18CSC302J- Computer Networks

Unit-3

Syllabus

1. DNS- DNS in the Internet, DNS Resolution, DNS Messages
2. **TELNET – SSH**
3. FTP-TFTP
4. WWW Architecture, Documents
5. HTTP, HTTP Request and Reply,
6. DHCP Operation, DHCP Configuration
7. SMTP, POP3, IMAP, MIME

Learning Resources

1. Douglas E. Comer, Internetworking with TCP/IP, Principles, protocols, and architecture,Vol 1 5th Edition,2006 ISBN: 0131876716, ISBN: 978-0131876712

REMOTE LOGIN

TELNET & SSH

REMOTE LOGIN

- The main task of the Internet and its TCP/IP protocol suite is to provide services for users.
- There are some specific client/server programs, it would be impossible to write a specific client-server program for each demand.
- 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.
- Two of these application programs are: TELNET and SSH

TELNET

- TELNET is an abbreviation for **TErminaL NETwork**.
- It is the standard TCP/IP protocol for virtual terminal service as proposed by 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.
- **TELNET is a general-purpose client-server application program.**

Concepts

- ✔ Time-Sharing Environment
- ✔ Network Virtual Terminal (NVT)
- ✔ Embedding
- ✔ Options, Symmetry, and Suboption Negotiation
- ✔ Controlling the Server
- ✔ Out-of-Band Signaling
- ✔ Escape Character
- ✔ Modes of Operation
- ✔ User Interface
- ✔ Security Issue

1. Time-Sharing Environment

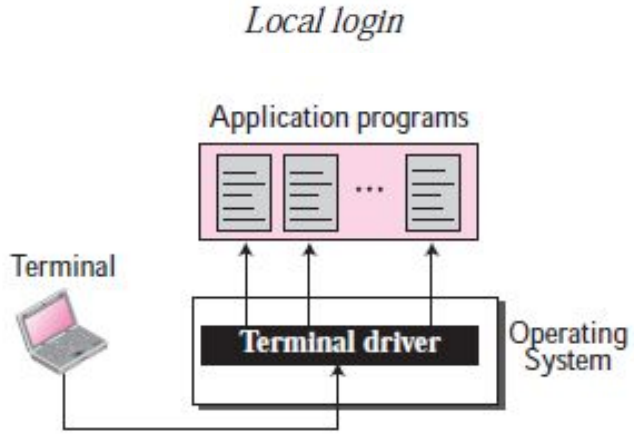
- TELNET was designed at a time when most operating systems, such as UNIX, were operating in a time-sharing environment.
- In such an environment, a large computer supports multiple users.
- The interaction between a user and the computer occurs through a terminal, which is usually a combination of keyboard, monitor, and mouse.
- Even a microcomputer can simulate a terminal with a terminal emulator.
- In a time-sharing environment, all of the processing must be done by the central computer.
- When a user types a character on the keyboard, the character is usually sent to the computer and echoed to the monitor.
- Time-sharing creates an environment in which each user has the illusion of a dedicated computer.
- The user can run a program, access the system resources, switch from one program to another, and so on.

Login

- In a time-sharing environment, users are part of the system with some right to access resources.
- Each authorized user has an identification and probably a password.
- The user identification defines the user as part of the system.
- To access the system, the user logs into the system with a user id or login name.
- The system also includes password checking to prevent an unauthorized user from accessing the resources
- Types Local login and Remote login

Local Login

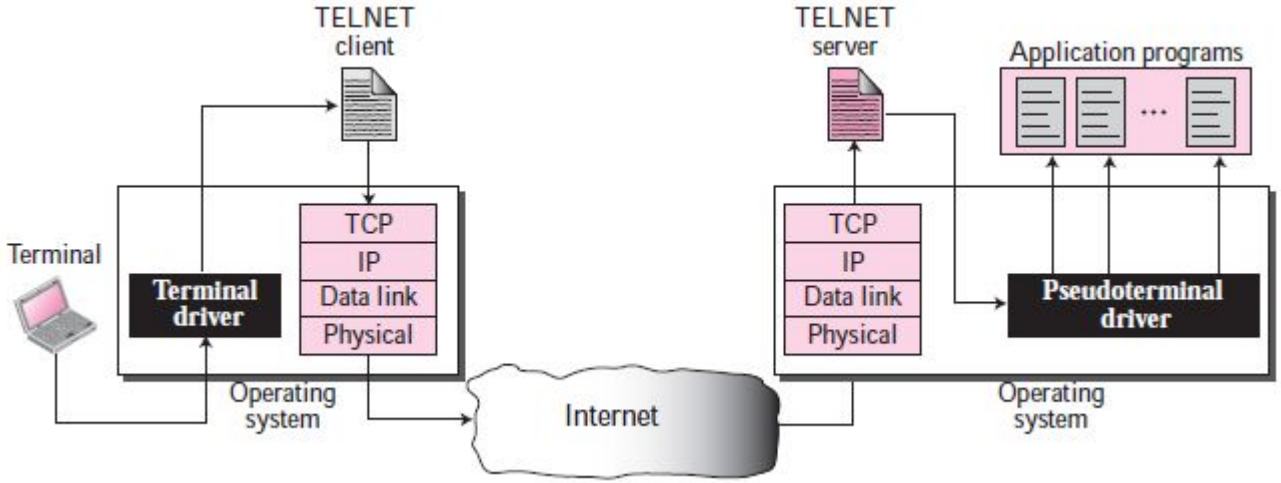
- **When a user logs into a local time-sharing system, it is called local login.**
- As a user types at a terminal or at a workstation running a terminal emulator, the keystrokes are accepted by the terminal driver.
- The terminal driver passes the characters to the operating system.
- The operating system, in turn, interprets the combination of characters and invokes the desired application program or utility



Remote Login

- **When a user wants to access an application program or utility located on a remote machine, he or she performs remote login.**
- Here the TELNET client and server programs come into use.
- 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 stack.

Remote login



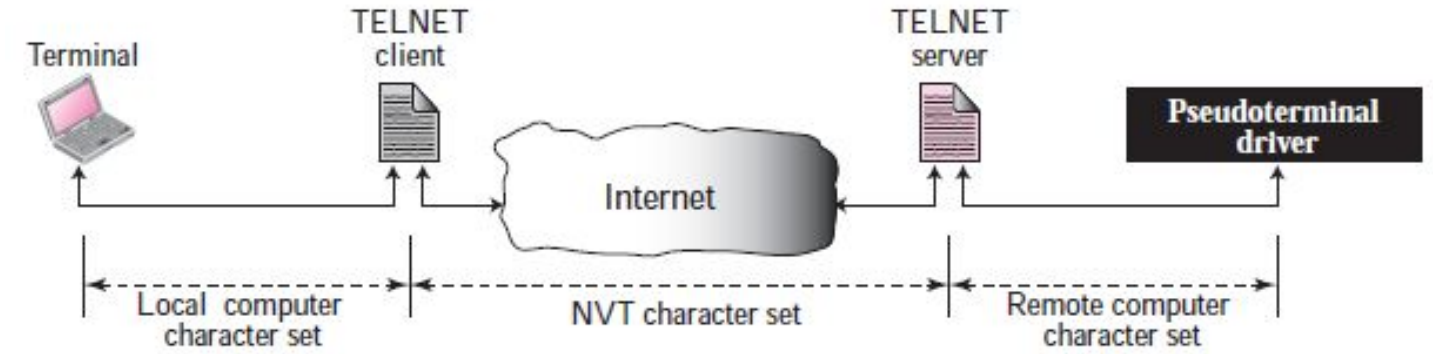
Remote Login-Working

- 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 OS 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.

2. Network Virtual Terminal (NVT)

- The mechanism to access a remote computer is complex.
- This is because every computer and its OS accepts a special combination of characters as tokens.
- For example, the end-of-file token in a computer running the DOS OS is Ctrl+z, while the UNIX OS recognizes Ctrl+d.
- When 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.

Concept of NVT



Concept of NVT

NVT Character Set

- NVT uses two sets of characters, one for data and one for control. Both are 8-bit bytes

Format of data and control characters

Data Characters

- For data, NVT normally uses what is called NVT ASCII.
- This is an 8-bit character set in which the seven lowest order bits are the same as US ASCII and the highest order bit is 0 (see Figure 20.4).
- Although it is possible to send an 8-bit ASCII (with the highest order bit set to be 0 or 1), this must first be agreed upon between the client and the server using option negotiation.

Control Characters

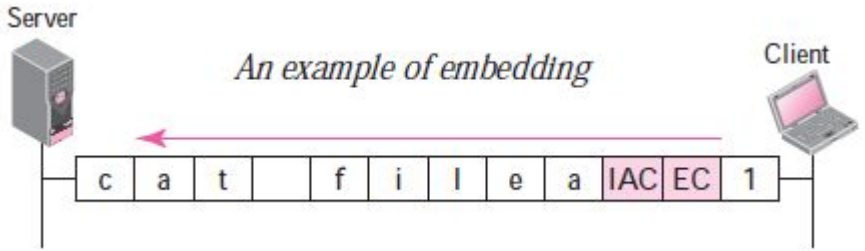
- To send control characters between computers (from client to server or vice versa), NVT uses an 8-bit character set in which the highest order bit is set to 1.

Some NVT control characters

Character	*Decimal*	*Binary*	*Meaning*
EOF	236	11101100	End of file
EOR	239	11101111	End of record
SE	240	11110000	Suboption end
NOP	241	11110001	No operation
DM	242	11110010	Data mark
BRK	243	11110011	Break
IP	244	11110100	Interrupt process
AO	245	11110101	Abort output
AYT	246	11110110	Are you there?
EC	247	11110111	Erase character
EL	248	11111000	Erase line
GA	249	11111001	Go ahead
SB	250	11111010	Suboption begin
WILL	251	11111011	Agreement to enable option
WONT	252	11111100	Refusal to enable option
DO	253	11111101	Approval to option request
DONT	254	11111110	Denial of option request
IAC	255	11111111	Interpret (the next character) as control

3. 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.
- She/he types: `cat file1`
- Where cat is a Unix command that displays the content of the file on the screen.
- Suppose if 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 is embedded in the data and sent to the remote server.


Server
An example of embedding
Client
c
a
t
f
i
l
e
a
IAC
EC
1

4. 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.
- Users with simpler terminals can use default features.
- Some control characters discussed previously are also used to define options.

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

The option descriptions are as follows:

❑ Binary

- This option allows the receiver to interpret every 8-bit character received, except IAC, as binary data.
- When IAC is received, the next character or characters are interpreted as commands.
- However, if two consecutive IAC characters are received, the first is discarded and the second is interpreted as data.

❑ Echo

- This option allows the server to echo data received from the client.
- This means that every character sent by the client to the sender will be echoed back to the
- screen of the client terminal.
- In this case, the user terminal usually does not echo characters when they are typed but waits until it receives them from the server.

❑ Suppress go-ahead

- This option suppresses the go-ahead (GA) character.

❑ Status

- This option allows the user or the process running on the client machine to get the status of the options being enabled at the server site.

- ❑ Timing mark
  - This option allows one party to issue a timing mark that indicates all previously received data has been processed.
- ❑ Terminal type.
  - This option allows the client to send its terminal type.
- ❑ Terminal speed
  - This option allows the client to send its terminal speed.
- ❑ Line mode
  - This option allows the client to switch to the line mode.

Option Negotiation

- To use any of the options mentioned in the previous slides first requires option negotiation between the client and the server. Four control characters are used for this purpose

NVT character set for option negotiation

Character	*Code*	*Meaning 1*	*Meaning 2*	*Meaning 3*
WILL	251	Offering to enable	Accepting to enable	
WONT	252	Rejecting to enable	Offering to disable	Accepting to disable
DO	253	Approving to enable	Requesting to enable	
DONT	254	Disapproving to enable	Approving to disable	Requesting to disable

- Two types-Enabling an Option and Disabling an Option

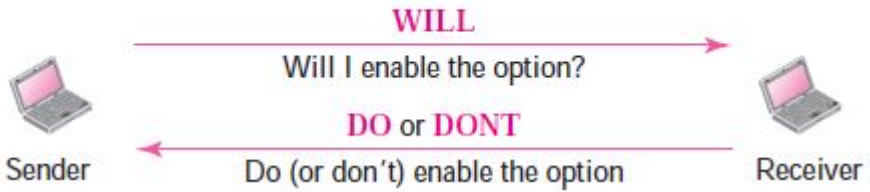
Enabling an Option

- Some options can only be enabled by the server, some only by the client, and some by both.
- An option is enabled either through an **offer or a request**.

a. Offer to Enable

- A party can offer to enable an option if it has the right to do so.
- The offering can be approved or disapproved by the other party.
- The offering party sends the WILL command, which means "Will I enable the option?" The other party sends either the DO command, which means "Please do," or the DONT command, which means "Please don't."

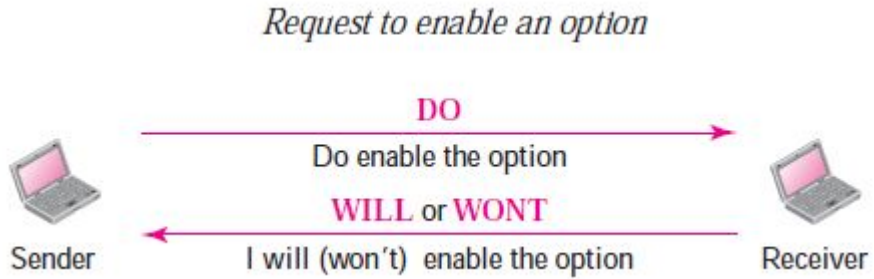
Offer to enable an option



Enabling an Option

b. Request to Enable

- A party can request from the other party the enabling of an option.
- The request can be accepted or refused by the other party.
- The requesting party sends the DO command, which means “Please do enable the option.”
- The other party sends either the WILL command, which means “I will,” or the WONT command, which means “I won’t.”

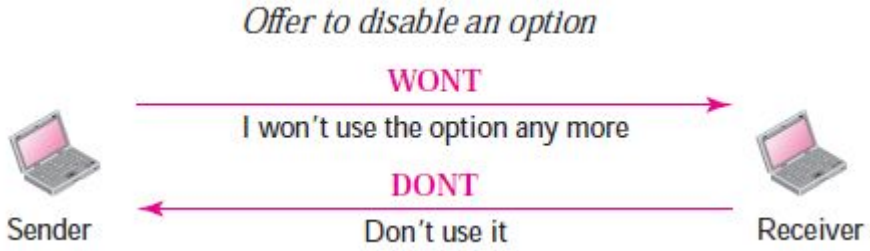


Disabling an Option

- An option that has been enabled can be disabled by one of the parties. An option is disabled either through an **offer or a request**.

a. Offer to Disable

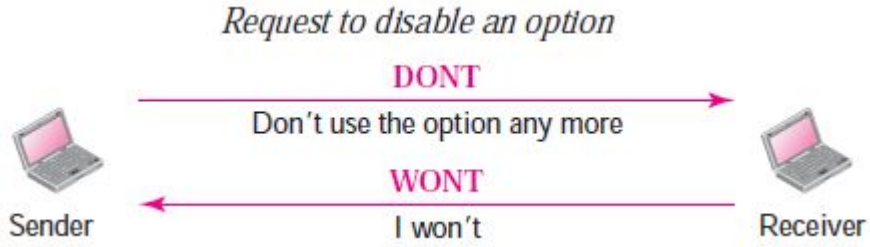
- A party can offer to disable an option.
- The other party must approve the offering; it cannot be disapproved.
- The offering party sends the WONT command, which means "I won't use this option any more."
- The answer must be the DONT command, which means "Don't use it anymore."



Disabling an Option

b. Request to Disable

- A party can request from another party the disabling of an option.
- The other party must accept the request; it cannot be rejected.
- The requesting party sends the DONT command, which means “Please don’t use this option anymore.”
- The answer must be the WONT command, which means “I won’t use it anymore.”



5. Symmetry

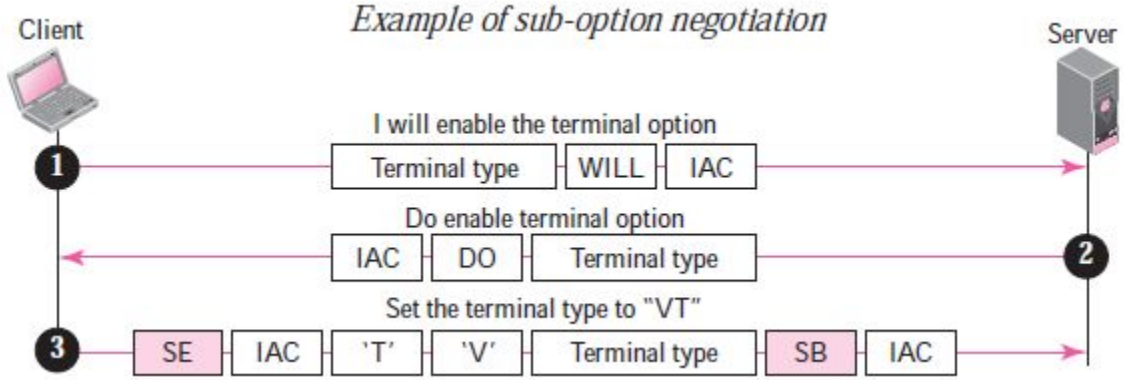
- One interesting feature of TELNET is its symmetric option negotiation in which the client and server are given equal opportunity.
- This means that, at the beginning of connection, it is assumed that both sides are using a default TELNET implementation with no options enabled.
- If one party wants an option enabled, it can offer or request.
- The other party has the right to approve the offer or reject the request if the party is not capable of using the option or does not want to use the option.
- This allows for the expansion of TELNET.
- A client or server can install a more sophisticated version of TELNET with more options.
- When it is connected to a party, it can offer or request these new options.
- If the other party also supports these options, the options can be enabled; otherwise, they are rejected.

6. Suboption Negotiation

- Some options require additional information.
- For example, to define the type or speed of a terminal, the negotiation includes a string or a number to define the type or speed.
- In either case, the two suboption characters are needed for suboption negotiation.
- For example, the type of the terminal is set by the client

NVT character set for suboption negotiation

Character	*Decimal*	*Binary*	*Meaning*
SE	240	11110000	Suboption end
SB	250	11111010	Suboption begin


Example of sub-option negotiation
Client
Server
I will enable the terminal option
Terminal type
WILL
IAC
1
Do enable terminal option
IAC
DO
Terminal type
2
Set the terminal type to "VT"
3
SE
IAC
'T'
'V'
Terminal type
SB
IAC

7. Controlling the Server

- Some control characters can be used to control the remote server.
- When an application program is running on the local computer, special characters are used to interrupt
- (abort) the program (for example, Ctrl+c), or erase the last character typed (for example, delete key or backspace key), and so on.
- However, when a program is running on a remote computer, these control characters are sent to the remote machine.
- The user still types the same sequences, but they are changed to special characters and sent to the server

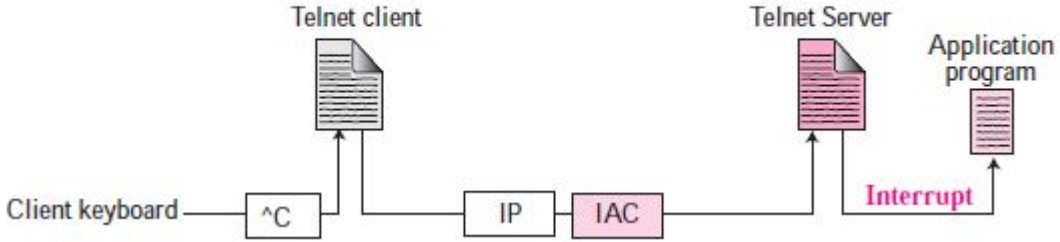
Characters used to control a program running on remote server

Character	*Decimal*	*Binary*	*Meaning*
IP	244	11110100	Interrupt process
AO	245	11110101	Abort output
AYT	246	11110110	Are you there?
EC	247	11110111	Erase the last character
EL	248	11111000	Erase line

- IP (interrupt process)
  - When a program is being run locally, the user can interrupt (abort) the program
  - TELNET defines the IP control character that is read and interpreted as the appropriate command for invoking the interrupting function in the remote machine.
- AO (abort output)
  - This is the same as IP, but it allows the process to continue without creating output.
- AYT (are you there?)
  - This control character is used to determine if the remote machine is still up and running, especially after a long silence from the server.
  - When this character is received, the server usually sends an audible or visual signal to confirm that it is running.
- EC (erase character)
  - When a user sends data from the keyboard to the local machine, the delete or backspace character can erase the last character typed.
  - To do the same in a remote machine, TELNET defines the EC control character.
- EL (erase line)
  - This is used to erase the current line in the remote host.

- For example, below figure shows how to interrupt a runaway application program at the server site.
- The user types Ctrl+c, but the TELNET client sends the combination of IAC and IP to the server

Example of interrupting an application program



8. Out-of-Band Signaling

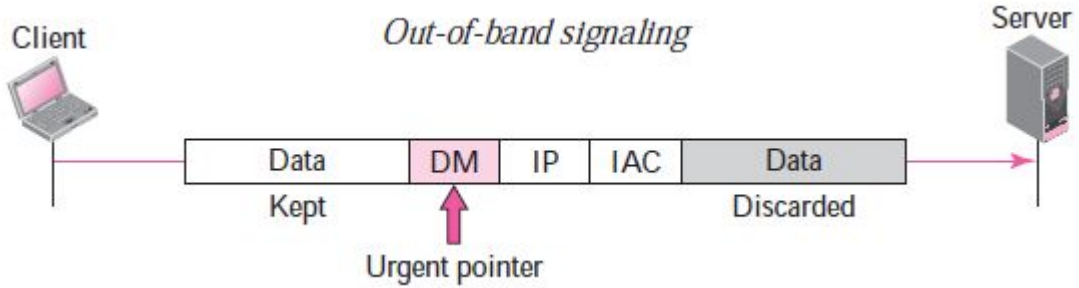
- To make control characters effective in special situations, TELNET uses out-of-band signaling.
- In out-of-band signaling, the control characters are preceded by IAC and are sent to the remote process.

- Imagine a situation in which an application program running at the server site has gone into an infinite loop and does not accept any more input data.
- The user wants to interrupt the application program, but the program does not read data from the buffer.
- The TCP at the server site has found that the buffer is full and has sent a segment specifying
- that the client window size should be zero.
- In other words, the TCP at the server site is announcing that no more regular traffic is accepted.
- To remedy such a situation, an urgent TCP segment should be sent from the client to the server.
- The urgent segment overrides the regular flow-control mechanism.
- Although TCP is not accepting normal segments, it must accept an urgent segment.

Out-of-Band Signaling

Example

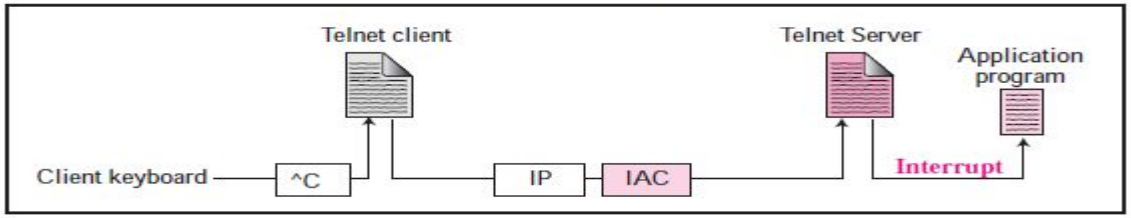
- When a TELNET process (client or server) wants to send an out-of-band sequence of characters to the other process (client or server), it embeds the sequence in the data stream and inserts a special character called a DM (data mark).
- However, to inform the other party, it creates a TCP segment with the urgent bit set and the urgent pointer pointing to the DM character.
- When the receiving process receives the data, it reads the data and discards any data preceding the control characters (IAC and IP, for example).
- When it reaches the DM character, the remaining data are handled normally.
- In other words, the DM character is used as a synchronization character that switches the receiving
- process from the urgent mode to the normal mode and resynchronizes the two ends



9. Escape Character

- A character typed by the user is normally sent to the server.
- However, sometimes the user wants characters interpreted by the client instead of the server.
- In this case, the user can use an escape character, normally Ctrl+] (shown as ^]).
- Below figure compares the interruption of an application program at the remote site with the interruption of the client process at the local site using the escape character.
- The TELNET prompt is displayed after this escape character.

Two different interruptions



a. Interrupting the application program

b. Interrupting the client

10. Modes of Operation

- Most TELNET implementations operate in one of three modes: default mode, character mode, or line mode.

a. 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 (or printer) but does not send it until a whole line is completed.
- After sending the whole line to the server, the client waits for the GA (go ahead) command from the server before accepting a new line from the user.
- The operation is half-duplex.
- Half-duplex operation is not efficient when the TCP connection itself is full-duplex, and so this mode is becoming obsolete.

Default Mode-Example

- The default mode to show the concept and its deficiencies even though it is almost obsolete today.
- The client and the server negotiate the terminal type and terminal speed and then the server checks the login and password of the user

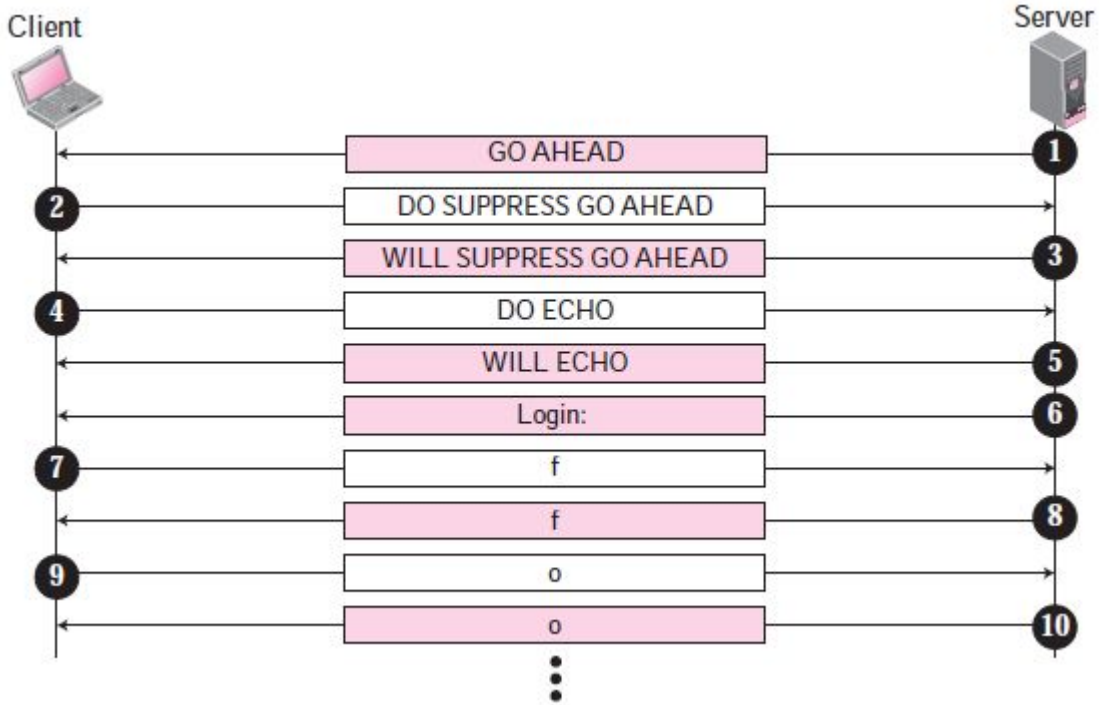
b. Character Mode

- In the character mode, each character typed is sent by the client to the server.
- The server normally echoes the character back to be displayed on the client screen.
- In this mode the echoing of the character can be delayed if the transmission time is long (such as in a satellite connection).
- It also creates overhead (traffic) for the network because three TCP segments must be sent for each character of data:

    1. The user enters a character that is sent to the server.
    2. The server acknowledges the received character and echoes the character back (in one segment).
    3. The client acknowledges the receipt of the echoed character.

Character Mode-Example

- Show how the client switches to the character mode.
- This requires that the client request the server to enable the SUPPRESS GO AHEAD and ECHO options



c. 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.
- Although the line mode looks like the default mode, it is not.
- The default mode operates in the half-duplex mode; the line mode is full-duplex with the client sending one line after another, without the need for an intervening GA (go ahead) character from the server.

11. User Interface

- The normal user does not use TELNET commands as defined above.
- Usually, the OS (UNIX, for example) defines an interface with user-friendly commands.
- The interface is responsible for translating the user-friendly commands to the previously defined commands in the protocol.

Examples of interface commands

Command	*Meaning*	*Command*	*Meaning*
open	Connect to a remote computer	set	Set the operating parameters
close	Close the connection	status	Display the status information
display	Show the operating parameters	send	Send special characters
mode	Change to line or character mode	quit	Exit TELNET

12. Security Issue

- TELNET suffers from security problems.
- Although TELNET requires a login name and password (when exchanging text), often this is not enough.
- A microcomputer connected to a broadcast LAN can easily eavesdrop using snooper software and capture a login name and the corresponding password (even if it is encrypted).
- Need proper authentication and security.

TELNET-SUMMARY

Concepts

- Time-Sharing Environment
  - Login
    - Local login
    - Remote login
- Network Virtual Terminal (NVT)
  - NVT Character sets
    - Data Characters
    - Control Characters
- Embedding
- Options
  - Option Negotiation
  - Enabling an Option
    - Offer to Enable
    - Request to Enable
  - Disabling an Option
    - Offer to Disable
    - Request to Disable
- Symmetry
- Suboption Negotiation
- Controlling the Server
- Out-of-Band Signaling
- Escape Character
- Modes of Operation
  - Default mode
  - Character mode
  - Line mode
- User Interface
- Security Issue

SECURE SHELL (SSH)

- Another popular remote login application program is Secure Shell (SSH).
- SSH, like TELNET, uses TCP as the underlying transport protocol, but SSH is more secure and provides more services than TELNET.

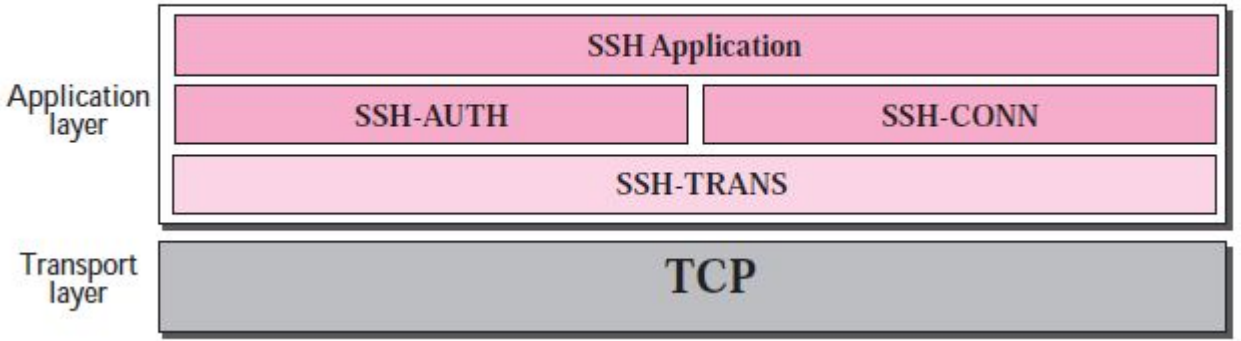
Versions

- There are two versions of SSH: SSH-1 and SSH-2, which are totally incompatible. The first version, SSH-1 is now deprecated because of security flaws in it.
- We discuss only SSH-2.

Components

- SSH is a proposed application-layer protocol with four components

Components of SSH



❑ SSH Transport-Layer Protocol (SSH-TRANS)

- Since TCP is not a secured transport layer protocol, SSH first uses a protocol that creates a secured channel on the top of TCP.
- This new layer is an independent protocol referred to as SSH-TRANS.
- When the software implementing this protocol is called, the client and server first use the TCP protocol to establish an insecure proconnection.
- The services provided by this protocol are:
    - 1. Privacy or confidentiality of the message exchanged.
    - 2. Data integrity, which means that it is guaranteed that the messages exchanged
    - between the client and server are not changed by an intruder.
    - 3. Server authentication, which means that the client is now sure that the server is the
    - one that it claims to be.
    - 4. Compression of the messages that improve the efficiency of the system and makes
    - attack more difficult.

❑ SSH Authentication Protocol (SSH-AUTH)

- After a secure channel is established between the client and the server and the server is authenticated for the client, SSH can call another software that can authenticate the client for the server.

❑ SSH Connection Protocol (SSH-CONN)

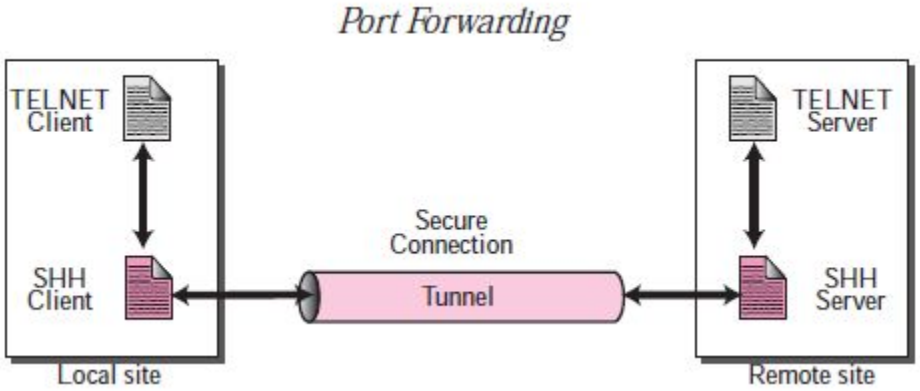
- After the secured channel is established and both server and client are authenticated for each other,
- SHH can call a piece of software that implements the third protocol, SSHCONN.
- One of the services provided by the SSH-CONN protocol is to do multiplexing.
- SSH-CONN takes the secure channel established by the two previous protocols and lets the client create multiple logical channels over it.

❑ SSH Applications

- After the connection phase is completed, SSH allows several application programs to use the connection.
- Each application can create a logical channel as described above and then benefit from the secured connection.
- In other words, remote login is one of the services that can use the SSH-CONN protocols; other applications, such as a file transfer application can use one of the logical channels for this purpose.

Port Forwarding

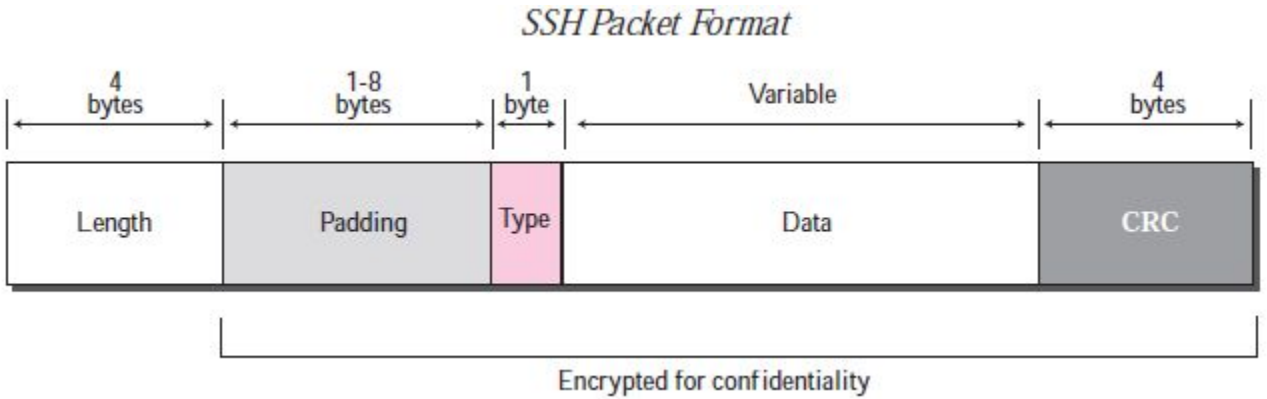
- One of the interesting services provided by the SSH protocol is to provide port forwarding.
- We can use the secured channels available in SSH to access an application program that does not provide security services.
- Application such as TELNET and SMTP can use the services of SSH using port forwarding mechanism.
- SSH port forwarding mechanism creates a tunnel through which the messages belonging to other protocol can travel.
- For this reason, this mechanism is sometimes referred to as SSH tunneling.
- We can change a direct, but insecure, connection between the TELNET client and the TELNET server by port forwarding.
- The TELNET client can use the SHH client on the local site to make a secure connection with the SSH server on the remote site.
- Any request from the TELNET client to the TELNET server is carried through the tunnel
- provided by the SSH client and server.
- Any response from the TELNET server to the TELNET client is also carried through the tunnel provided by the SSH client and server.


Port Forwarding
TELNET
Client
SHH
Client
Local site
Secure
Connection
Tunnel
TELNET
Server
SHH
Server
Remote site

Format of the SSH Packets

The following is the brief description of each field:

- ❑ Length
  - This 4-byte field defines the length of the packet including the type, the data, and the CRC field, but not the padding and the length field.
- ❑ Padding
  - One to eight bytes of padding is added to the packet to make the attack on the security provision more difficult.
- ❑ Type
  - This one-byte field defines the type of the packet used by SSH protocols.
- ❑ Data
  - This field is of variable length.
  - The length of the data can be found by deducting the five bytes from the value of the length field.
- ❑ CRC
  - The cyclic redundancy check filed is used for error detection (see Appendix D).


SSH Packet Format
4 bytes
1-8 bytes
1 byte
Variable
4 bytes
Length
Padding
Type
Data
CRC
Encrypted for confidentiality

SSH-SUMMARY

Versions

Components

- **SSH Transport-Layer Protocol (SSH-TRANS)**
- **SSH Authentication Protocol (SSH-AUTH)**
- **SSH Connection Protocol (SSH-CONN)**
- **SSH Applications**

Port Forwarding

Format of the SSH Packets

Example:

Application PuTTY Terminal