NRI INSTITUTE OF INFORMATION SCIENCE & TECHNOLOGY BHOPAL

DEPARTMENT OF INFORMATION TECHNOLOGY

LAB MANUAL

COMPUTER NETWORK LAB

(IT-502)

INFORMATION TECHNOLOGY (IT)

EXPERIMENT NO. : 1

TYPES OF CONNECTORS

Network media is the actual path over which an electrical signal travels as it moves from one component to another. There are some common types of network media, including twisted-pair Cable, coaxial cable, fiber-optic cable, and wireless.

1. Twisted-Pair Cable

- Twisted-pair cable is a type of cabling that is used for telephone communications and most modern Ethernet networks. A pair of wires forms a circuit that can transmit data. The pairs are twisted to provide protection against crosstalk, the noise generated by adjacent pairs. When electrical current flows through a wire, it creates a small, circular magnetic field around the wire. When two wires in an electrical circuit are placed close together, their magnetic fields are the exact opposite of each other. Thus, the two magnetic fields cancel each other out. They also cancel out any outside magnetic fields.

- Two basic types of twisted-pair cable exist: unshielded twisted pair (UTP) and shielded twisted pair (STP).

a) UTP Cable

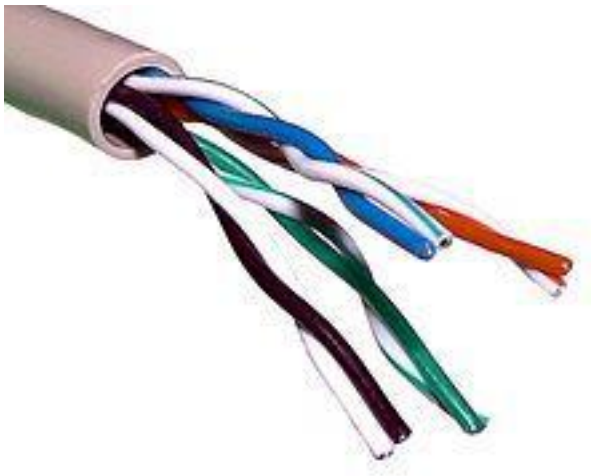
- UTP cable is a medium that is composed of pairs of wires UTP cable is used in a variety of networks. Each of the eight individual copper wires in UTP cable \is covered by an insulating material. In addition, the wires in each pair are twisted around each other.
- UTP cable must follow precise specifications governing how many twists or braids are permitted per meter (3.28 feet) of cable.
- UTP cable often is installed using a Registered Jack 45 (RJ-45) connector (see Figure 8-2). The RJ-45 is an eight-wire connector used commonly to connect computers onto a local-area network (LAN), especially Ethernets.
- When used as a networking medium, UTP cable has four pairs of either 22- or 24-gauge copper wire. UTP used as a networking medium has an impedance of 100 ohms; this differentiates it from other types of twisted-pair wiring such as that used for telephone wiring, which has impedance of 600 ohms.
- UTP cable offers many advantages. Because UTP has an external diameter of approximately 0.43 cm (0.17 inches), its small size can be advantageous during installation. UTP cable is easy to install and is less expensive than other types of networking media. In fact, UTP costs less per meter than any other type of LAN cabling.
- UTP cable is more prone to electrical noise and interference than other types of networking media, and the distance between signal boosts is shorter for UTP than it is for coaxial and fiber-optic cables.

The following summarizes the features of UTP cable:

- Speed and throughput—10 to 1000 Mbps
- Average cost per node—Least expensive
- Media and connector size—Small
- Maximum cable length—100 m (short)

Commonly used types of UTP cabling are as follows:

- Category 1—Used for telephone communications. Not suitable for transmitting data.
- Category 2—Capable of transmitting data at speeds up to 4 megabits per second (Mbps).
- Category 3—Used in 10BASE-T networks. Can transmit data at speeds up to 10 Mbps.
- Category 4—Used in Token Ring networks. Can transmit data at speeds up to 16 Mbps.
- Category 5—Can transmit data at speeds up to 100 Mbps.
- Category 5e —Used in networks running at speeds up to 1000 Mbps (1 gigabit per second [Gbps]).
- Category 6—Typically, Category 6 cable consists of four pairs of 24 American Wire Gauge (AWG) copper wires. Category 6 cable is currently the fastest standard for UTP.



Unshielded twisted pair cable

b) <u>Shielded Twisted-Pair Cable</u>

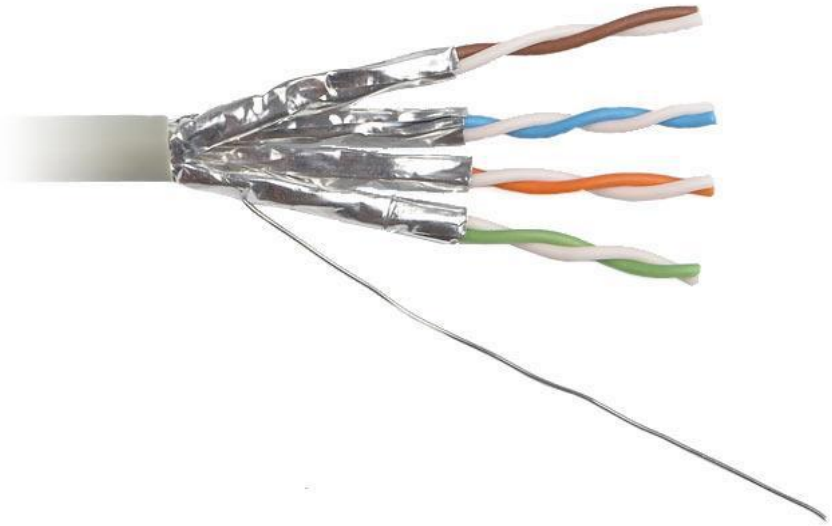
- Shielded twisted-pair (STP) cable combines the techniques of shielding, cancellation, and wire twisting. Each pair of wires is wrapped in a metallic foil .The four pairs of wires then are wrapped in an overall metallic braid or foil, usually 150-ohm cable. STP usually is installed with STP data connector, which is created especially for the STP cable. However, STP cabling also can use the same RJ connectors that UTP uses.
- Although STP prevents interference better than UTP, it is more expensive and difficult to install. In addition, the metallic shielding must be grounded at both ends. If it is improperly grounded, the shield acts like an antenna and picks up unwanted signals. Because of its cost

and difficulty with termination, STP is rarely used in Ethernet networks. STP is primarily used in Europe.
The following summarizes the features of STP cable:

- Speed and throughput—10 to 100 Mbps
- Average cost per node—Moderately expensive
- Media and connector size—Medium to large
- Maximum cable length—100 m (short)

When comparing UTP and STP, keep the following points in mind:

- The speed of both types of cable is usually satisfactory for local-area distances.
- These are the least-expensive media for data communication. UTP is less expensive than STP.
- Because most buildings are already wired with UTP, many transmission standards are adapted to use it, to avoid costly rewiring with an alternative cable type.



c) <u>Coaxial Cable</u>

- Coaxial cable consists of a hollow outer cylindrical conductor that surrounds a single inner wire made of two conducting elements. One of these elements, located in the center of the cable, is a copper conductor. Surrounding the copper conductor is a layer of flexible insulation. Over this insulating material is a woven copper braid or metallic foil that acts both as the second wire in the circuit and as a shield for the inner conductor. This second layer, or shield, can help reduce the amount of outside interference. Covering this shield is the cable jacket.

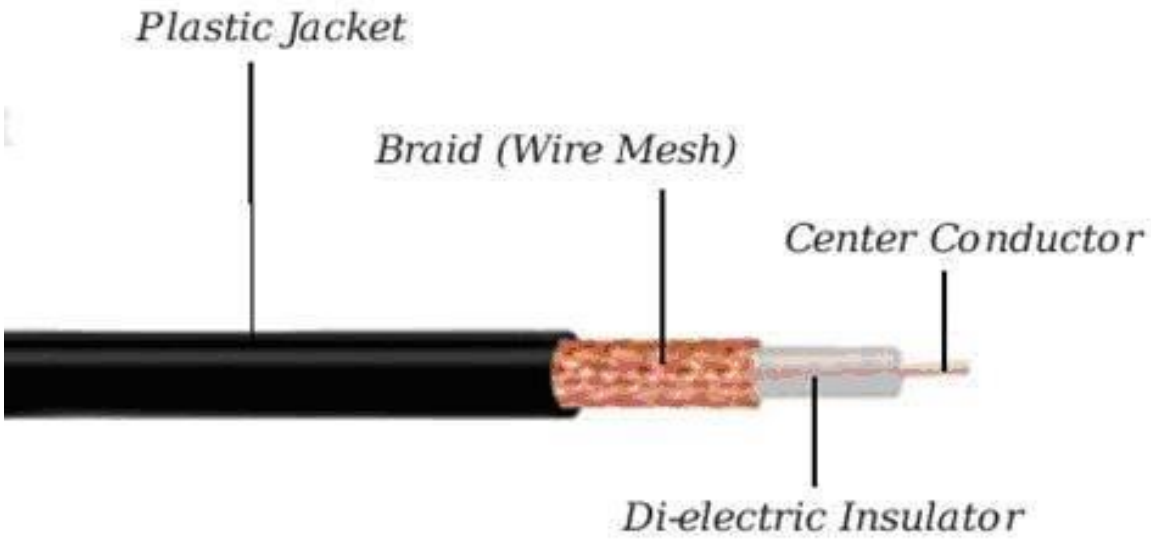
- Coaxial cable supports 10 to 100 Mbps and is relatively inexpensive, although it is more costly than UTP on a per-unit length. However, coaxial cable can be cheaper for a physical bus topology because less cable will be needed. Coaxial cable can be cabled over longer distances than twisted-pair cable. For example, Ethernet can run approximately 100 meters (328 feet) using twisted-pair cabling. Using coaxial cable increases this distance to 500m (1640.4 feet).
- For LANs, coaxial cable offers several advantages. It can be run with fewer boosts from repeaters for longer distances between network nodes than either STP or UTP cable. Repeaters regenerate the signals in a network so that they can cover greater distances. Coaxial cable is less expensive than fiber-optic cable, and the technology is well known; it has been used for many years for all types of data communication.
- When working with cable, you need to consider its size. As the thickness, or diameter, of the cable increases, so does the difficulty in working with it. Many times cable must be pulled through existing conduits and troughs that are limited in size. Coaxial cable comes in a variety of sizes. The largest diameter (1 centimeter [cm]) was specified for use as Ethernet backbone cable because historically it had greater transmission length and noise-rejection characteristics. In the past, coaxial cable with an outside diameter of only 0.35 cm (sometimes referred to as Thinnet) was used in Ethernet networks.

BNC T-connectors are female devices for connecting two cables to a network interface card (NIC). A BNC barrel connector facilitates connecting two cables together.

The following summarizes the features of coaxial cables:

- Speed and throughput—10 to 100 Mbps
- Average cost per node—Inexpensive
- Media and connector size—Medium
- Maximum cable length—500 m (medium)

Components of a Co-Axial Cable



2. Wireless Communication

- Wireless communication uses radio frequencies (RF) or infrared (IR) waves to transmit data between devices on a LAN. For wireless LANs, a key component is the wireless hub, or access point, used for signal distribution (see Figure 8-8).
- To receive the signals from the access point, a PC or laptop must install a wireless adapter card (wireless NIC). Wireless signals are electromagnetic waves that can travel through the vacuum of outer space and through a medium such as air. Therefore, no physical medium is necessary for wireless signals, making them a very versatile way to build a network. Wireless signals use portions of the RF spectrum to transmit voice, video, and data. Wireless frequencies range from 3 kilohertz (kHz) to 300 gigahertz (GHz). The data-transmission rates range from 9 kilobits per second (kbps) to as high as 54 Mbps.
- The primary difference between electromagnetic waves is their frequency. Low-frequency electromagnetic waves have a long wavelength (the distance from one peak to the next on the sine wave), while high-frequency electromagnetic waves have a short wavelength.

Some common applications of wireless data communication include the following:

- Accessing the Internet using a cellular phone
- Establishing a home or business Internet connection over satellite
- Beaming data between two hand-held computing devices
- Using a wireless keyboard and mouse for the PC

Another common application of wireless data communication is the wireless LAN (WLAN), which is built in accordance with Institute of Electrical and Electronics Engineers (IEEE) 802.11 standards. WLANs typically use radio waves (for example, 902 megahertz [MHz]), microwaves (for example, 2.4 GHz), and IR waves (for example, 820 nanometers [nm]) for communication. Wireless technologies are a crucial part of the today's networking.

EXPERIMENT NO. : 2

How to Configure a Wireless Router as Bridge

A wireless router is typically used to send out a wireless signal which will let other devices access the Internet. This is not a wireless router's only use though. It can also be modified for use as a bridge. A bridge is a device that brings together two or more different networks. When using a bridge, information can be passed between the two different networks.

Things Required:

- Wireless Router
- DSL Modem
- Ethernet Cables

1. While the power is off on the DSL modem and wireless router, plug an Ethernet cable into the DSL modem and connect the other end of the cable into the "Internet" port of the wireless router. Take a separate Ethernet cable and plug it into port "1" of the router, then plug the other end of the cable into the computer.

2. Power on both the wireless router and DSL modem.

3. On the computer connected to the wireless router, log on to a web browser. Type the IP address of the wireless router into the In the URL bar. This can be found either on the bottom side of the router or in the packaging the router came in. Hit "Enter" and the router setup site will load.

4. Type in "admin" for both the username and password (this is the default settings). Once that has been completed, the configuration site will load. Navigate to where the IP address is listed (this will vary slightly from model to model). Change a few of the numbers of the IP address to any other numbers so it is no longer under the default settings.

5. Disable both the "DNS" and "DNHP" servers as well as the firewall settings. The main server will provide all these, and if left on, the DNS, DNHP and firewall settings of the wireless router would conflict with the main server's settings.

6. Delete all the information listed in "Port Forwarding," then click "Save Changes." Once the changes have completely been saved, shut down the router.

7. Wait a few moments, and then power the router back on. It will now perform as the bridge to the two networks.

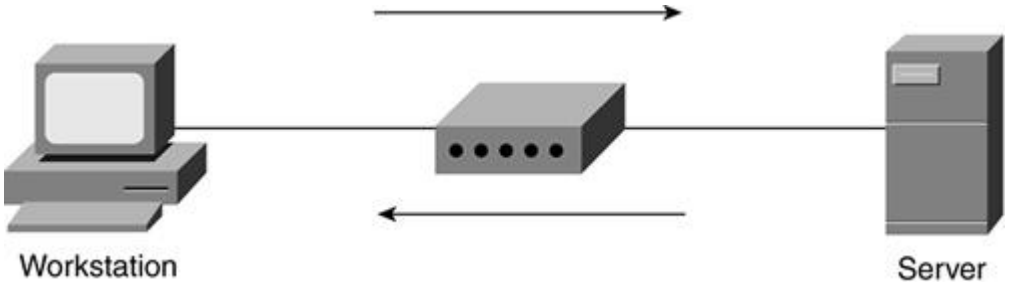
EXPERIMENT NO. : 3

Types of LAN Equipment

"Local-Area Networking Introduction," discussed the different types of hardware found in a LAN environment hubs, bridges, and switches and how each piece of hardware functions specifically in an Ethernet environment.

a) Repeaters (Layer 1 Devices)

A repeater is a network device used to regenerate or replicate a signal. Repeaters are used in transmission systems to regenerate analog or digital signals distorted by transmission loss. Repeaters are used in both local-and wide-area networking environments to extend the distance a signal can reach. In the LAN environment, you would use a repeater to extend the distance a data signal can travel on a cable



b) Hubs(Layer 1 Devices)

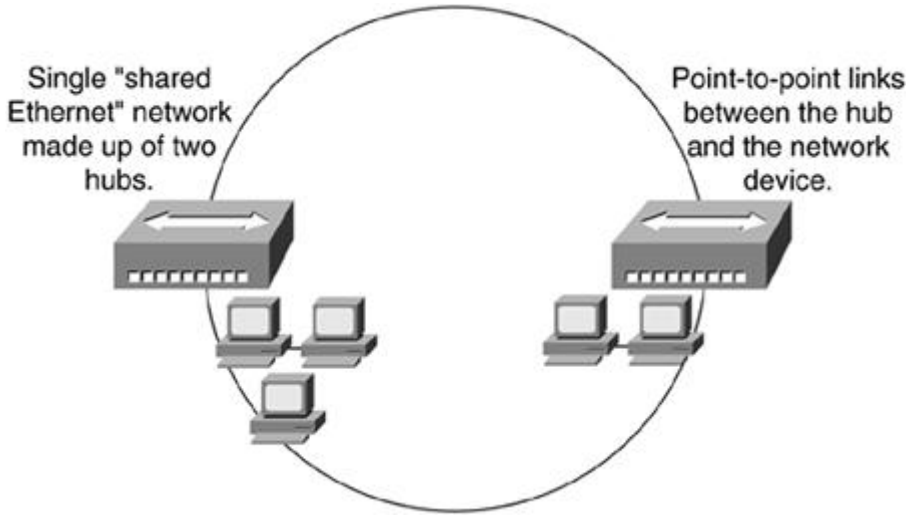
A hub is often used to connect small LAN segments in which the number of devices is generally 24 or fewer, and hubs are multiport repeaters. Hubs provide the signal amplification required to allow a segment to be extended a greater distance.

A hub takes an incoming signal on any one port and repeats it out all ports to enable users to share the Ethernet network resources.

Ethernet hubs create star topologies in 10-Mbps or 100-Mbps half-duplex Ethernet LANs.

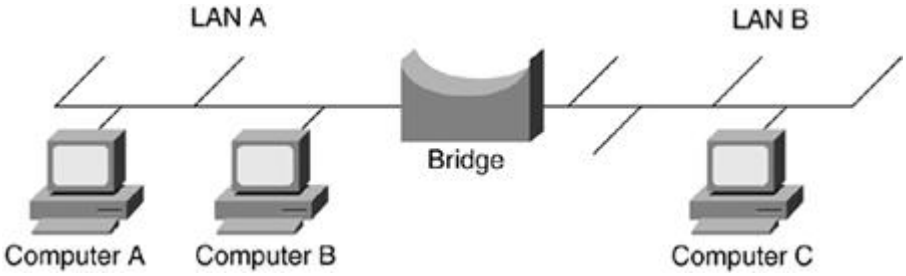
It is the hub that enables several point-to-point segments to be joined together into one single network.

A shared Ethernet LAN means that all members of the network are contending for transmission of data onto a single network.



c) Bridges (Layer 2 Devices)

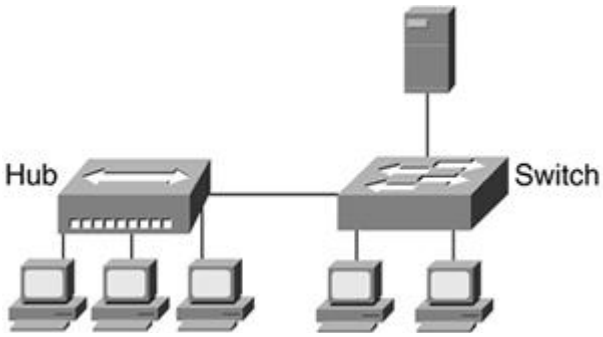
Repeaters and hubs have no intelligence; they just repeat whatever signal is received from one port out all ports without looking at what is being sent or received. Bridges add a level of intelligence to the network by using the MAC address to build a table of hosts, mapping these hosts to a network segment and containing traffic within these network segments. For example, Figure 5-6 illustrates a bridged network with two network segments.



d) Switches (Layer 2 Devices)

Switches sit in the same place in the network as hubs. Unlike hubs, however, switches examine each frame and process the frame accordingly instead of just repeating the signal to all ports. Switches map the MAC addresses of the nodes residing on each network segment and then allow only the necessary traffic to pass through the switch. A switch performs the same functions as a bridge; so when the switch receives a frame, it examines the destination and source MAC addresses and compares them to a table of network segments and addresses. If the segments are the same, the frame is dropped, or filtered; if the segments differ, the frame is forwarded to the proper segment.

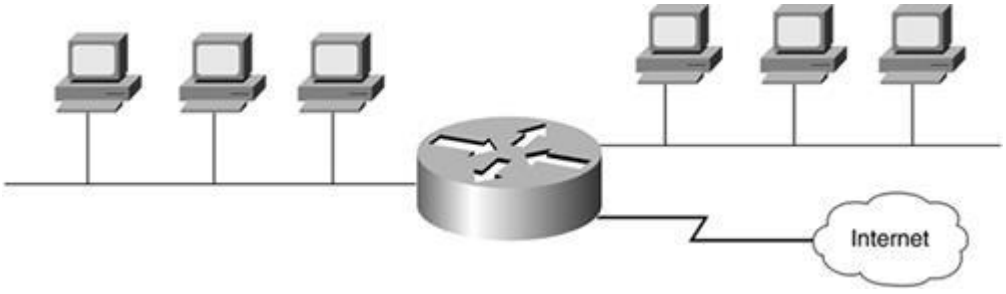
The filtering of frames and regeneration of forwarded frames enables switches to split a network into separate collision domains. Frame regeneration enables greater distances and more network devices, or nodes, to be used in the total network design, and lowers the overall collision rates. In switched networks, each segment is an independent collision domain, whereas in shared networks all nodes reside in one, big, shared collision domain.



e)Routers (Layer 3 Devices)

Routers are devices that forward data packets from one LAN or WAN to another. Based on routing tables and routing protocols, routers read the network address in the packet contained within each transmitted frame. Routers then select a sending method for the packet based on the most expedient route. This most expedient route is determined by factors such as traffic load, line quality, and available bandwidth. Routers work at Layer 3 (network) in the protocol stack, whereas bridges and switches work at Layer 2 (data link).

Routers segment LANs to balance traffic within workgroups and to filter traffic for security purposes and policy management.



EXPERIMENT NO. : 4

Standard Network topology

Network topology is the physical interconnections of the elements (links, nodes, etc.) of a computer network.A local area network (LAN) is one example of a network that exhibits both a physical topology and a logical topology. Any given node in the LAN has one or more links to one or more other nodes in the network and the mapping of these links and nodes in a graph results in a geometrical shape that may be used to describe the physical topology of the network. Likewise, the mapping of the data flows between the nodes in the network determines the logical topology of the network. The physical and logical topologies may or may not be identical in any particular

Basic topology types

- Bus topology
- Star topology
- Ring topology
- Tree topology

1. **Bus network topology:**

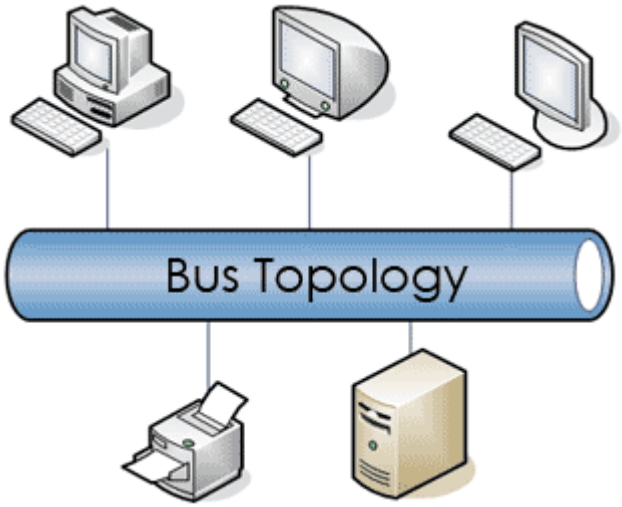
   In local area networks where bus technology is used, each machine is connected to a single cable. Each computer or server is connected to the single bus cable through some kind of connector. A terminator is required at each end of the bus cable to prevent the signal from bouncing back and forth on the bus cable.

   ### Linear bus

   The type of network topology in which all of the nodes of the network are connected to a common transmission medium which has exactly two endpoints (this is the 'bus', which is also commonly referred to as the backbone, or trunk)

   ### Distributed bus

   The type of network topology in which all of the nodes of the network are connected to a common transmission medium which has more than two endpoints that are created by adding branches to the main section of the transmission medium – the physical distributed bus topology functions in exactly the same fashion as the physical linear bus topology (i.e., all nodes share a common transmission medium).



2. **Star network topology**

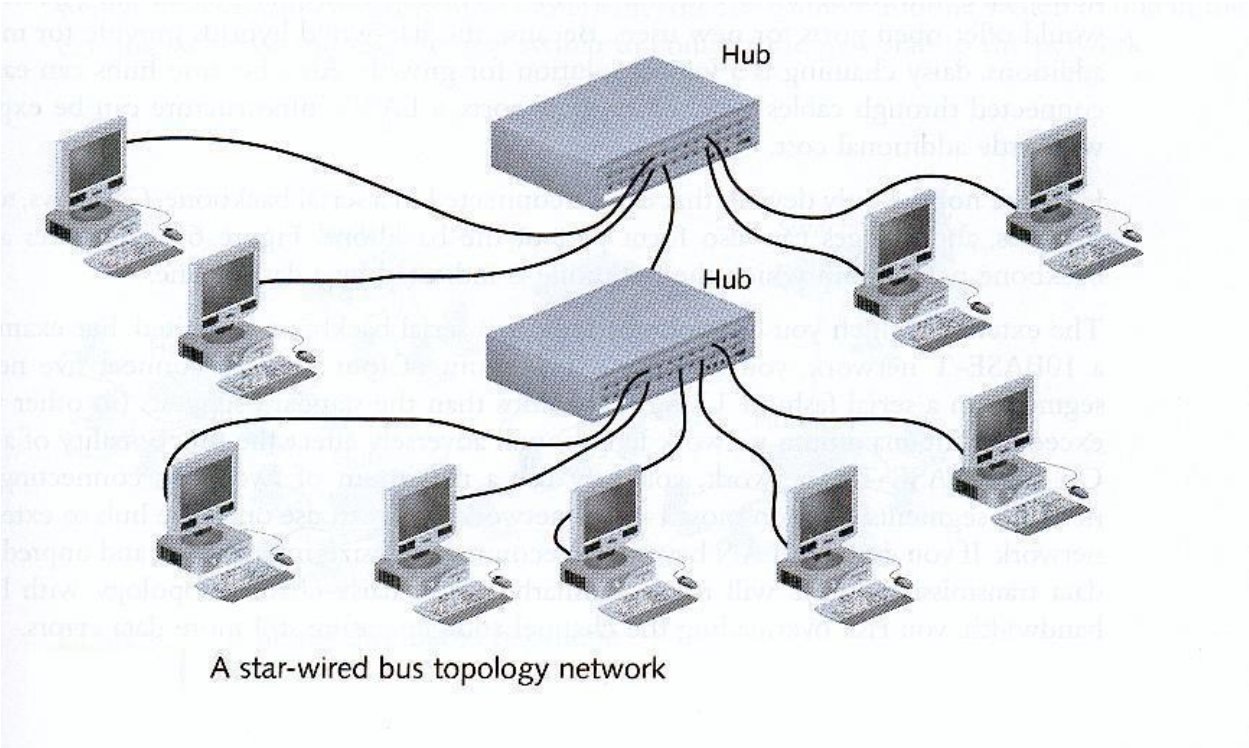
In local area networks where the star topology is used, each machine is connected to a central hub. In contrast to the bus topology, the star topology allows each machine on the network to have a point to point connection to the central hub. All of the traffic which transverses the network passes through the central hub. The hub acts as a signal booster or repeater which in turn allows the signal to travel greater distances.

Extended star

A type of network topology in which a network that is based upon the physical star topology has one or more repeaters between the central node (the 'hub' of the star) and the peripheral or 'spoke' nodes, the repeaters being used to extend the maximum transmission distance of the point-to-point links between the central node and the peripheral nodes beyond that which is supported by the transmitter power of the central node or beyond that which is supported by the standard upon which the physical layer of the physical star network is based.

Distributed Star

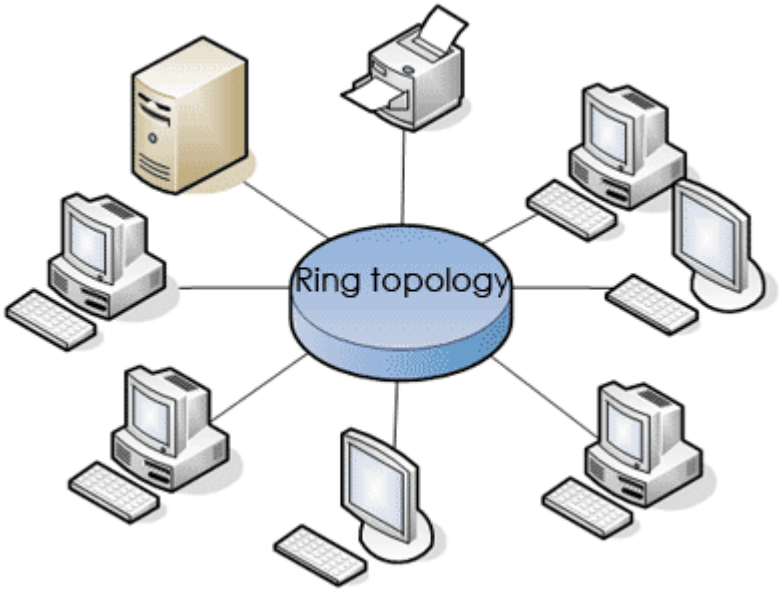
A type of network topology that is composed of individual networks that are based upon the physical star topology connected together in a linear fashion – i.e., 'daisy-chained' – with no central or top level connection point (e.g., two or more 'stacked' hubs, along with their associated star connected nodes or 'spokes').



A star-wired bus topology network

3. **Ring network topology**

In local area networks where the ring topology is used, each computer is connected to the network in a closed loop or ring. Each machine or computer has a unique address that is used for identification purposes. The signal passes through each machine or computer connected to the ring in one direction. Ring topologies typically utilize a token passing scheme, used to control access to the network. By utilizing this scheme, only one machine can transmit on the network at a time.



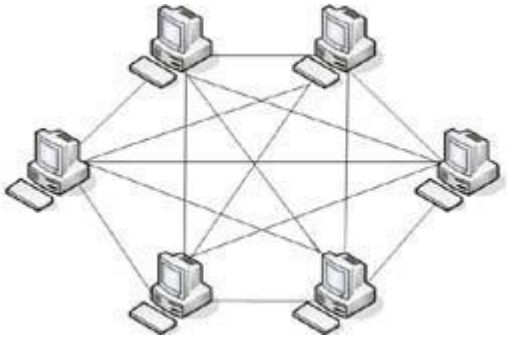
4. **Mesh network topology**

The value of fully meshed networks is proportional to the exponent of the number of subscribers, assuming that communicating groups of any two endpoints, up to and including all the endpoints, is approximated by Reed's Law.
Fully connected

The physical fully connected mesh topology is generally too costly and complex for practical networks, although the topology is used when there are only a small number of nodes to be interconnected.

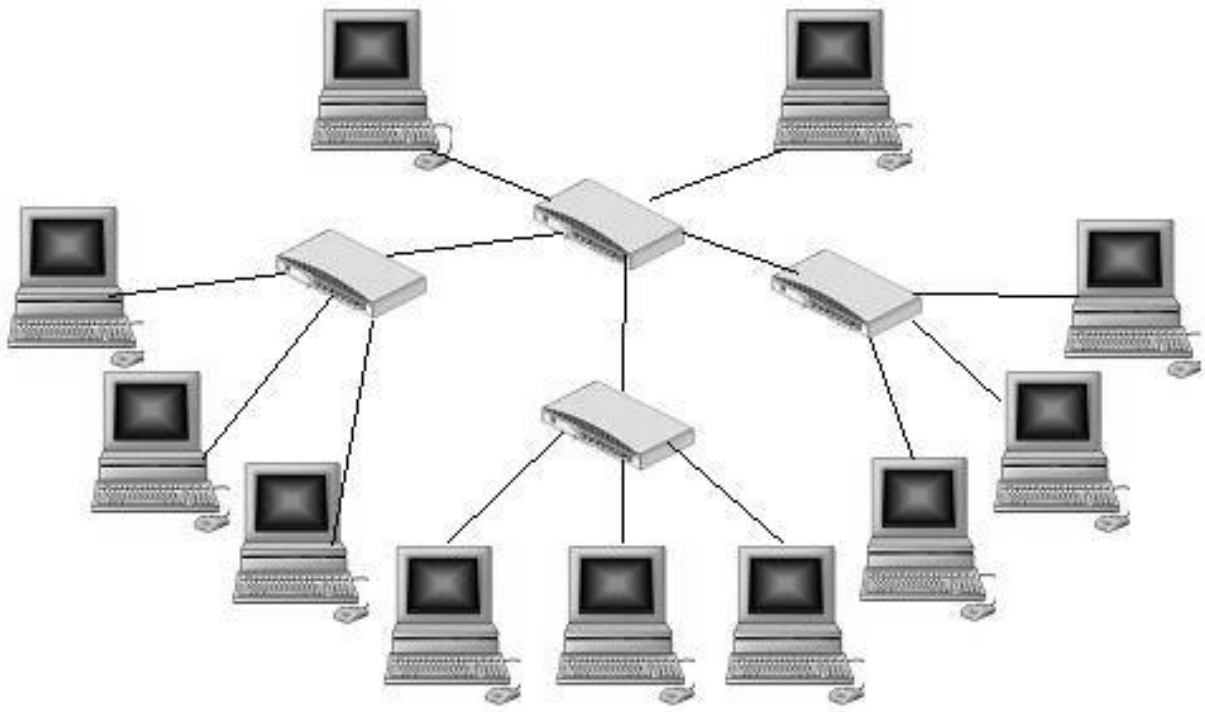
Partially connected

The type of network topology in which some of the nodes of the network are connected to more than one other node in the network with a point-to-point link – this makes it possible to take advantage of some of the redundancy that is provided by a physical fully connected mesh topology without the expense and complexity required for a connection between every node in the network.



5. **Tree network topology**

Also known as a **hierarchical network**. The type of network topology in which a central 'root' node (the top level of the hierarchy) is connected to one or more other nodes that are one level lower in the hierarchy (i.e., the second level) with a point-to-point link between each of the second level nodes and the top level central 'root' node, while each of the second level nodes that are connected to the top level central 'root' node will also have one or more other nodes that are one level lower in the hierarchy (i.e., the third level) connected to it, also with a point-to-point link, the top level central 'root' node being the only node that has no other node above it in the hierarchy (The hierarchy of the tree is symmetrical.) Each node in the network having a specific fixed number, of nodes connected to it at the next lower level in the hierarchy, the number, being referred to as the 'branching factor' of the hierarchical tree.



EXPERIMENT NO. : 5

Implement various types of error correcting techniques.

- *Error detection* is the detection of errors caused by noise or other impairments during transmission from the transmitter to the receiver.[1]
- *Error correction* is the detection of errors and reconstruction of the original, error-free data. Introduction

The general idea for achieving error detection and correction is to add some redundancy (i.e., some extra data) to a message, which receivers can use to check consistency of the delivered message, and to recover data determined to be erroneous. Error-detection and correction schemes can be either systematic or non-systematic. Good error control performance requires the scheme to be selected based on the characteristics of the communication channel. Common channel models include memory-less models where errors occur randomly and with a certain probability, and dynamic models where errors occur primarily in bursts.

Implementation

Error correction may generally be realized in two different ways:

- *Automatic repeat request (ARQ)* (sometimes also referred to as *backward error correction*): This is an error control technique whereby an error detection scheme is combined with requests for retransmission of erroneous data.
- *Forward error correction (FEC)*: The sender encodes the data using an *error-correcting code (ECC)* prior to transmission. The additional information (redundancy) added by the code is used by the receiver to recover the original data.

Error detection schemes

Error detection is most commonly realized using a suitable hash function (or checksum algorithm). A hash function adds a fixed-length *tag* to a message, which enables receivers to verify the delivered message by re-computing the tag and comparing it with the one provided.

There exists a vast variety of different hash function designs. However, some are of particularly widespread use because of either their simplicity or their suitability for detecting certain kinds of errors (e.g., the cyclic redundancy check's performance in detecting burst errors).

Random-error-correcting codes based on minimum distance coding can provide a suitable alternative to hash functions when a strict guarantee on the minimum number of errors to be detected is desired

- **Parity bits**

  A **parity bit** is a bit that is added to a group of source bits to ensure that the number of set bits (i.e., bits with value 1) in the outcome is even or odd. It is a very simple scheme that can be used to detect

single or any other odd number (i.e., three, five, etc.) of errors in the output. An even number of flipped bits will make the parity bit appear correct even though the data is erroneous.

- **Checksums**

  A **checksum** of a message is a modular arithmetic sum of message code words of a fixed word length (e.g., byte values). The sum may be negated by means of aones'-complement operation prior to transmission to detect errors resulting in all-zero messages.

  Checksum schemes include parity bits, check digits, and longitudinal redundancy checks.

- **Cyclic redundancy checks (CRCs)**

  A **cyclic redundancy check (CRC)** is a single-burst-error-detecting cyclic code and non-secure hash function designed to detect accidental changes to digital data in computer networks. It is characterized by specification of a so-called *generator polynomial*, which is used as the divisor in a polynomial long division over a finite field, taking the input data as the dividend, and where the remainder becomes the result.

  Even parity is a special case of a cyclic redundancy check, where the single-bit CRC is generated by the divisor $x + 1$.

- **Cryptographic hash functions**

  A **cryptographic hash function** can provide strong assurances about data integrity, provided that changes of the data are only accidental (i.e., due to transmission errors). Any modification to the data will likely be detected through a mismatching hash value. Furthermore, given some hash value, it is infeasible to find some input data (other than the one given) that will yield the same hash value. Message authentication codes,

Error-correcting codes

Any error-correcting code can be used for error detection. A code with *minimum Hamming distance*, *d*, can detect up to $d - 1$ errors in a code word. Using minimum-distance-based error-correcting codes for error detection can be suitable if a strict limit on the minimum number of errors to be detected is desired.

Codes with minimum Hamming distance $d = 2$ are degenerate cases of error-correcting codes, and can be used to detect single errors. The parity bit is an example of a single-error-detecting code.

Error correction

- **Automatic repeat request**

  Automatic Repeat request (ARQ) is an error control method for data transmission that makes use of error-detection codes, acknowledgment and/or negative acknowledgment messages, and timeouts to

achieve reliable data transmission. An *acknowledgment* is a message sent by the receiver to indicate that it has correctly received a data frame.

Three types of ARQ protocols are Stop-and-wait ARQ, Go-Back-N ARQ, and Selective Repeat ARQ.

Applications

Applications that require low latency (such as telephone conversations) cannot use Automatic Repeat request (ARQ); they must use Forward Error Correction (FEC). By the time an ARQ system discovers an error and re-transmits it, the re-sent data will arrive too late to be any good.

Applications where the transmitter immediately forgets the information as soon as it is sent (such as most television cameras) cannot use ARQ; they must use FEC because when an error occurs, the original data is no longer available. (This is also why FEC is used in data storage systems such as RAID and distributed data store).

Applications that use ARQ must have a return channel. Applications that have no return channel cannot use ARQ.

Applications that require extremely low error rates (such as digital money transfers) must use ARQ.

Internet

In a typical TCP/IP stack, error control is performed at multiple levels:

- Each Ethernetframe carries a CRC-32checksum. Frames received with incorrect checksums are discarded by the receiver hardware.
- The IPv4 header contains a checksum protecting the contents of the header. Packets with mismatching checksums are dropped within the network or at the receiver.
- The checksum was omitted from the IPv6 header in order to minimize processing costs in network routing and because current link layer technology is assumed to provide sufficient error detection.
- UDP has an optional checksum covering the payload and addressing information from the UDP and IP headers. Packets with incorrect checksums are discarded by the operating systemnetwork stack. The checksum is optional under IPv4, only, because the Data-Link layer checksum may already provide the desired level of error protection.
- TCP provides a checksum for protecting the payload and addressing information from the TCP and IP headers. Packets with incorrect checksums are discarded within the network stack, and eventually get retransmitted using ARQ, either explicitly (such as through triple-ack) or implicitly due to a timeout.

EXPERIMENT NO. : 6

Implement various types of DLL protocols

The **Data Link Layer** (DLL) of the OSI Reference Model, also known as *Layer 2*, is responsible for providing error-free communication between adjacent devices on a network. To accomplish this, the DLL protocol must perform several functions:

Data Link Layer function

- ***Framing and Synchronization*:** The Physical Layer deals with the transmission of bits. At the DLL, bits are organized and blocked together into units typically called frames. Some information, then, must be added to the frame to indicate its beginning and end. The frame usually comprises a header, a trailer, and an information payload.

- ***Data Delimiting*:** Because the DLL protocol adds overhead to the information being transmitted, the system receiving the information has to be able to tell the difference between the overhead added by the transmitter and the actual data being transmitted.

- ***Detection/correction of bit errors*:** Most DLL protocols use a cyclic redundancy check (CRC) to detect bit errors. The CRC is typically found in the trailer. If the DLL also acts to correct bit errors, it typically does so by requesting a retransmission from the transmitter, a process known as backward error correction. Many modern DLL protocols simply discard errored frames, which require the upper layers (typically the Transport Layer) to resolve the loss.

- ***Addressing*:** If the underlying Physical Layer is a point-to-multipoint (P2MP) or multipoint-to-multipoint (MP2MP) circuit, then the DLL must also define an addressing structure to ensure that the correct system reads and copies the data. If present, the address is typically carried in the frame header.

- ***Access control*:** Again, if the underlying Physical Layer is a point-to-multipoint (P2MP) or multipoint-to-multipoint (MP2MP) circuit, the DLL must ensure orderly access to the medium to prevent systems from colliding with one another on the circuit. Medium access control (MAC) schemes were commonly found in local area networks, most of which were originally broadcast in nature.

- ***Link establishment and termination*:** Some DLL protocols operate on the connectionless communication model, while other operates on a connection-oriented communication model. If the DLL is connection-oriented, then it must define procedures for establishing and terminating the DLL connection between devices. In most cases, once the DLL connection is established between two

systems, it is usually not terminated as long as the network is operational. Although less common today, this model is still present for such things as dial-access to the Internet.

- ***Flow control*:** The DLL may also implement flow control mechanisms. Although typically associated with connection-oriented DLLs, there are some connectionless DLLs (e.g., some version of Ethernet) that also define flow control mechanisms.

Protocol examples

- Address Resolution Protocol (ARP)
- ARCnet
- ATM
- Cisco Discovery Protocol (CDP)
- Controller Area Network (CAN)
- Econet
- Ethernet
- Ethernet Automatic Protection Switching (EAPS)
- Fiber Distributed Data Interface (FDDI)
- Frame Relay
- High-Level Data Link Control (HDLC)
- IEEE 802.2 (provides LLC functions to IEEE 802 MAC layers)
- IEEE 802.11 wireless LAN
- LattisNet
- Link Access Procedures, D channel (LAPD)
- LocalTalk
- Multiprotocol Label Switching (MPLS)
- Nortel Discovery Protocol (NDP)
- Split multi-link trunking (SMLT)
- Point-to-Point Protocol (PPP)
- Serial Line Internet Protocol (SLIP) (obsolete)
- Spanning Tree Protocol
- StarLan
- Token ring
- Unidirectional Link Detection (UDLD)

EXPERIMENT NO. : 7

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 NO. : 8

Study and Installation of Standard Network Simulator

In communication and computer network research, **network simulation** is a technique where a program models the behavior of a network either by calculating the interaction between the different network entities (hosts/routers, data links, packets, etc.) using mathematical formulas, or actually capturing and playing back observations from a production network. The behavior of the network and the various applications and services it supports can then be observed in a test lab; various attributes of the environment can also be modified in a controlled manner to assess how the network would behave under different conditions. When a simulation program is used in conjunction with live applications and services in order to observe end-to-end performance to the user desktop, this technique is also referred to as network emulation.

Network simulator

A **network simulator** is a piece of software or hardware that predicts the behavior of a network, without an actual network being present. A network simulator is a software program that imitates the working of a computer network. In simulators, the computer network is typically modeled with devices, traffic etc. and the performance is analyzed. Typically, users can then customize the simulator to fulfill their specific analysis needs. Simulators typically come with support for the most popular protocols in use today, such as WLAN, Wi-Max, UDP, and TCP.

Simulations

- Most of the commercial simulators are GUI driven, while some network simulators require input scripts or commands (network parameters). The network parameters describe the state of the network (node placement, existing links) and the events (data transmissions, link failures, etc.). An important output of simulations is the trace files. Trace files can document every event that occurred in the simulation and are used for analysis. Certain simulators have added functionality of capturing this type of data directly from a functioning production environment, at various times of the day, week, or month, in order to reflect average, worst-case, and best-case conditions. Network simulators can also provide other tools to facilitate visual analysis of trends and potential trouble spots.
- Most network simulators use discrete event simulation, in which a list of pending "events" is stored, and those events are processed in order, with some events triggering future events—such as the event of the arrival of a packet at one node triggering the event of the arrival of that packet at a downstream node.
- Some network simulation problems, notably those relying on queueing theory, are well suited to Markov chain simulations, in which no list of future events is maintained and the simulation consists of transiting between different system "states" in a memory less fashion. Markov chain simulation is typically faster but less accurate and flexible than detailed discrete event simulation. Some simulations are cyclic based simulations and these are faster as compared to event based simulations.
- Simulation of networks can be a difficult task. For example, if congestion is high, then estimation of the average occupancy is challenging because of high variance. To estimate the likelihood of a buffer overflow in a network, the time required for an accurate answer can be extremely large. Specialized techniques such as "control variants" and "importance sampling" have been developed to speed simulation.

Examples of network simulators

Examples of notable network simulation software are, ordered after how often they are mentioned in research papers:

1. ns2/ns3
2. OPNET
3. NetSim

1. NS (simulator)

Ns (from **network simulator**) is a name for series of discrete eventnetwork simulators, specifically **ns-1**, **ns-2** and **ns-3**. These simulators are used in the simulation of routing protocols, among others, and are heavily used in ad-hoc networking research, and support popular network protocols, offering simulation results for wired and wireless networks alike.

Design

Ns-3 is built using C++ and Python and scripting is available with either language. Split over 30 modules, features of ns-3 include:

- Callback-driven events
- Attribute system that manages default and per-object simulation values
- Helpers that allow using simpler API when configuring simulations

Workflow for ns

It includes four steps:

- Implement protocol models
- Setup simulation scenario, i.e. create Tcl file describing type of scenario, e.g. number of nodes, kind of agent working on nodes etc.
- Run simulation, i.e. Run the Tcl file
- Analyze simulation results, i.e. by GNU Awk and gnuplot

Components

- Ns, the simulator itself has Nam, the network animator to visualize ns (or other) output.
- Pre-processing component for Traffic and topology generators.
- Post-processing for Simple trace analysis, often in Awk, Perl, or Tcl.

Installation of NS 2

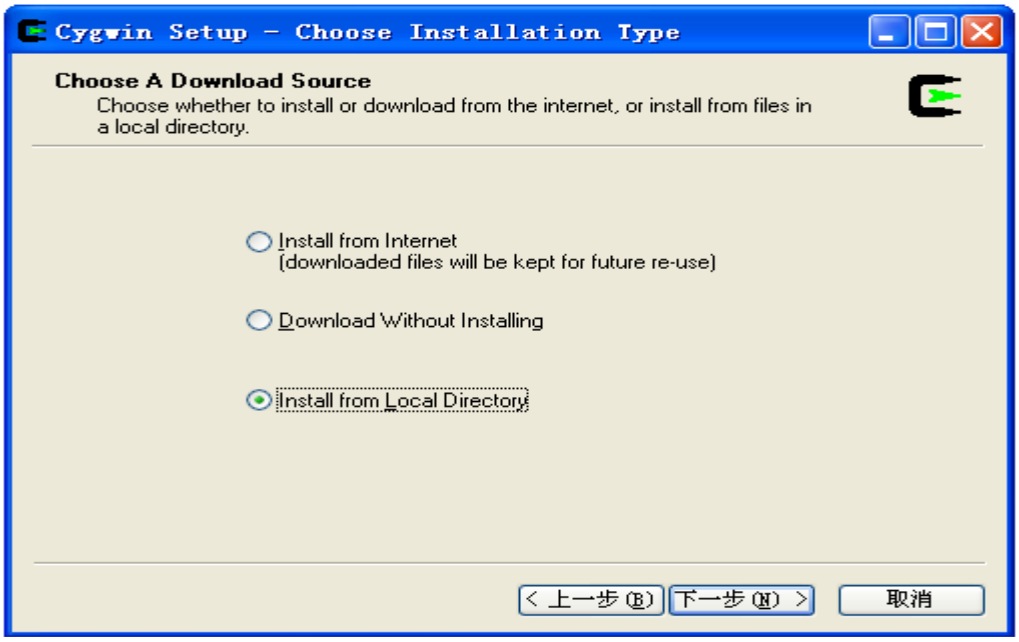
1. Install CYGWIN

- **Download zip file of ns2.29 (allinone) from:**

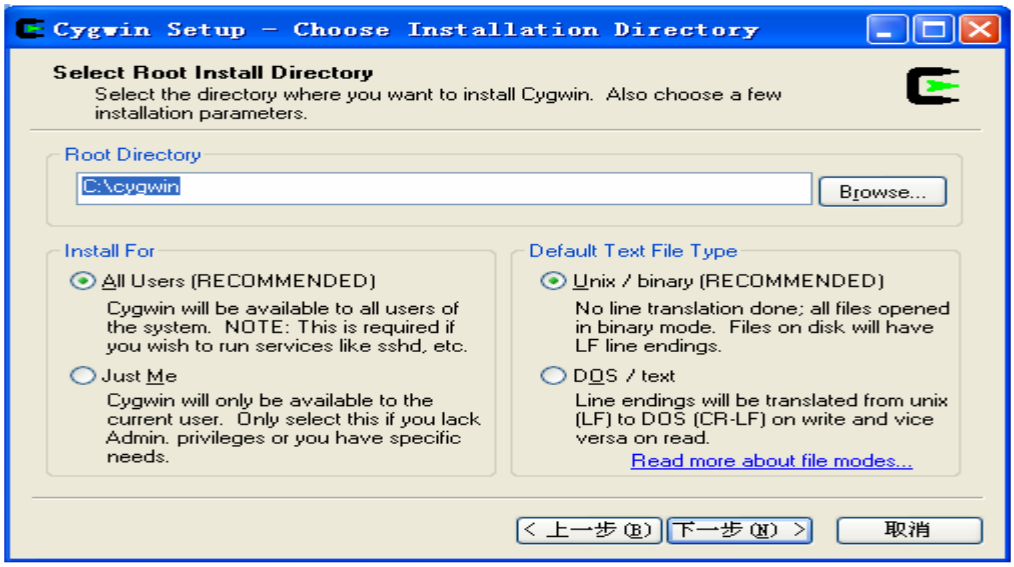
http://www.isi.edu/nsnam/dist/ns-allinone-2.29.2.tar.gz

- **Download cygwin setup.exe from www.cygwin.com**
- **Click on "cygwin.exe".**

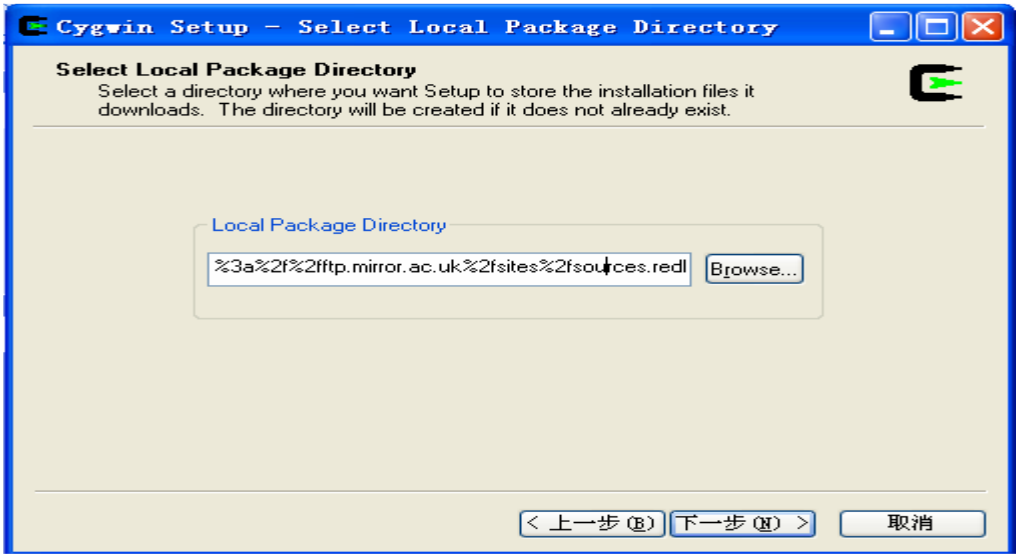
- **Select "install local directory"**



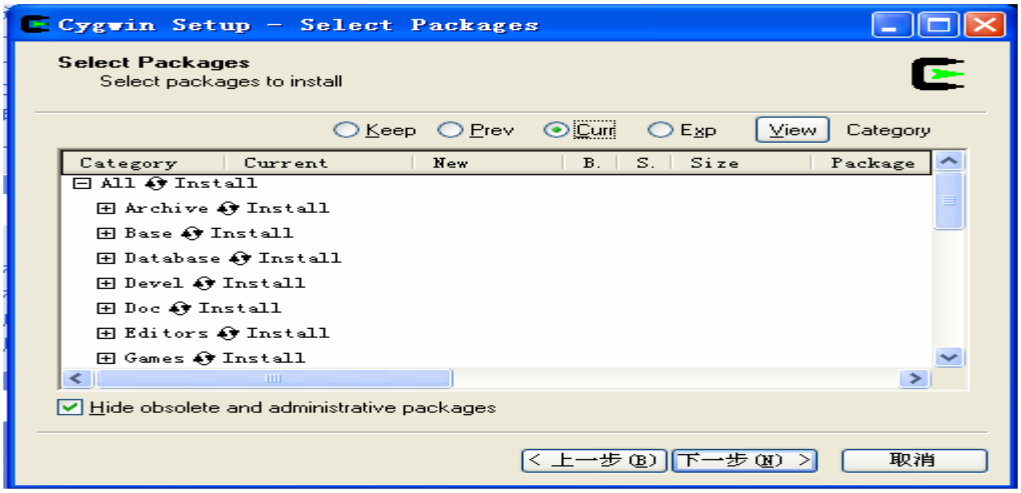
- **Select browse for "cygwin" file (I selected as default).**



- Install
  "C:\NS2\NS-2.29 Install files\Cygwin files\ftp%3a%2f%2fftp.mirror.ac.uk%2fsites%2fsources.redhat.com%2fftp%2fcygwin"

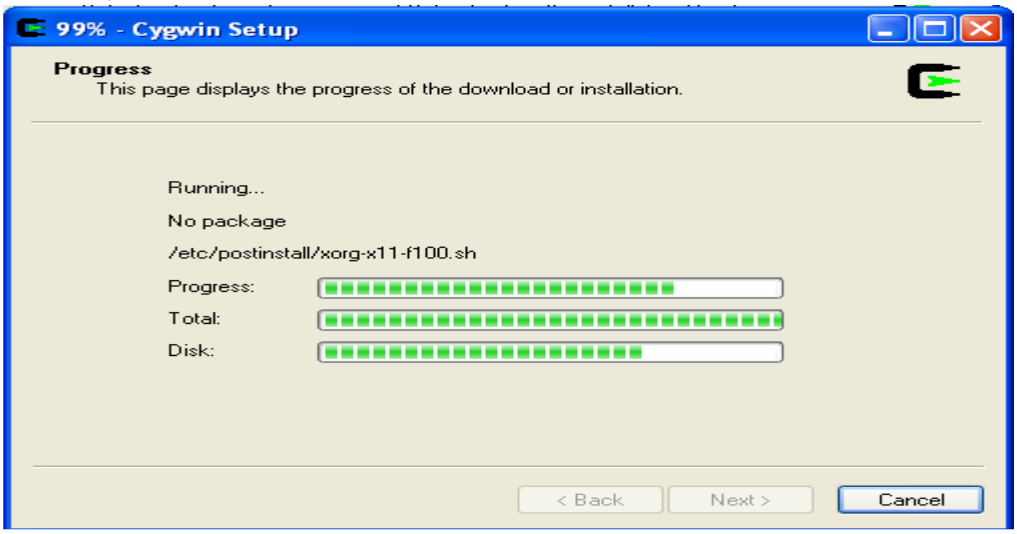


- **Select to install all**

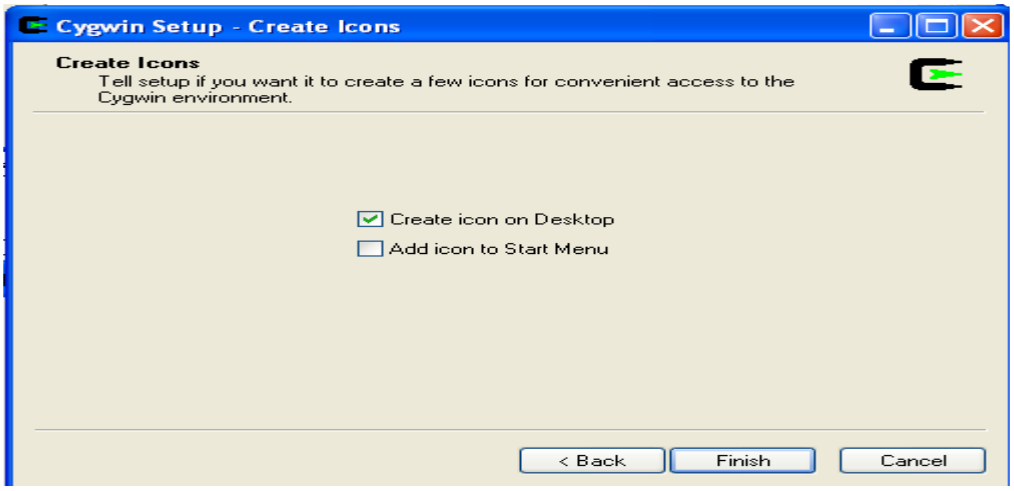


You can select “install”, “default”, “uninstall”, “install”, and “reinstall” behind the first line “all”. If we select “install”, then all sub items will be automatically selected. Otherwise you may lose some items.

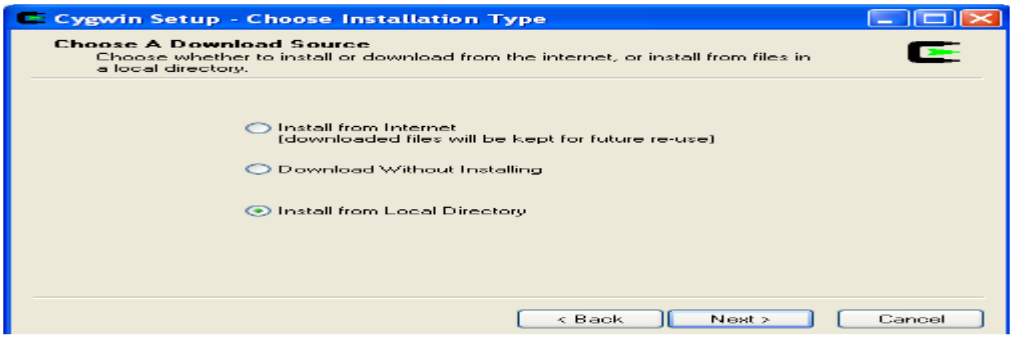
- **Installing**

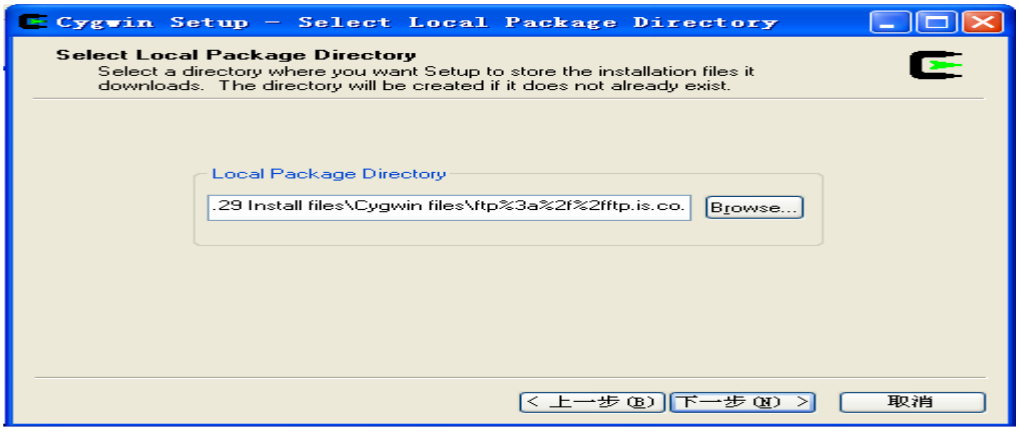


- Finish in installing “ftp%3a%2f%2fftp.mirror.ac.uk%2fsites%2fsources.redhat.com%2fftp%2fcygwin”

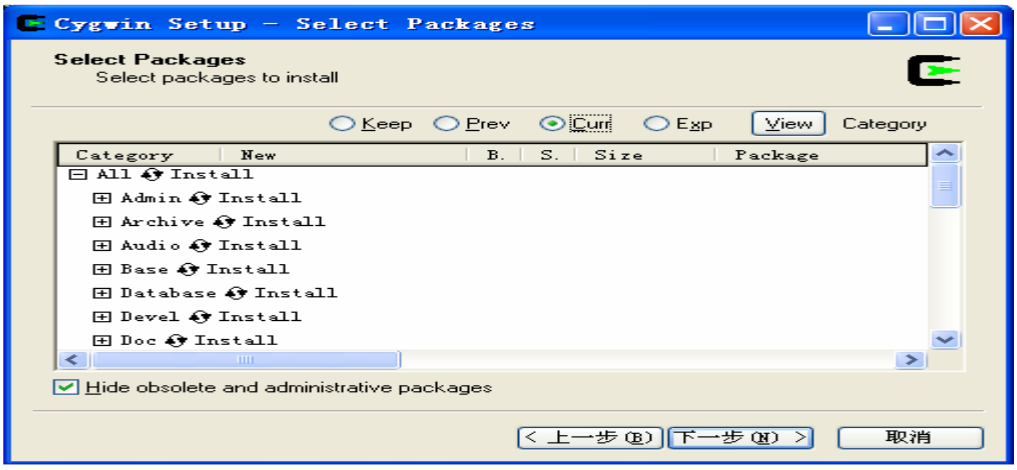


- Install “Install “C:\NS2\NS-2.29 Install files\Cygwin files \ ftp%3a%2f%2fftp.is.co.za%2fmirrors%2fcygwin”.

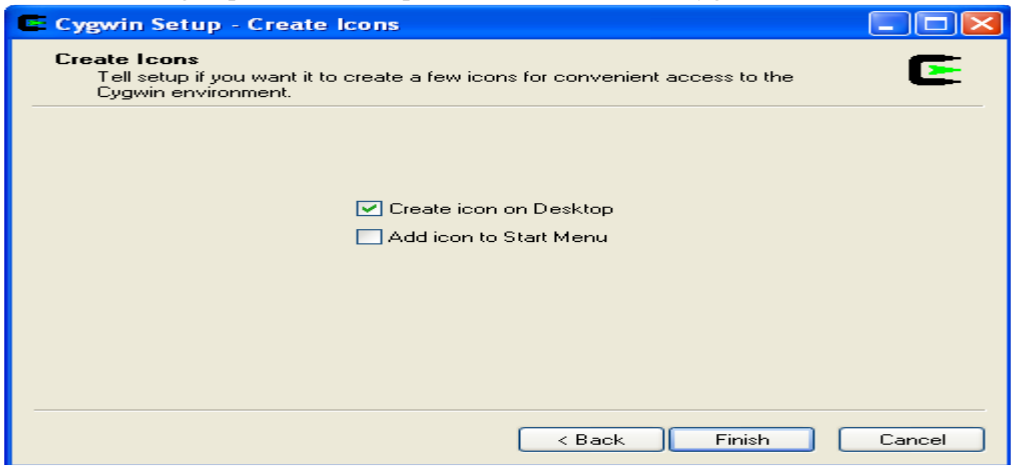




- Select to install all.

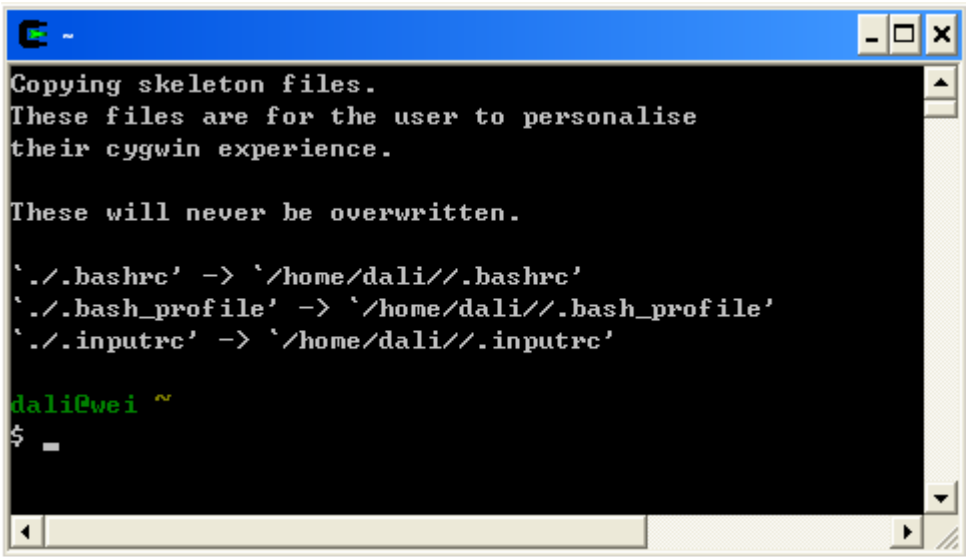


- Finish in stalling “ftp%3a%2f%2fftp.is.co.za%2fmirrors%2fcygwin”



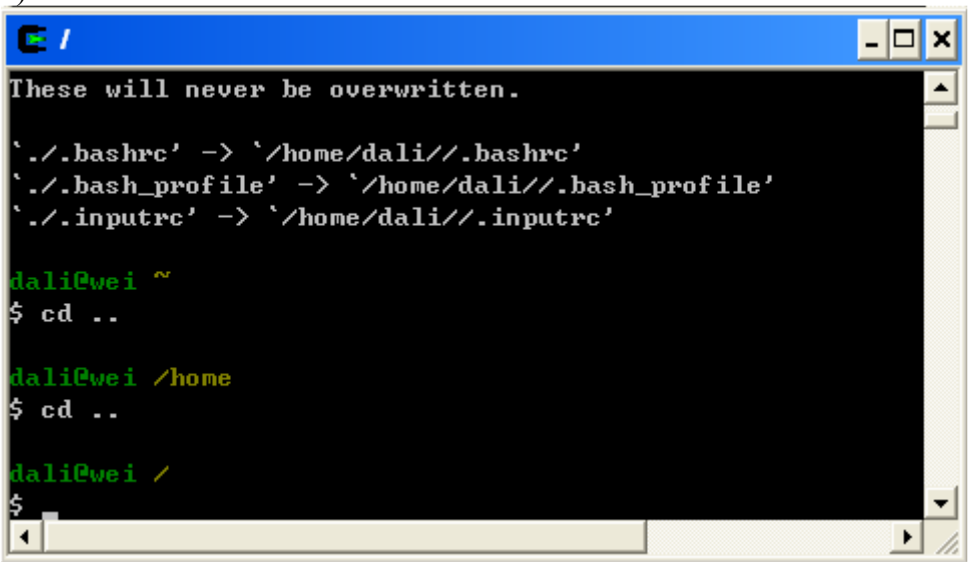
2. **Install NS2**

a. Copy ns-allinone-2.29.2.tar to directory c:/cygwin/usr/local
b. Unzip “ns-allinone-2.29.2.tar”
c. Click on desktop icon “cygwin”



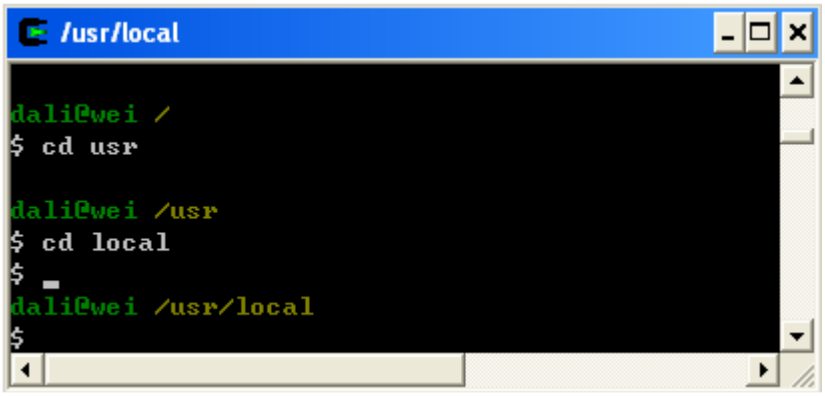
d. Type “cd ..” to go to the upper folder(“cd” must be low case. And there is one space between “d” and “.”)

e. Type “cd ..” again.

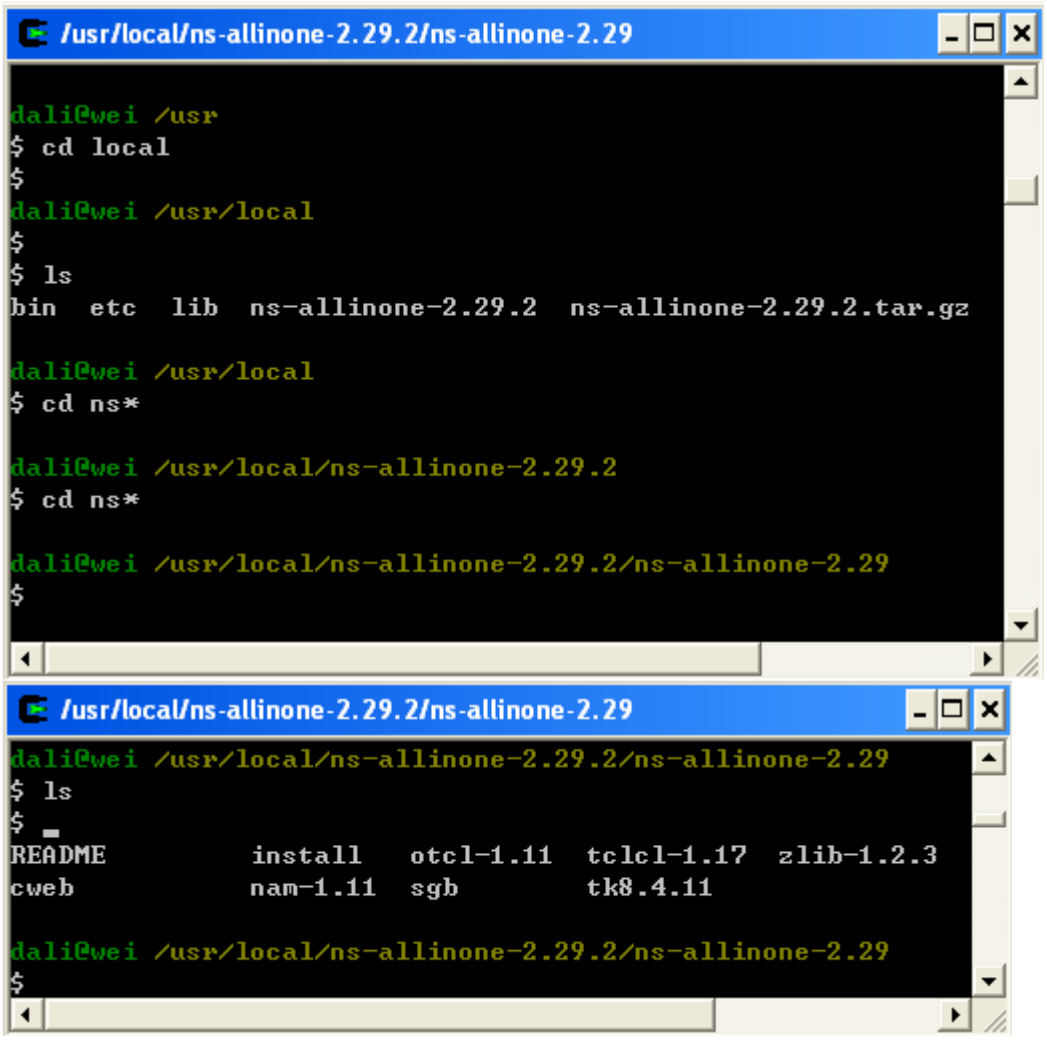


f. “cd usr”, go to folder “usr”

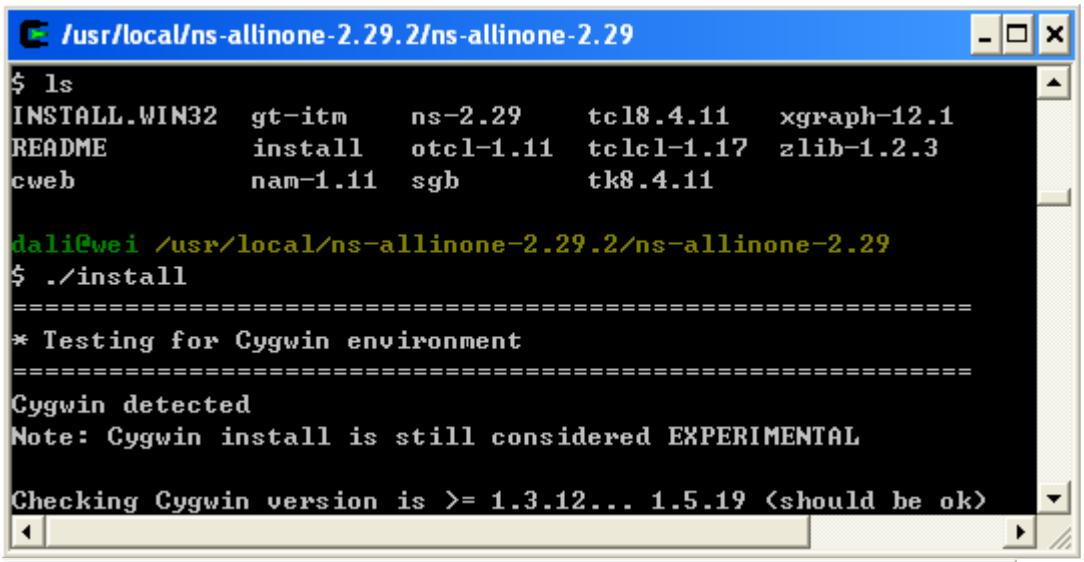
g. “cd local”, go to folder “local”

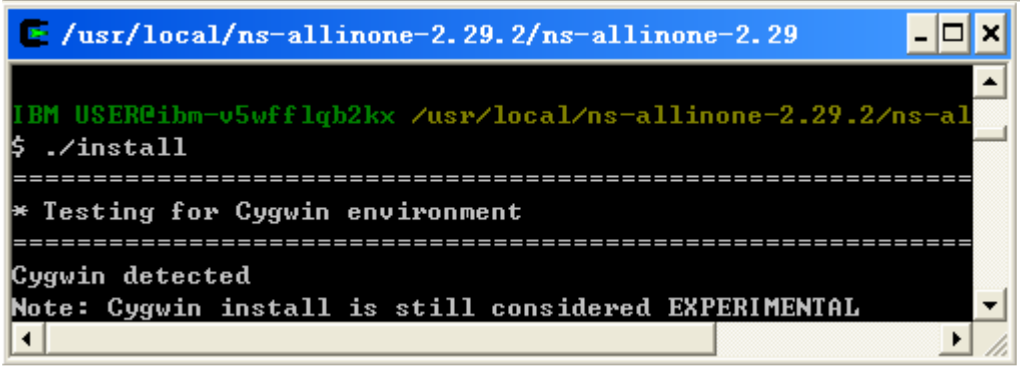


h. Find the “install.exe” file

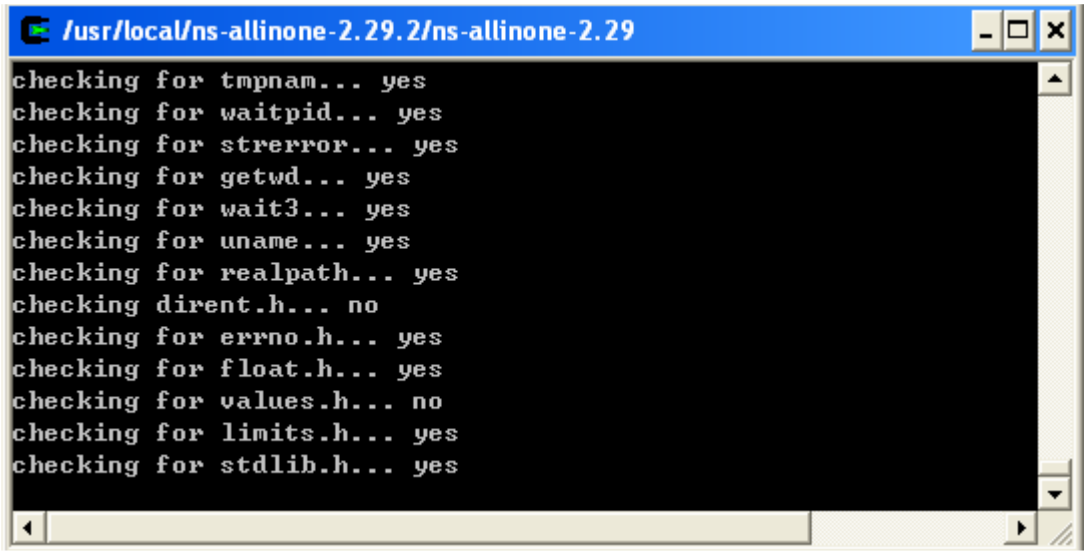


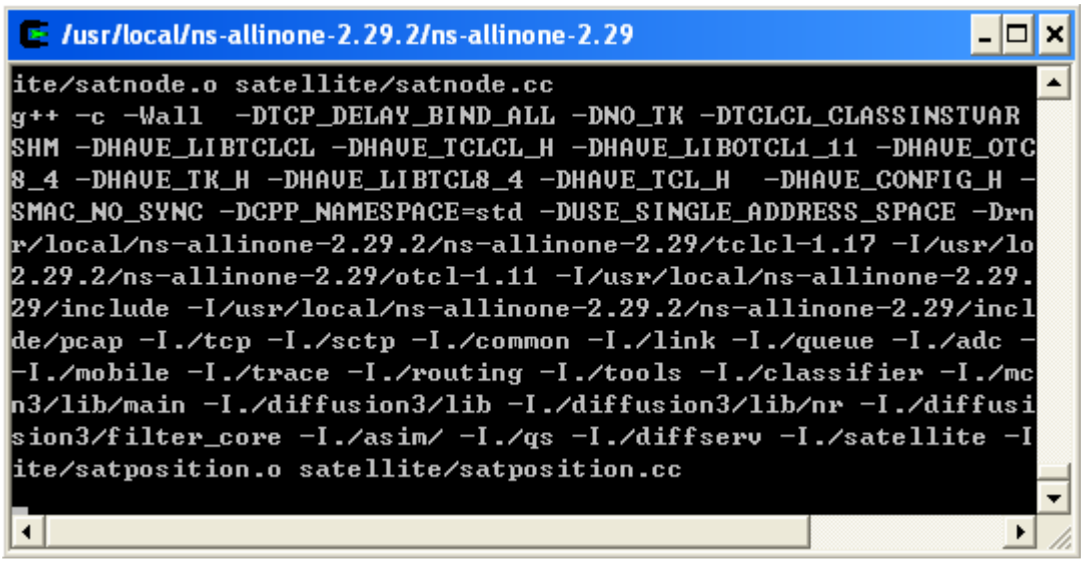
i. Start to run the installation “./install” (Attention: After finish installing, some comments will come out. In my computer with celeron 2.66 CPU and 512 M Memory, this step took around 75 minutes.)



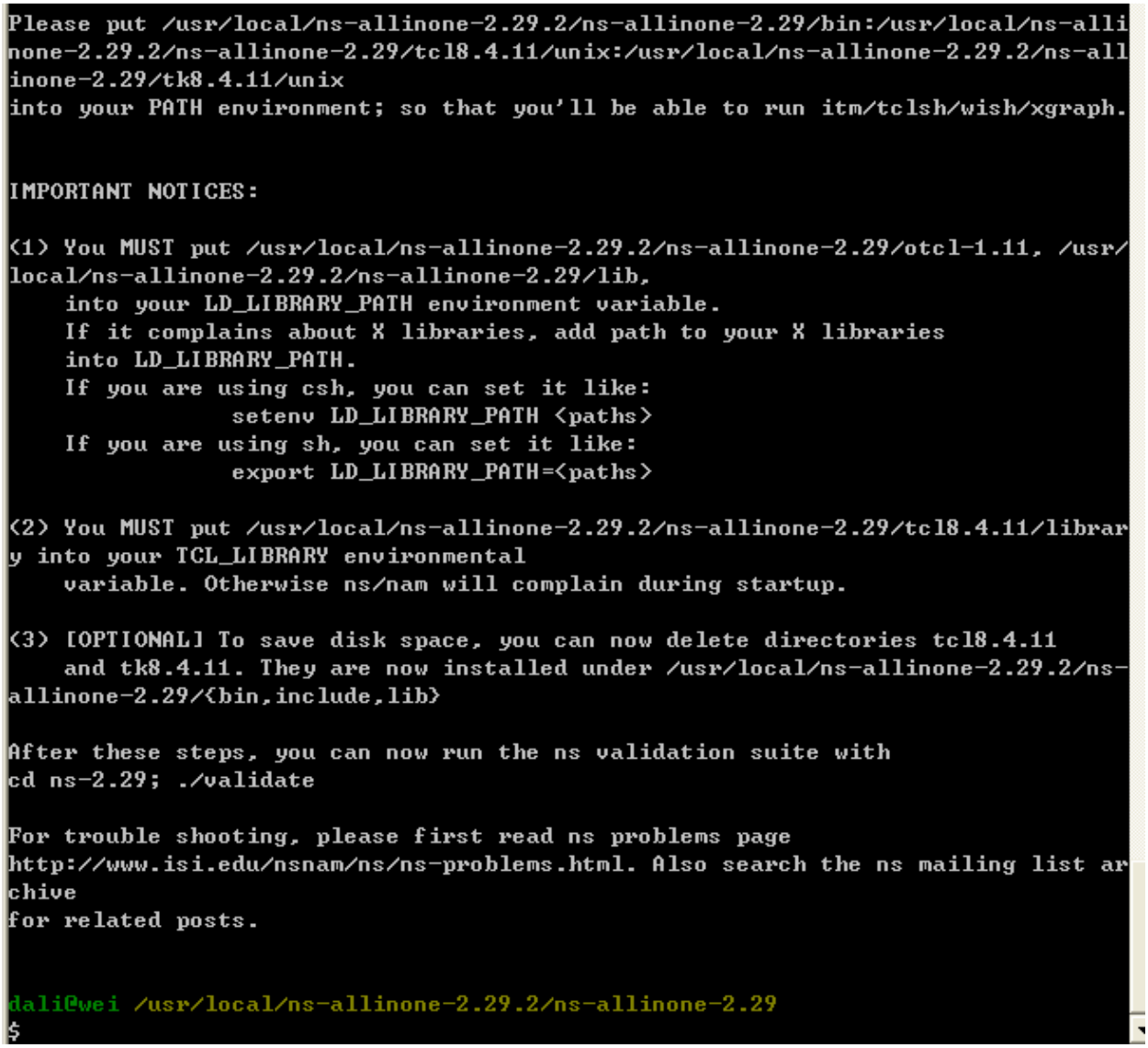


j. Installing




/usr/local/ns-allinone-2.29.2/ns-allinone-2.29
ite/satnode.o satellite/satnode.cc
g++ -c -Wall -DTCP_DELAY_BIND_ALL -DNO_TK -DTCLCL_CLASSINSTVAR
SHM -DHAVE_LIBTCLCL -DHAVE_TCLCL_H -DHAVE_LIBOTCL1_11 -DHAVE_OTC
8_4 -DHAVE_TK_H -DHAVE_LIBTCL8_4 -DHAVE_TCL_H -DHAVE_CONFIG_H -
SMAC_NO_SYNC -DCPP_NAMESPACE=std -DUSE_SINGLE_ADDRESS_SPACE -Drn
r/local/ns-allinone-2.29.2/ns-allinone-2.29/tclcl-1.17 -I/usr/lo
2.29.2/ns-allinone-2.29/otcl-1.11 -I/usr/local/ns-allinone-2.29.
29/include -I/usr/local/ns-allinone-2.29.2/ns-allinone-2.29/incl
de/pcap -I./tcp -I./sctp -I./common -I./link -I./queue -I./adc -
-I./mobile -I./trace -I./routing -I./tools -I./classifier -I./mc
n3/lib/main -I./diffusion3/lib -I./diffusion3/lib/nr -I./diffusi
sion3/filter_core -I./asim/ -I./qs -I./diffserv -I./satellite -I
ite/satposition.o satellite/satposition.cc

3. **Configure system variables and library paths**
    a. Installing is finished and the following window appears: Make sure to copy all these contents.



EXPERIMENT NO. : 8

Different Types of Cryptographic Algorithms

DES

DES (the Data Encryption Standard) is a symmetric block cipher developed by IBM. The algorithm uses a 56-bit key to encipher/decipher a 64-bit block of data. The key is always presented as a 64-bit block, every 8th bit of which is ignored. However, it is usual to set each 8th bit so that each group of 8 bits has an odd number of bits set to 1.

The algorithm is best suited to implementation in hardware, probably to discourage implementations in software, which tend to be slow by comparison. However, modern computers are so fast that satisfactory software implementations are readily available.

DES is the most widely used symmetric algorithm in the world, despite claims that the key length is too short. Ever since DES was first announced, controversy has raged about whether 56 bits is long enough to guarantee security.

The key length argument goes like this. Assuming that the only feasible attack on DES is to try each key in turn until the right one is found, then 1,000,000 machines each capable of testing 1,000,000 keys per second would find (on average) one key every 12 hours. Most reasonable people might find this rather comforting and a good measure of the strength of the algorithm.

Those who consider the exhaustive key-search attack to be a real possibility (and to be fair the technology to do such a search is becoming a reality) can overcome the problem by using double or triple length keys. In fact, double length keys have been recommended for the financial industry for many years.

Use of multiple length keys leads us to the Triple-DES algorithm, in which DES is applied three times. If we consider a triple length key to consist of three 56-bit keys K1, K2, K3 then encryption is as follows:

- Encrypt with K1
- Decrypt with K2
- Encrypt with K3

Decryption is the reverse process:

- Decrypt with K3
- Encrypt with K2
- Decrypt with K1

Setting K3 equal to K1 in these processes gives us a double length key K1, K2.

Setting K1, K2 and K3 all equal to K has the same effect as using a single-length (56-bit key). Thus it is possible for a system using triple-DES to be compatible with a system using single-DES.

RSA

RSA is a public key algorithm invented by Rivest, Shamir and Adleman. The key used for encryption is different from (but related to) the key used for decryption.

The algorithm is based on modular exponentiation. Numbers e, d and N are chosen with the property that if A is a number less than N, then (Ae mod N)d mod N = A.

This means that you can encrypt A with e and decrypt using d. Conversely you can encrypt using d and decrypt using e (though doing it this way round is usually referred to as signing and verification).

- The pair of numbers (e,N) is known as the public key and can be published.
- The pair of numbers (d,N) is known as the private key and must be kept secret.

The number e is known as the public exponent, the number d is known as the private exponent, and N is known as the modulus. When talking of key lengths in connection with RSA, what is meant is the modulus length.

An algorithm that uses different keys for encryption and decryption is said to be asymmetric.

Anybody knowing the public key can use it to create encrypted messages, but only the owner of the secret key can decrypt them.

Conversely the owner of the secret key can encrypt messages that can be decrypted by anybody with the public key. Anybody successfully decrypting such messages can be sure that only the owner of the secret key could have encrypted them. This fact is the basis of the digital signature technique.

Without going into detail about how e, d and N are related, d can be deduced from e and N if the factors of N can be determined. Therefore the security of RSA depends on the difficulty of factorizing N. Because factorization is believed to be a hard problem, the longer N is, the more secure the cryptosystem. Given the power of modern computers, a length of 768 bits is considered reasonably safe, but for serious commercial use 1024 bits is recommended.

The problem with choosing long keys is that RSA is very slow compared with a symmetric block cipher such as DES, and the longer the key the slower it is. The best solution is to use RSA for digital signatures and for protecting DES keys. Bulk data encryption should be done using DES.

MD5

MD5 is a hashing algorithm that takes a message of up to 264 bits and reduces it to a digest of 128 bits (16 bytes).

The algorithm is a development of the MD4 algorithm invented by Ronald Rivest and announced in 1990. Unfortunately, MD4 was flawed, so Rivest made some revisions, and the resulting algorithm was christened MD5.

Any hashing (or digest) algorithm should be such that, given a digest and the corresponding message from which it was derived, it should be computationally infeasible to construct a different message with the same digest.

AES

AES, short for Advanced Encryption Standard, is a widely adopted symmetric encryption scheme used, for instance, to secure electronic communication and messages. AES – as its name implies - has been the outcome of standardization and evaluation process which took years to select from the best encryption algorithms. Finally, in 2001, the Rijndael algorithm has been chosen as winner by the US National Institute of Standards and Technology (NIST) to be implemented as underlying security algorithm of the AES standard which as of the these days has largely replaced its predecessor and derivates of DES (Data Encryption Standard) which is longer considered secure due to its small 56-bit key length for example.

The Rijndael algorithm, invented by two cryptographers Vincent Rijmen and Joan Daemen, implements the mathematical operations substitution, transposition, as well as permutation to plaintext, the term used to describe input in the cryptography domain. The AES Advanced Encryption Standard uses 10 rounds of these algebraic operations in a complex scheme to produce encrypted output, or cipher text as it is called in expert terms. AES-192 and AES-256 have 12 and 14 rounds, respectively.

In the AES implementation of Rijndael the algorithm operates on 128 bits block ciphers, and comprises key lengths of 128, 192 and 256 bits. It is common to refer to the symmetric key AES encryption standard as AES-128, AES-192 and AES-256 depending on the key strength. More about encryption can also be found in Bright Hub's article Types of Encryption which explains the difference between asymmetric and symmetric encryption also shedding a light on stream and block ciphers.

Whereas cryptography aims at securing plain text does cryptanalysis try to break the key or underlying algorithm of an encryption scheme, Rijndael in the case of AES here. Cracking a 256-bit key is computationally infeasible but cryptanalysts who are aware of the inner working of Rijndael and who apply much more sophisticated methods than brute-force believe that the security margin is narrowing. Check out our article Can AES Encryption be Cracked? which takes into account the latest news about the security or strength of AES.

SHA-1

SHA stands for Secure Hash Algorithm. It consists of five hash functions designed by the National Security Agency (NSA) and published by the National Institute of Standards and Technology (NIST). The five algorithms are SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512. SHA-1 is the most commonly used of the SHA series.

Hash algorithms are called secure when

1. It is impossible to find a message that corresponds to a given message digest.
2. It is impossible to find two different messages that produce the same message digest.
3. If a message is changed even by a single character, the result will be a completely different message digest.

SHA-1 has these properties and is therefore referred to as secure. It is designed to work with the Digital Signature Algorithm (DSA). SHA-1 is a one-way hash function. One-way functions are characterized by two properties. The first is that they are one-way. This means that you can take a message and compute a hash

value, but you cannot take a hash value and recreate the original message. It is also collision-free and, thus, no two messages can hash to the same value.

SHA-1 produces a 160-bit message digest with a maximum length of 264 ?1. The message M to be hashed must have a length of l bit, where 0 l 264. The message digest is the fixed-length output of a message. The message digest is then input to the DSA, which will then generate the signature for the message. Signing the message digest instead of the message offers improved performance because the message digest will be much smaller than the message. The recipient of the message will then use the same hash algorithm to verify the signature. Any change that occurs during transit will result in a different message digest and, thus, the signature will not verify. Once it is verified as true, the recipient is able to unlock the message. This method prevents unauthorized users from viewing messages that are not intended for them.

When computing a message digest, SHA-1 processes blocks of 512 bits. The total length of the message digest will be a multiple of 512. This process is known as padding of the message.

SHA-1 differs from SHA-0 only by a single bitwise rotation in the message schedule of its compression function.

Cryptanalysis is the method of obtaining encrypted information without using the hash value. Breaking a hash function implies showing that the one-way property does not hold for it. Cryptographers have demonstrated that it just might be possible for the SHA-1 hash algorithm to be broken. Some have presented a collision for 58-round SHA-1, found with 233 hash operations. A brute force search would require 280 operations. However, experts argue that this might not happen for some time. Nevertheless, attacks always get better, and the National Institute of Standards and Technology (NIST) already has standards for longer – and harder-to-break – hash functions: SHA-224, SHA-256, SHA-384, and SHA-512.

Applications of SHA-1

SHA-1 can be used in a variety of applications:

1. Security applications that require authentication
2. E-mail
3. Electronic funds transfer
4. Software distribution
5. Data storage

EXPERIMENT NO. : 9

Voice over IP

Voice over IP (VoIP) commonly refers to the communication protocols, technologies, methodologies, and transmission techniques involved in the delivery of voice communications and multimedia sessions over Internet Protocol (IP) networks, such as the Internet. Other terms commonly associated with VoIP are *IP telephony*, *Internet telephony*, *voice over broadband* (VoBB), *broadband telephony*, and *broadband phone*.

Internet telephony refers to communications services —voice, fax, SMS, and/or voice-messaging applications— that are transported via the Internet, rather than the public switched telephone network (PSTN). The steps involved in originating a VoIP telephone call are signaling and media channel setup, digitization of the analog voice signal, encoding, packetization, and transmission as Internet Protocol (IP) packets over a packet-switched network. On the receiving side, similar steps (usually in the reverse order) such as reception of the IP packets, decoding of the packets and digital-to-analog conversion reproduce the original voice stream.Even though IP Telephony and VoIP are terms that are used interchangeably, they are actually different; IP telephony has to do with digital telephony systems that use IP protocols for voice communication, while VoIP is actually a subset of IP Telephony. VoIP is a technology used by IP telephony as a means of transporting phone calls.

VoIP systems employ session control protocols to control the set-up and tear-down of calls as well as audio codecs which encode speech allowing transmission over an IP network as digital audio via an audio stream. The choice of codec varies between different implementations of VoIP depending on application requirements and network bandwidth; some implementations rely on narrowband and compressed speech, while others support high fidelitystereo codecs. Some popular codecs include u-law and a-law versions of G.711;G.722 which is a high-fidelity codec marketed as HD Voice by Polycom, a popular open source voice codec known as iLBC, a codec that only uses 8kbps each way called G.729, and many others.

VoIP is available on many smartphones and Internet devices so that users of portable devices that are not phones, may place calls or send SMS text messages over 3G or Wi-Fi.

Protocols

Voice over IP has been implemented in various ways using both proprietary and open protocols and standards. Examples of the network protocols used to implement VoIP include:

- H.323
- Media Gateway Control Protocol (MGCP)
- Session Initiation Protocol (SIP)
- Real-time Transport Protocol (RTP)
- Session Description Protocol (SDP)
- Inter-Asterisk eXchange (IAX)

The H.323 protocol was one of the first VoIP protocols that found widespread implementation for long-distance traffic, as well as local area network services. However, since the development of newer, less complex protocols such as MGCP and SIP, H.323 deployments are increasingly limited to carrying existing long-haul network traffic. In particular, the Session Initiation Protocol (SIP) has gained widespread VoIP market penetration.

A notable proprietary implementation is the Skype protocol, which is in part based on the principles of peer-to-peer (P2P) networking.

Security

- VoIP telephone systems are susceptible to attacks as are any Internet-connected devices. This means that hackers who know about these vulnerabilities (such as insecure passwords) can institute denial-of-service attacks, harvest customer data, record conversations and break into voice mailboxes.
- Another challenge is routing VoIP traffic through firewalls and network address translators. Private Session Border Controllers are used along with firewalls to enable VoIP calls to and from protected networks. For example, Skype uses a proprietary protocol to route calls through other Skype peers on the network, allowing it to traverse symmetric NATs and firewalls. Other methods to traverse NATs involve using protocols such as STUN or Interactive Connectivity Establishment (ICE).
- Many consumer VoIP solutions do not support encryption, although having a secure phone is much easier to implement with VoIP than traditional phone lines. As a result, it is relatively easy to eavesdrop on VoIP calls and even change their content. An attacker with a packet sniffer could intercept your VoIP calls if you are not on a secure VLAN. However, physical security of the switches within an enterprise and the facility security provided by ISPs make packet capture less of a problem than originally foreseen. Further research has shown that tapping into a fiber optic network without detection is difficult if not impossible. This means that once a voice packet is within the Internet backbone it is relatively safe from interception.
- There are open source solutions, such as Wireshark, that facilitate sniffing of VoIP conversations. Securing the content of conversations from malicious observers requires encryption and cryptographic authentication which is sometimes difficult to find at a consumer level. The existing security standard Secure Real-time Transport Protocol (SRTP) and the new ZRTP protocol are available on Analog Telephone Adapters (ATAs) as well as various softphones. It is possible to use IPsec to secure P2P VoIP by using opportunistic encryption. In 2005, Skype invited a researcher, Dr Tom Berson, to assess the security of the Skype software, and his conclusions are available in a published report.

Securing VoIP

- To prevent the above security concerns government and military organizations are using voice over secure IP (VoSIP), secure voice over IP (SVoIP), and secure voice over secure IP (SVoSIP) to protect confidential and classified VoIP communications. Secure voice over IP is accomplished by encrypting VoIP with protocols such as SRTP or ZRTP. Secure voice over IP is accomplished by using Type 1 encryption on a classified network, like SIPRNet. Public Secure VoIP is also available with free GNU programs and in many popular commercial VoIP programs via libraries such as ZRTP.

Design and Implementation of the Voice over IP Network

The converged network is intended to support both voice and data communications and the design process should consider both of these requirements. Each network presents unique challenges, however ten factors should be considered in all cases. Note that not all steps progress linearly, and that some feedback between steps is required (Figure):

1. .Define the objective of the converged network, including the applications to be supported, such as video conferencing or a voice-enabled Web site. Reduce this objective to a short mission statement so that all parties involved understand the challenges at hand. In addition, document other ancillary factors that are driving this project. For example, perhaps the PBX system is scheduled for an upgrade, and now is an appropriate time to consider an overhaul of the entire networking infrastructure. Other circumstances might be support for a new enterprise application that is not possible within the confines of the existing infrastructure, or carrier contracts that are about to renew and thus afford the opportunity for voice/data network consolidation.

2. Understand the current voice and data operating environments. Include existing loading factors on LAN and WAN segments, plus the anticipated bandwidth requirements for any new data applications. Consider current call patterns and anticipated growth during your busy hour periods.

3. Prepare a System Requirements Document that delineates functions that the network must provide, and include the appropriate design objectives, such as end-to-end delay, reliability, redundancy, and so on. Determine if one architectural infrastructure, such as the ITU-T's H.323 or IETF's SIP, is preferred for this network application. Also consider any changes that will be necessary to the telephone dialing plans, WAN links, IP subnets or other issues that could impact end users.

4. Consider the adjunct systems that will support this network, such as network management, network analysis, security, and end-user help desks. Verify that any of these support systems, such as the network management console or protocol analyzer, are also prepared to support the new environment. Modify the Systems Requirements Document as necessary based on the results of this study.

5. Consult with both current and prospective vendors to obtain their input on the systems requirements. Solicit input from other customers of your vendors, your colleagues, or other

   Network managers in similar industries that may have already experienced the same challenges. Modify the Systems Requirements Document as necessary based on the results of this input.

6. Develop installation and systems acceptance plans, including interoperability testing between variouscomponents. Also verify that all signaling protocolsbetween dissimilar networks are compatible.

7. Solicit presentations and bids from qualified vendors, evaluate these proposals, and award the appropriate contracts. In some cases, it may be necessary to hold two rounds of bids: one that is open to all interested vendors, and a second round (called the short list) that is limited to the most qualified. Include in the bid documents the additional support systems that have been identified, such as network management and troubleshooting tools, so that acceptance testing of the system can proceed once installation is complete.

8. Develop an Installation and Cutover schedule with the winning vendor(s), plus other organizations (such as carriers) that are also part of the project. Verify the critical paths in the schedule, such as equipment order and circuit installation, and reach agreements from all involved parties regarding the importance of this schedule.

9. Install the new system in a test lab or pilot location, and test its operation and applications before expanding to the larger network. At this stage, it is also very helpful to use a protocol analyzer, such as the Sniffer Portable analyzer from Network General, to document successful communications transactions, such as call setups. This baseline documentation, done in a controlled environment, may prove quite helpful if deployment problems arise down the road. It is also helpful to verify that all dialing plan and network address changes have been successfully migrated to the new environment with this small group of stations before rolling out the implementation to hundreds or thousands of end users. Modify the Installation and Cut-over Schedule as necessary based on the results of this pilot installation.

10. As the production network installation proceeds, verify operation of all network components, including interoperability between multivendor systems and adjunct systems, such as voice mail processors. Conduct end user and system manager training classes, and document procedures that these individuals will need to know in order to be effective in the new environment. Complete the as-built drawings of the network and any other appropriate documentation as required.

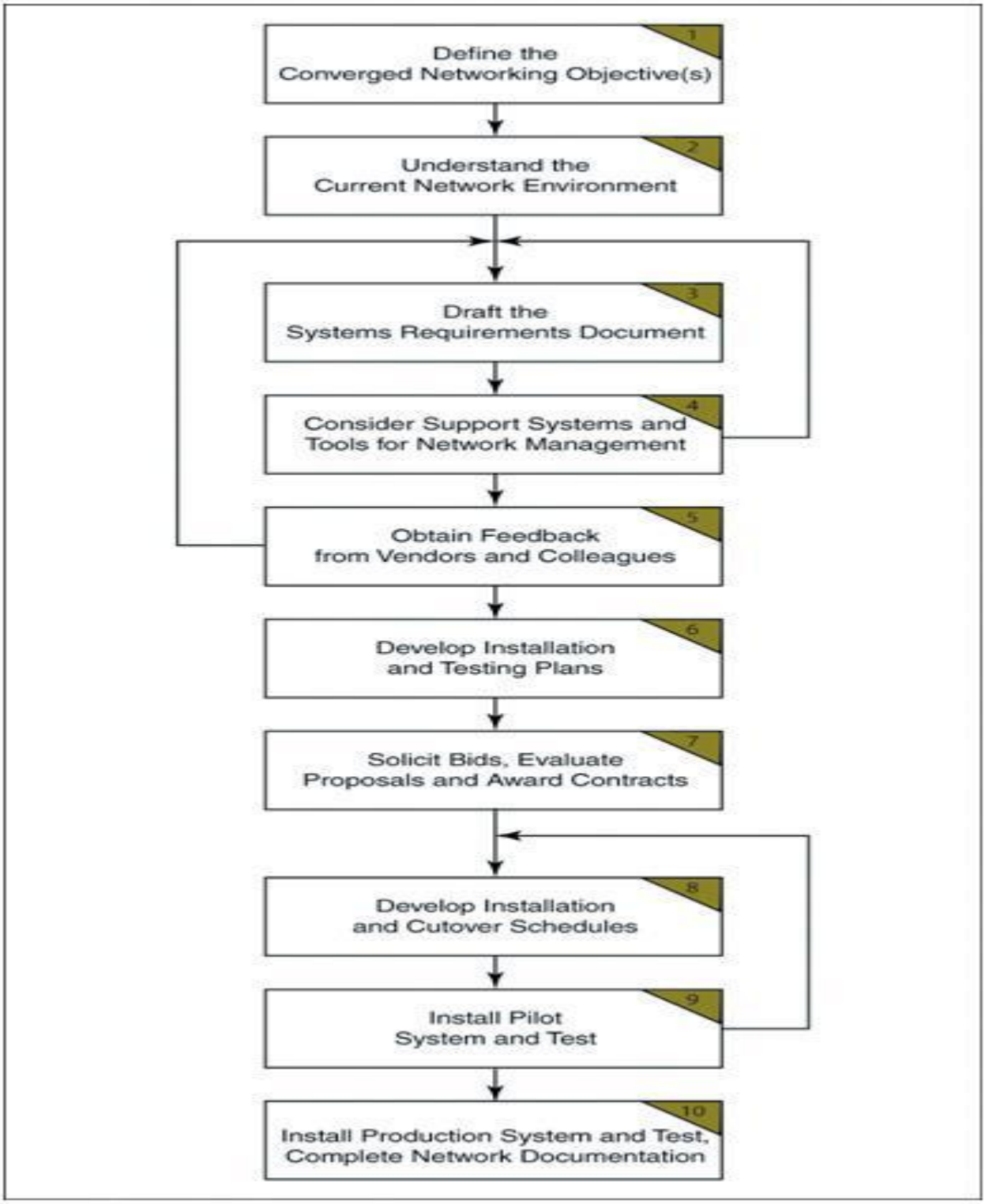


Figure . VoIP Network Design and Implementation Plan