Lecture 7.3

Application Layer: Remote Login(TELNET)

Dr. Vandana Kushwaha

Department of Computer Science

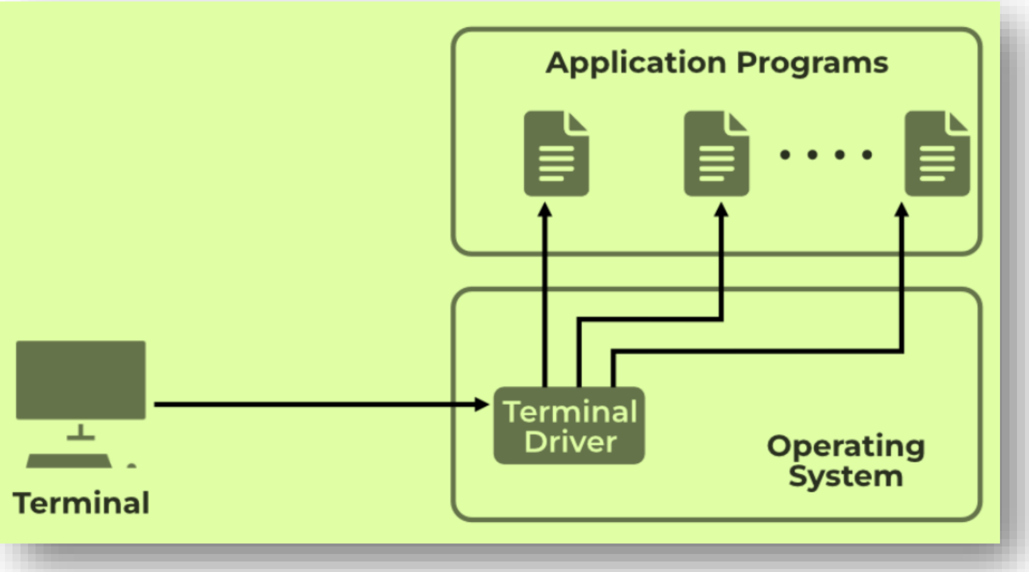
Institute of Science, BHU, Varanasi

Introduction

- In the **Internet**, users may want to **run** **application programs** at a **remote site** and create **results** that can be **transferred** to their **local site.**
- For **example**, **students** may want to **connect** to their **university computer lab** from their **home** to access **application programs** for doing **homework assignments** or **projects.**
- The better solution is a **general-purpose** **client/server program** that lets a user **access any** **application program** on a **remote computer**; in other words, allow the **user** to **log on** to a **remote computer.**
- After **logging on,** a **user** can use the **services available** on the **remote computer** and **transfer** the **results back** to the **local computer.**

Local Login

- Whenever a **user** logs into its **local system**, it is known as local login.
- The Procedure of Local Login
  - **Keystrokes** are **accepted** by the **terminal driver** when the **user types** at the **terminal.**
  - Terminal Driver passes these **characters** to **OS.**
  - Now, **OS validates** the **combination** of **characters** and **opens** the **required application.**



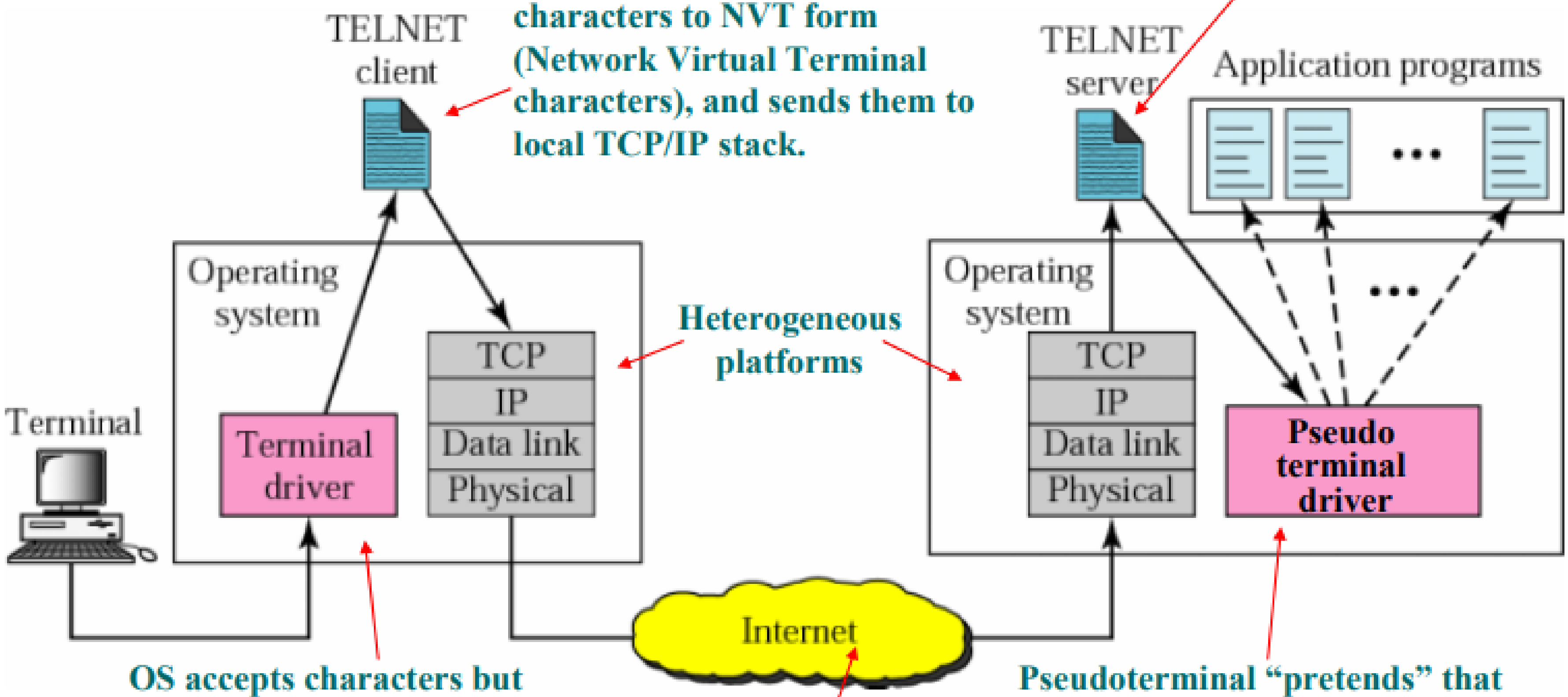
TELNET

- **TELNET** is an abbreviation for ***TErminaL NETwork*** *is* a **client/server application program** *is the **standard** **TCP/IP protocol** for **virtual*** **terminal service** as proposed by the **International Organization for Standards (ISO).**
- **TELNET** enables the **establishment** of a **connection** to a **remote system** in such a way that the **local terminal** appears to be a **terminal** at the **remote system**.
- When a **user** wants to **access** an **application program** or **utility** located on a **remote machine**, he performs **remote log-in** using a **TELNET client** and **server programs**.
- The **user sends** the **keystrokes** to the **terminal driver,** where the **local operating system accepts** the **characters** but **does not interpret** them.
- The **characters are sent** to the **TELNET client**, which **transforms the characters** to a **universal character set** called ***network virtual terminal (NVT) characters*** *and **delivers** them to the **local TCP/IP*** **protocol stack.**

TELNET

- The **commands or text**, in **NVT form**, **travel through** the **Internet** and **arrive at** the **TCP/IP stack** at the **remote machine.**
- Here the **characters** are **delivered to** the **operating system** and passed to the **TELNET server**, which **changes the characters** to the corresponding **characters understandable** by the **remote computer.**
- However, the **characters cannot be passed directly** to the **operating system** because the **remote operating system** is **not designed to receive characters** from a **TELNET server.**
- It is designed to **receive characters** from a **terminal driver.**
- The **solution** is to **add** a piece of **software** called a ***pseudoterminal driver*** *which* ***pretends*** *that the* ***characters*** *are* ***coming from*** a **terminal.**
- The **operating system** then **passes** the **characters** to the **appropriate application program.**

Remote login



TELNET server transforms the characters back to the form understandable by the OS

TELNET client transforms the characters to NVT form (Network Virtual Terminal characters), and sends them to local TCP/IP stack.

Heterogeneous platforms

OS accepts characters but doesn't interpret them. Instead it sends them to the TELNET client

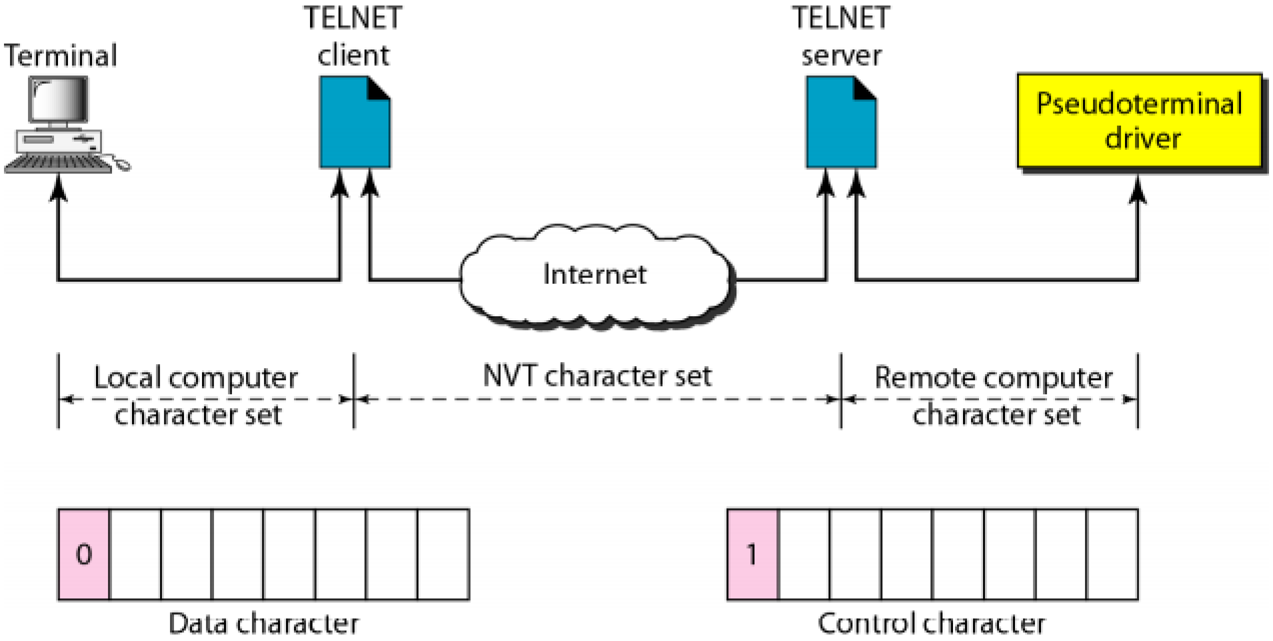
Text travels in NVT form

Pseudoterminal "pretends" that the characters are coming from a terminal (the application is not aware that the inputs are coming from a remote terminal.)

Network Virtual Terminal(NVT)

- The **mechanism** to access a **remote computer** is **complex** because every **computer** and its **operating system** accept a special combination of **characters** as tokens.
- For **example,** the **end-of-file** token in a computer running the **DOS operating system** is **Ctrl+z**, while the **UNIX operating system** recognizes **Ctrl+d.**
- We are dealing with **heterogeneous systems**. If we want to access any **remote computer** in the world, we must first **know** what **type of computer** we will be **connected to**, and we must also **install** the specific **terminal emulator** used by **that computer.**
- **TELNET** solves this problem by defining a **universal interface** called the **network virtual terminal (NVT) character set.**
- Via this interface, the **client TELNET translates characters** (data or commands) that come from the **local terminal** into **NVT form** and **delivers** them to the **network.**
- The **server TELNET**, on the other hand, **translates data** and **commands** from **NVT form** into the **form acceptable** by the **remote computer**.

Network Virtual Terminal



NVT Character Set

- **NVT** uses **two sets of characters**, one for **data** and the other for **control,** both are **8-bit bytes**.
- For **data, NVT** is an **8-bit character set** in which the **7 lowest-order bits** are the same as **ASCII** and the **highest-order bit** is **0**.
- To send **control characters** between computers (from client to server or vice versa), **NVT uses** an **8-bit character set** in which the **highest-order bit** is **set to 1.**

Embedding

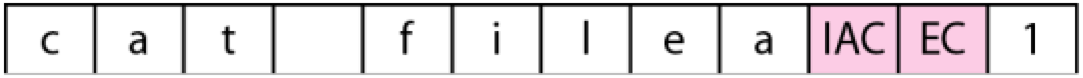
- **TELNET** uses only **one TCP connection**.
- The **server uses** the **well-known port 23**, and the **client uses** an **ephemeral port.**
- The **same connection** is **used** for **sending both data and control characters.**
- **TELNET** accomplishes this by **embedding the control characters in the data stream.**
- However, to **distinguish data from control characters**, each sequence of **control characters is preceded by a special control character** called ***interpret as control (IAC).***
- For **example,** imagine a user wants a **server to display a file** *(file1) on a **remote server.***
- One can type ***cat file1***

Embedding

- However, suppose the **name** of the **file** has been **mistyped *(filea instead of file1).***
- *The* **user** uses the **backspace key** to **correct** this situation.

  cat filea<backspace>1

- However, in the **default implementation** of **TELNET**, the **user cannot edit locally;** the **editing is done at the remote server**.
- The **backspace character** is **translated into two remote characters (IAC EC),** which are **embedded in the data** and sent to the **remote server.**
- What is sent to **the server** is shown below



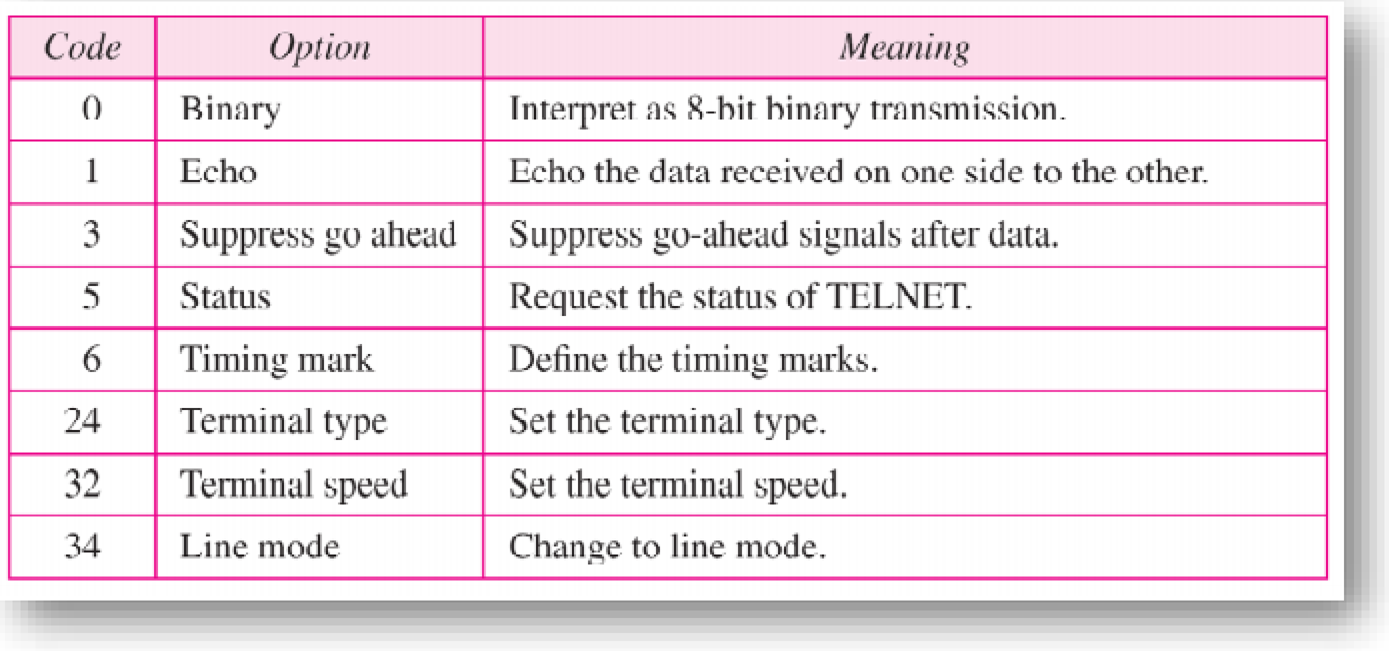
c	a	t		f	i	l	e	a	IAC	EC	1

Typed at the remote terminal

- **EC:** Erase Character

Options

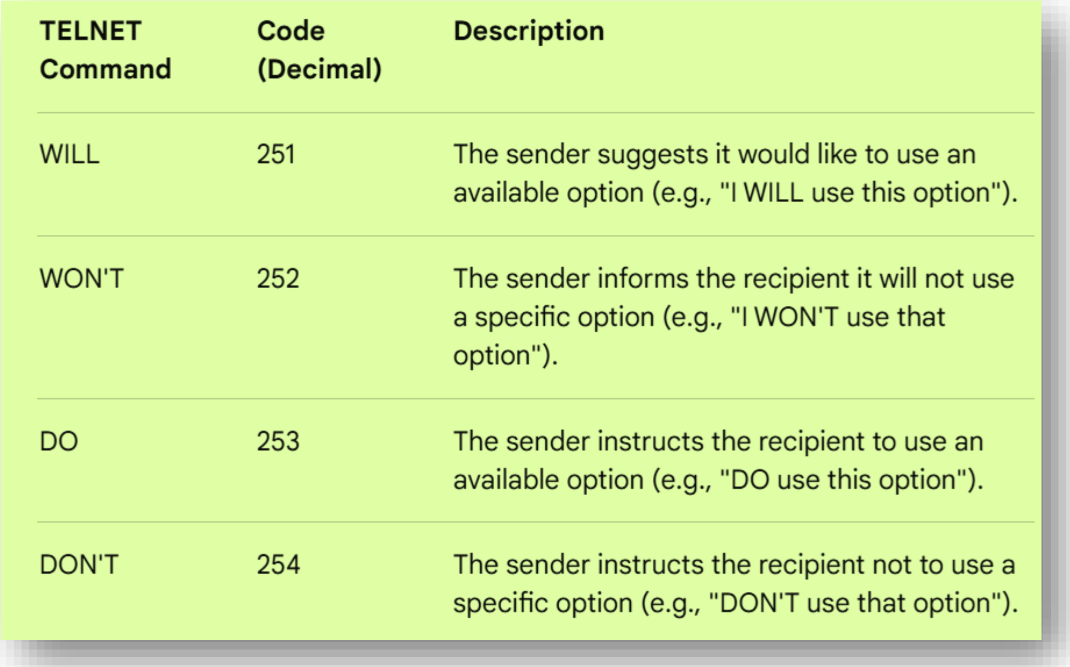
- **TELNET** lets the **client** and **server** negotiate **options** before or during the use of the service.
- **Options** are **extra features** available to a **user** with a more sophisticated terminal.



Code	*Option*	*Meaning*
0	Binary	Interpret as 8-bit binary transmission.
1	Echo	Echo the data received on one side to the other.
3	Suppress go ahead	Suppress go-ahead signals after data.
5	Status	Request the status of TELNET.
6	Timing mark	Define the timing marks.
24	Terminal type	Set the terminal type.
32	Terminal speed	Set the terminal speed.
34	Line mode	Change to line mode.

Option Negotiation

- To **use** any of the **options** mentioned in the **previous section** first requires **option negotiation** between the **client** and the **server.**
- Four **control characters** are used for this **purpose**; these are shown in **Table** below.



TELNET Command	Code (Decimal)	Description
WILL	251	The sender suggests it would like to use an available option (e.g., "I WILL use this option").
WON'T	252	The sender informs the recipient it will not use a specific option (e.g., "I WON'T use that option").
DO	253	The sender instructs the recipient to use an available option (e.g., "DO use this option").
DON'T	254	The sender instructs the recipient not to use a specific option (e.g., "DON'T use that option").

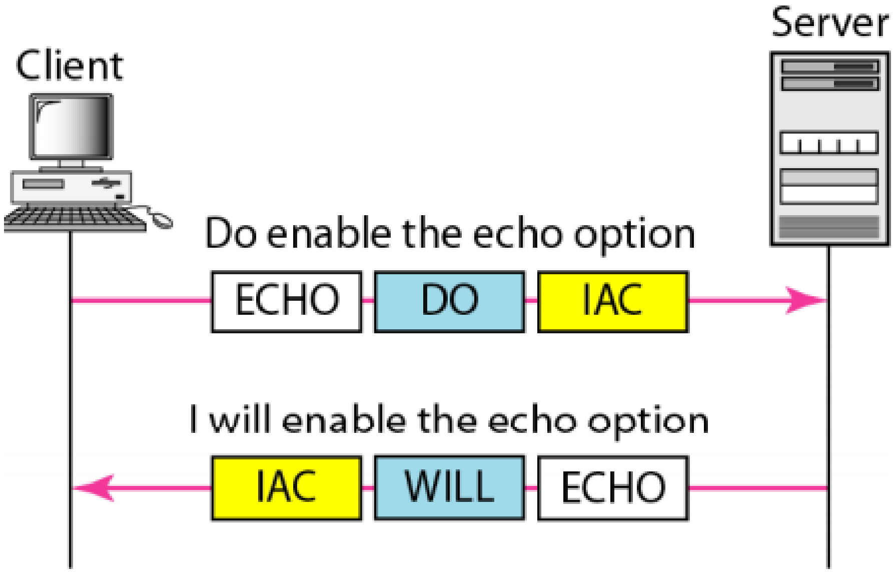
- **Format** of **option negotiation command**:

<IAC> <{WILL|WON’T|DO|DON’T}> <option code>

Example of Option Negotiation

- The **client** wants the **server** to **echo each character** sent to the **server.**
- In other words, when a **character** is **typed** at the user **keyboard terminal**, it goes to the **server** and is **sent back** to the **screen** of the **user.**
- The **echo option** is **enabled by the server** because it is the **server** that sends the characters back to the **user terminal.**
- Therefore, the **client** should ***request from the server*** *the* ***enabling*** of the **option using DO.**
- The **request consists of three characters**: IAC, DO, and ECHO.
- The **server accepts the request** and **enables** the **option.**
- **Server** informs the **client** by sending the **three-character approval**: IAC, WILL, and ECHO.

Option Negotiation



Mode of Operation

- Most **TELNET** implementations operate in one of **three modes**:

    1. ***Default mode,***
    2. ***Character mode,***
    3. ***Line mode.***

1. Default Mode

- The **default mode** is used if no other modes are invoked through **option negotiation.**
- In this mode, the **echoing** is **done** by the **client.**
- The **user types a character**, and the **client echoes** the **character on the screen** but does **not send it until** a **whole line** is **completed.**

Mode of Operation

2. Character Mode

- In the **character mode**, each **character typed** is **sent** by the **client** to the **server**.
- In this mode the **echoing** of the **character** **can be delayed** if the **transmission time is long** (such as in a satellite connection).
- In telnet **Character mode**, only **one character** is **transmitted** at a time.
- The **server** will **acknowledge** the **receipt** of **character** by **echoing it back** to the **TELNET client.**
- The **TELNET client** will send back a **TCP ACK** packet to the **TELNET server** to inform the **receipt** of **echo** from **server.**
- It also **creates** **overhead (traffic)** for the network because **three TCP segments** **must be sent for each character of data**.

Mode of Operation

3. Line Mode

- A new mode has been proposed to **compensate** for the **deficiencies** of the **default mode** and the **character mode.**
- In this mode, called the **line mode**, **line editing** (echoing, character erasing, line erasing, and so on) is **done by the client**.
- The **client** then **sends** the **whole line to the server.**

Applications and Limitations

- **TELNET** protocol is **mostly used** by **network admin** to **access** and **manage** **network devices** **remotely.**
- It **helps** them **access** the **device** by **telnetting** to the **IP address** or **hostname** of a **remote device.**
- It allows **users** to access any **application** on a **remote computer** thus **helps** them to **establish** a **connection** to a **remote system.**
- **Telnet** is **suitable** for **private networks (LAN)** **only.**
- **Telnet** is **vulnerable to** **security attacks** as it **transfers** the data in **plain text .**