Most activities on the web rely on DNS to quickly make a connection between our computer & remote hosts of our desired location on the internet. The DNS service can be mapped to a phone book service where we receive the phone number using the name of the person we are looking to communicate with.

To understand the simple working of the DNS service, look at the image below:

TELNET

TELNET provides communication facilities between two hosts using the CLI.

TELNET is used for communication through the command line interface between remote device(s) or server(s). It stands for TELetype NETwork & configures elements of networking hardware.

Some characteristic features of TELNET are :

It is a client-server type of protocol.

It is a bidirectional and interactive communication feature for terminals and terminal-oriented processes.

Information is distributed over an 8-bit byte-oriented data connection.

On local machines, it is implemented as a program telnet. On remote machines, it works as the daemon in .telnet. A “daemon” is synonymous with a server or agent.

The port number used is number 23.

The two hosts can communicate over the TELNET user interface through two means:
line-by-line or character-by-character basis.

FTP:

It models a protocol to download, upload, and transfer files between two devices over the internet.

FTP stands for "File Transfer Protocol" and connects two computer systems to transfer files over a network. Users need to grant access using FTP to receive and send files. Transferring files is a straightforward mechanism, so why do we need FTP ? Because it overcomes these problems between two systems :

Different file conventions.

Different ways to represent text and data in the files.

Different directory structures.

Data Transfer :
Data connections made during data transfer processes that use complex rules as data types can vary greatly.

Control Connection :
It uses relatively simple rules for communication and is used to transfer a line of command or response at a time.

These are the common characteristics of FTP protocol :

Users require an internet connection to accomplish FTP transfers.

It promotes remote transfers.

The Port number for FTP is 20 for data and 21 for control.

The basic client model of FTP contains three components:

the user interface, control process, and data transfer process whereas the server model contains two units : the server control process and the server data transfer process

FTP is one of the fastest ways to transfer files, is efficient, and needs a username & password to access the server which makes it secure. However, it is not compatible with every system and doesn't allow running concurrent transfers to multiple receivers.

4. TFTP

A concise version of FTP, it provides a lightweight file transfer mechanism.

A simplified version of FTP, Trivial File Transfer Protocol (TFTP) is mainly used for reading and writing files to or from a remote server. It also facilitates file transfer, however, with no user authentication. Major characteristics of TFTP are :

The port number used is number 69.

It has limited features and provides no security during the transfer of files.

It is a lightweight file transfer mechanism.

It is often used on private local networks where adapting FTP can be expensive in its implementation or cost. It comes in handy where there are no hard disk drives or storage devices as the implementation is easy using a small amount of memory.

The five types of messages used in the TFTP protocol are :

RRQ :
Request to read a file

WRQ :
Request to write to a file

DATA :
Contains a block of file data

ACK :
Used by the peer to acknowledge each block of DATA

ERROR :
Used by the peer to indicate erroneous operations

5. NFS

It provides a model to share files remotely between servers over a network.

The **Network File System** (NFS) is a distributed file system protocol that is portable across different machines, operating systems, network architectures, and transport protocols.

The protocol mounts a file system present in a network & enables interactions with it as though that system is mounted locally. Users can use CLI commands to create, remove, read, write & perform other functions on the remote files accessed using NFS.

Some of the common characteristics to look at are :

The Port number used is 2049.

It is an open standard i.e. anyone can implement this protocol.

It has many versions, the most common of them being NFS v3.

5. SMTP

The SMTP protocol is necessary for the completion of email-related jobs.

Email services have been used extensively since their emergence in the late 1960s at the Massachusetts Institute of Technology when a message was sent from one device to another using ARPANET.

Hence, it becomes crucial to understand SMTP. It stands for Simple Mail Transfer Protocol and assists in sending mail over the internet.

The SMTP protocol uses two basic models to work efficiently :

End-to-end Method :
It helps in communicating with email servers between different organizations.

Store-and-forward Method :
It helps in communicating with email servers within the same organization.

Let us now look at some characteristics of this protocol :

The port number is number 25.

- It uses email addresses as a basis to function and send messages to devices.
- Like email, the SMTP program is also of a client-server architecture.

EXPERIMENT NO. 10

Configure 802.11 WLAN.

Wi-Fi enabled devices are connected wirelessly and can connect to the Internet via a wireless access point. Wi-Fi can function in geographical location and can be used where wiring and cable connection is not feasible.

In this experiment we will learn the different standards and the simulation of Wi-Fi network. It also explains the concept of hidden node and exposed node problem and solve these issues.

Wi-Fi Networks

Wi-Fi (Wireless Fidelity) uses the IEEE 802.11 standard. Wi-Fi has some other extensions like 802.11a, 802.11b, and 802.11g. Wi-Fi technology operating at a frequency of 2.4 GHz and uses radio communication. [v]

IEEE 802.11 Standards

Following are the different standards for Wi-Fi

- 802.11 is the wireless local area networks (WLANs) standard. Supports 1- 2 Mbps.
- 802.11a is a high speed WLANs standard for 5 GHz band. It uses an orthogonal frequency division multiplexing (OFDM) encoding scheme.
- 802.11b is a wireless standard for 2.4 GHz band. It supports 11 Mbps. It uses only DSSS (Direct Sequence Spread Spectrum).
- 802.11d is a international roaming. This automatically configures devices to meet local radio frequency (RF) regulations.
- 802.11e address the quality of service (QoS) requirements for all IEEE wireless radio interfaces.
- 802.11f defines inter-access point communications for multiple vendor-distributed WLANs.
- 802.11g establishes an additional modulation technique for 2.4 GHz band. This supports speeds up to 54 Mbps.
- 802.11h supports the spectrum management of the 5 GHz band.
- 802.11i define the current security weaknesses for both encryption and authentication protocols.

Parameter	802.11a	802.11b	802.11g
Standard approved	Sept 1999	Sept 1999	June 2003
Available bandwidth	300MHz	83.5MHz	83.5MHz
No. of overlapping channel	4	3	3
Frequency	5GHz	2.4GHz	2.4GHz

Parameter	802.11a	802.11b	802.11g
Typical Data Rate	23 Mbit/s	4.5 Mbit/s	19 Mbit/s
Maximum Data Rate	54 Mbit/s	11 Mbit/s	54 Mbit/s
Range	115 feet	115 feet	125 feet
Compatibility	None	None	backward compatible with b
Advantages	fast maximum speed, regulated frequencies prevent signal interference from other devices	lowest cost, signal range is good and not easily obstructed	fast maximum speed, signal range is good and not easily obstructed
Limitations	highest cost, shorter	slowest maximum speed,	costs more than 802.11b,

Parameter	802.11a	802.11b	802.11g
	range signal that is more easily obstructed	home appliances may interfere on the unregulated frequency band	appliances may interfere on the unregulated signal frequency

Hardware Requirements for Wi-Fi

The following hardware devices are required for connecting the Wi-Fi Network.

Access Point

Access Point (AP) acts as a bridge between the wired network and wireless devices. It allows multiple devices to connect through it for accessing the network. An AP can also act as a router through which the data transmission can be possible from one AP to another.

Wireless Network Card

A wireless network card (WNC) is required on each device on a wireless network. A desktop computer would need an internal card, which will usually have a small antenna or an external antenna on it. These antennas are optional on most equipment and they help to increase the signal on the card.

Transmitter

Transmitter is used for emitting the wireless signals and it also receive the connection requests where a wireless client will send the requests and receives the replies from the transmitter. In this case, the transmitter is the wireless router.

How to connect to the Wi-Fi Networks?

Wi-Fi Network is easy to connect. Suppose, we will think about our laptop with any operating systems, then we can easily connect to a Wi-Fi network for accessing or we can share different files on a network.

Once we have acquired the necessary wireless networking hardware then, connect it all together to form a network and allow each device to communicate. The instructions below will act as basic guidelines of what needs to be done.

The distance between each computer should be below 100 meters

Each computer should be on the same floor

Plug the AP into the power outlet and existing Ethernet jack on the network

Configure the access point (usually through a web browser)

Configure the client computers with the appropriate network settings required to be able to communicate with the AP.

Advantages of Wi-Fi:

Following are the different benefits of Wi-Fi Networks

- In wireless ad-hoc network mode, devices like consumer electronics and gaming applications can directly connect and exchange data with each other.
- Digital images can be transferred wirelessly from cameras and other devices.
- All connected devices within the range have access to Internet and inter-networking.
- Wi-Fi enables wireless voice-applications (Vo WLAN or WVOIP).
- Wi-Fi provides a secure computer networking gateway, firewall, DHCP server and an intrusion detection system along with other features.
- Cost of cabling and network deployment of Local Area Networks is significantly reduced.
- Can be used at placed where wiring and cable lay-out is not feasible
- Due to its cost effective nature, it can be used widely in different educational campuses and industries.
- Wi-Fi device can function in any type of geographical location.

Limitations

- Like any other types of technology, Wi-Fi has its set of drawbacks that are listed as follows:
- Global inconsistency of spectrum assignments and operational limitations.
- Overlapping of channels.
- Limited range of equivalent isotropically radiated power in some areas.
- Greater power consumption compared to lower bandwidth standards.
- Limited battery life due to range and reach requirements.
- Wi-Fi network range is also limited.

MAC Protocols

The 802.11 standards use a MAC layer known as CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance).

In CSMA/CA a Wireless node that wants to transmit & performs the following sequence:

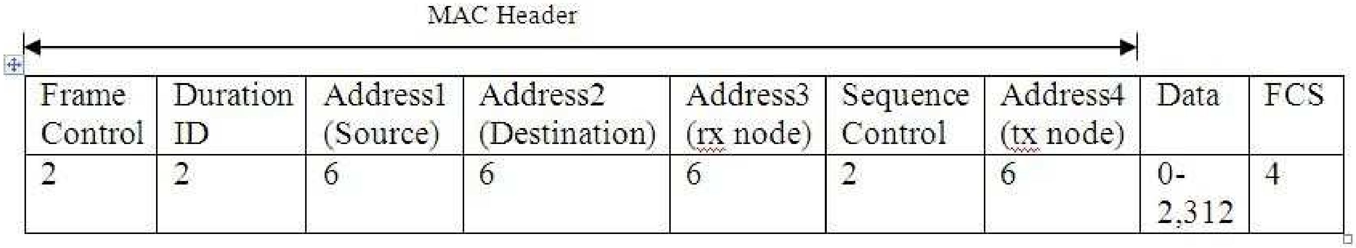
1. Listen on the desired channel.

2. If channel is idle (no active transmitters) it sends a packet.

3. If channel is busy then, the node waits until the transmission end then a contention period where minimum time a host must transmit before it can be sure that the no other host's packet has collided with its transmission.

4. If the channel is still idle at the end of the contention period, then the node transmits its packet otherwise it repeats the process defined in step-3 above until it gets a free channel.

:The MAC header format shown in the figure-01 below :



MAC Header

Frame Control	Duration ID	Address1 (Source)	Address2 (Destination)	Address3 (rx node)	Sequence Control	Address4 (tx node)	Data	FCS
2	2	6	6	6	2	6	0-2,312	4

Use of RTS/CTS to Exchange Data

Step 1:

At first the sender check whether the medium is idle or not, if so, after the Distributed Inter Frame Space (DIFS will check the status and sense before transmitting the data in the wireless medium) units of time, it will broadcasts a Request-to-Send (RTS) frame to the receiver address.

Step 2:

If the receiver is within the range, then it will wait for Short Inter Frame Space (SIFS is the small time interval between the data frame and its acknowledgment) unit of time, then only it will respond to the sender with a Clear-to-Send (CTS) frame.

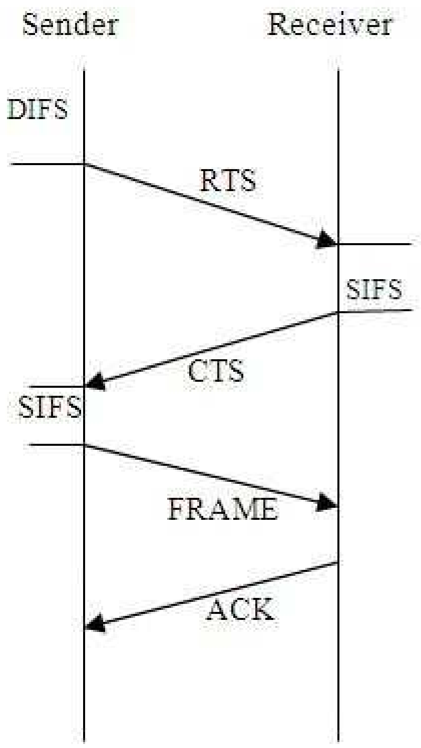
Step 3:

If the sender receive the CTS frame, then it will wait for another SIFS unit of time before sending the data frame to the receiver.

Step 4:

Finally, when the receiver will successfully receive the data frame, it will wait for SIFS unit of time and also send an Acknowledgement (ACK) message return to the sender.

Following figure-02 shows how data exchanges using RTC/CTS



Issues in Wi-Fi Networks

Wi-Fi suffers from two well known problems:

- Hidden Terminal Problem
- Exposed Terminal Problem

The Hidden Terminal Problem

The hidden node/ terminal problem found at a point to multipoint network and it is defined as being one in which three or more nodes are present. Let there are three nodes :node A, node B and node C.

A and C cannot hear each other.

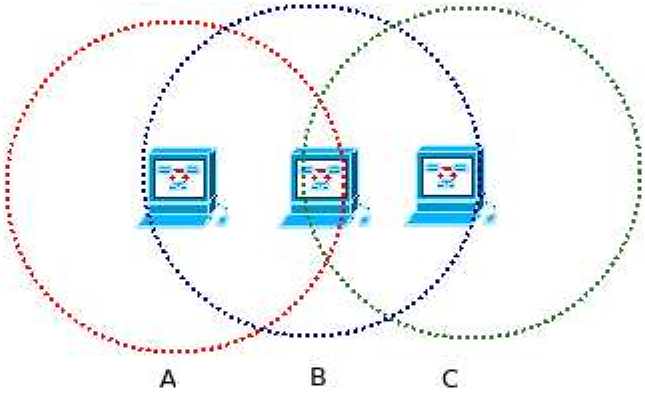
A sends to B, C cannot receive A.

C wants to send B, C senses a free medium.

Collision occurs at B.

A cannot receive the collision.

A is hidden for C.



Solution of Hidden Terminal Problem

The solution of hidden terminal problem is as follows.

When A wants to send a packet to B , A first sends a Request-to-send (RTS) to B.

On receiving RTS, B responds by sending Clear-to-Send (CTS).

When C overhears a CTS, it keeps quiet for the duration of the transfer.

Transfer duration is included in both RTS and CTS.

RTS and CTS are short frames, reduces collision chance.

The other methods that can be employed to solve hidden terminal problem are :

Increase transmitting power from the nodes.

- Use unidirectional antennas.

- Remove obstacles.

- Move the node.

- Use protocol enhancement software.

- Use antenna diversity.

Effect of Hidden Terminal Problem

- If one node hidden to another then the re-transmission will increase.
- It also increase the delay and decrease the throughput.

Exposed Terminal Problem

Suppose there are four nodes: node A, node B, node C and node D.

Here -

B can send to both A and C .

C can send to D, but not to A or B.

A and C cannot hear each other.

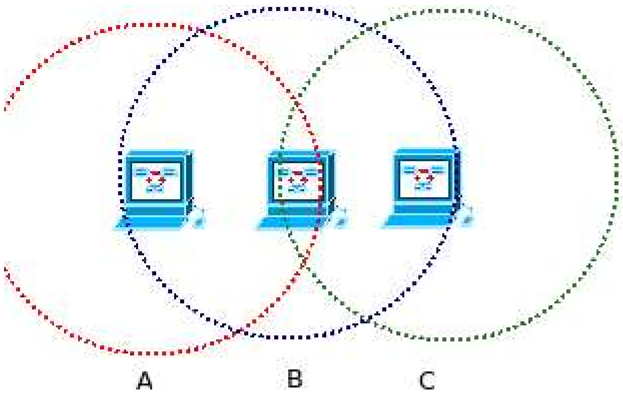
Now the Problem as follows :

- When B transmits to A, C detects the transmission using the carrier sense mechanism.

- So C defers transmitting to D.

- But C could have sent to D, so blocked unnecessarily.

The following figure-04 shows the Exposed Terminal Problem using node A,B,C and D



Solution to the Exposed Terminal Problem

Exposed terminal problems cannot be mitigated with RTS/CTS. This can be explained with the following scenario.

Suppose B sends RTS to A.

A sends CTS to B.

C hears RTS, but not CTS, assumes it is ok to send to D.

Simulating a Wi-Fi using Network Simulator 3

NS-3 is an open source network simulator, and supposed to be the future replacement of NS-2. We would be using NS-3 for this experiment. Providing a tutorial on NS-3 is, however, outside the scope of this experiment. The **Exercises** section presents two problems and their solutions using NS-3. We would use the existing codes and tweak it to get familiarized with the basic characteristics of Wi-Fi. Nevertheless, one could visit the NS-3website to learn more about this simulator.