CS321: Computer Networks

FTP, TELNET, SSH

Dr. Manas Khatua
Assistant Professor
Dept. of CSE
IIT Jodhpur
E-mail: manaskhatua@iitj.ac.in

FTP

- File Transfer Protocol (FTP) is the standard protocol provided by *TCP/IP*

- In a typical FTP session, the user is sitting in front of one host (the local host) and wants to transfer files to or from a remote host

- user accessing the remote account
  - user must provide a user identification and a password.

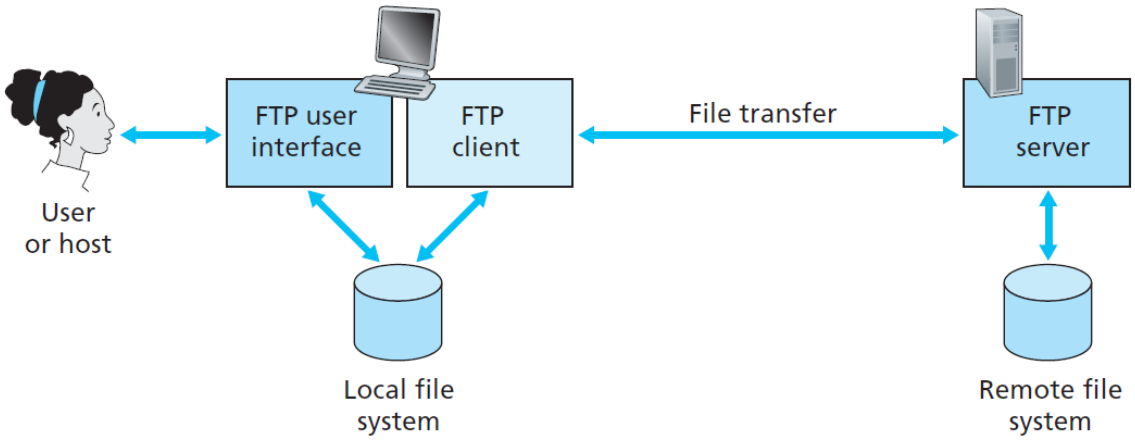


Figure 2.14 ♦ FTP moves files between local and remote file systems

HTTP vs FTP

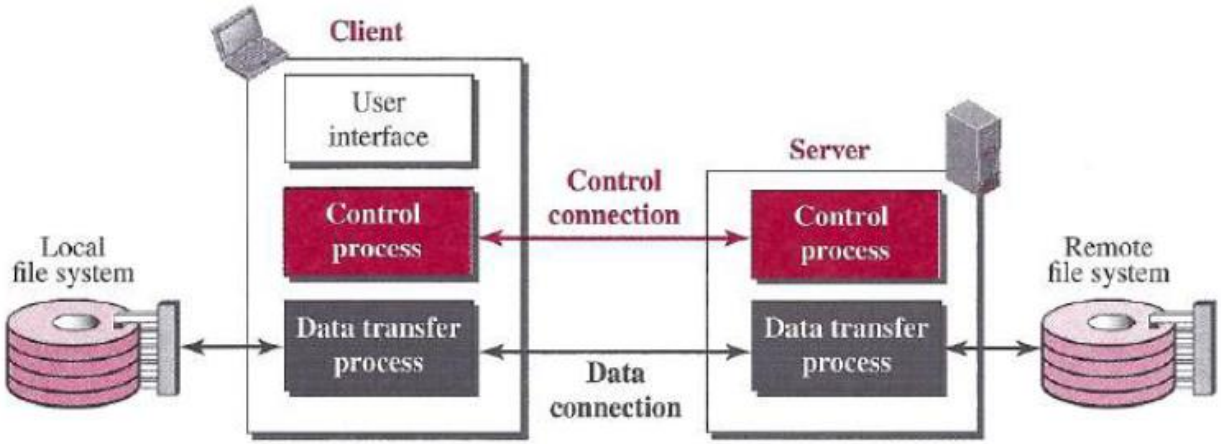
- HTTP and FTP
  - Are both application layer protocols
  - are both file transfer protocols
  - they both run on top of TCP

  - FTP uses two parallel TCP connections to transfer a file, a *control connection* and a *data connection*.
  - Throughout a session, the FTP server must maintain state about the user.

  - FTP is said to send its control information out-of-band
  - HTTP is said to send its control information in-band

Basic Model of FTP

- FTP must address the following:
  - two systems may use different file name conventions
  - two systems may have different ways to represent data
  - two systems may have different directory structures



- The client has three components: the user interface, the client control process, and the client data transfer process.
- The server has two components: the server control process and the server data transfer process.
- There are two connects: control and data connection

Cont…

- The two connections in FTP have different lifetimes.
  - The control connection remains connected during the entire interactive FTP session.
  - The data connection is opened and then closed for each file transfer activity

- FTP server uses two well-known TCP ports:
  - port 21 is used for the control connection,
  - port 20 is used for the data connection.

- Benefits for having two separate connections:
  - No need for complicated framing on the control connection.
  - Handling special cases, like cancelling a data connection, is simpler.
  - You can have multiple data transfers running at a time without having to establish multiple control connections.
  - It enables a trick, known as FXP, that can allow you to make two FTP servers exchange data directly between each other.

Control Connection

- Control communication is achieved through commands and responses.
- During this control connection, commands are sent from the client to the server and responses are sent from the server to the client.
- Commands are in the form of ASCII uppercase, which mayor may not be followed by an argument.

Table 26.4 *Some FTP commands*

Command	*Argument(s)*	*Description*
ABOR		Abort the previous command
CDUP		Change to parent directory
CWD	Directory name	Change to another directory
DELE	File name	Delete a file
LIST	Directory name	List subdirectories or files
MKD	Directory name	Create a new directory
PASS	User password	Password

Cont…

- Every FTP command generates at least one response

- A response has two parts:
  - Three-digit number : defines the code
  - Text : defines needed parameters or further explanations

Table 26.5 *Some responses in FTP*

Code	*Description*	*Code*	*Description*
125	Data connection open	250	Request file action OK
150	File status OK	331	User name OK; password is needed
200	Command OK	425	Cannot open data connection
220	Service ready	450	File action not taken; file not available
221	Service closing	452	Action aborted; insufficient storage

Data Connection

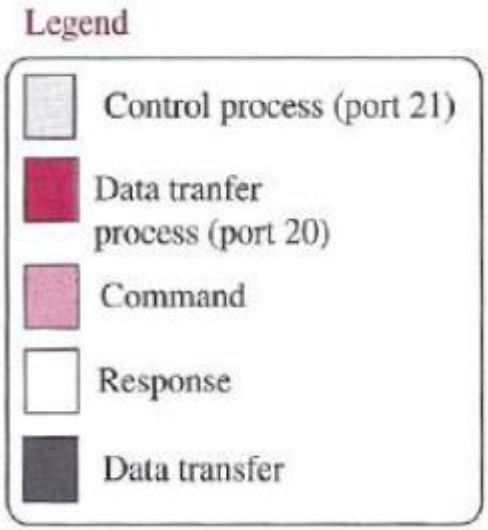
- the creation of a data connection is different from the control connection.

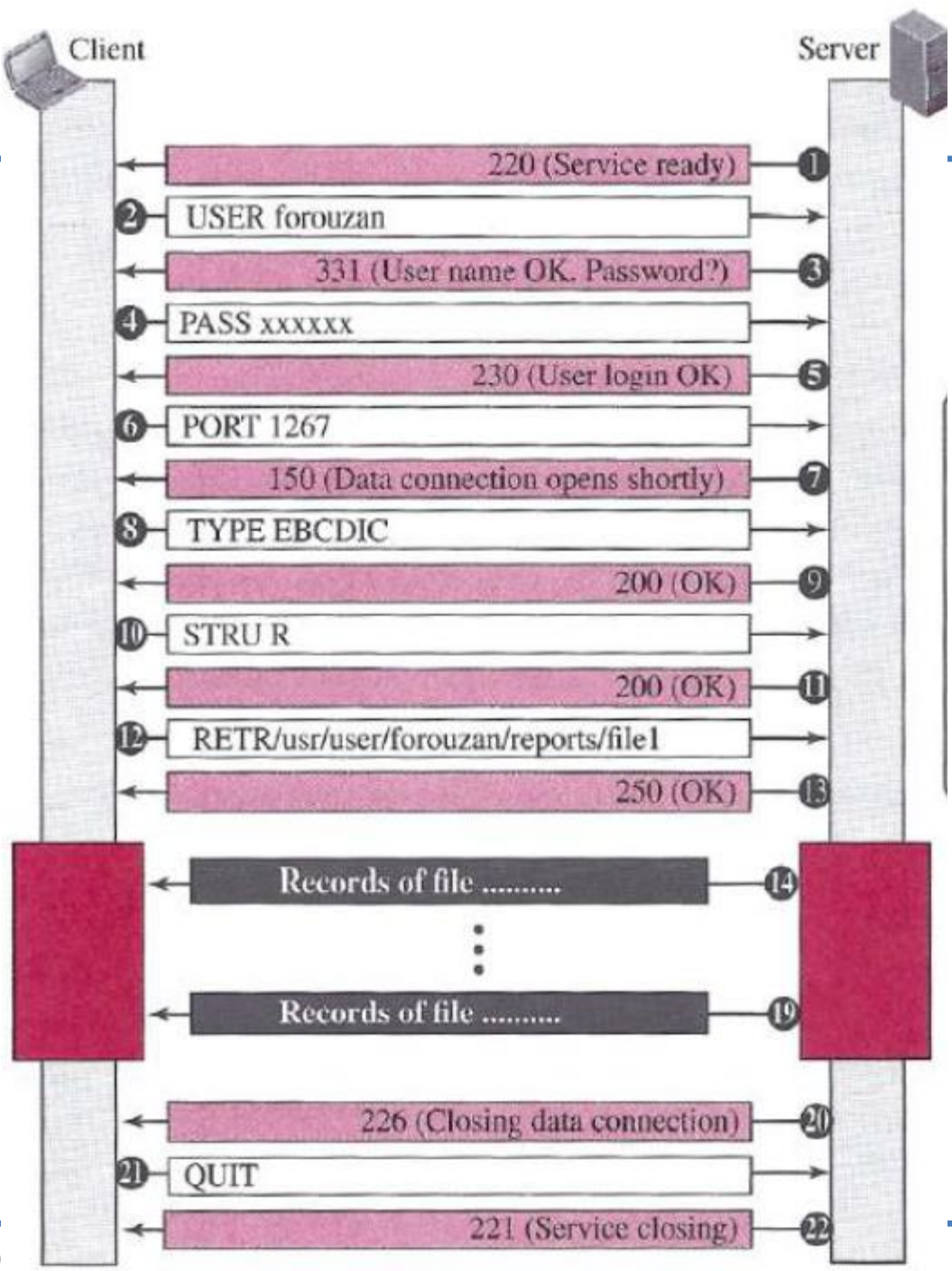
- Data connection steps:
  - The client, not the server, issues a passive open using an ephemeral port (>1023).
  - Using the PORT command the client sends this port number to the server.
  - The server receives the port number and issues an active open using the well-known port 20 and the received ephemeral port number.

Communication over Data Connection

- We prepare for data transmission through the control connection.
- The heterogeneity problem is resolved by defining three attributes of communication:
  - file type: *ASCII*, *EBCDIC*, or *image* file.
  - data structure: *file, record,* or *page* structure
  - transmission mode: *stream, block,* or *compressed* mode
- The file structure format (used by default) has no structure. It is a continuous stream of bytes.
- In the record structure, the file is divided into *records.* This can be used only with text files.
- In the page structure, the file is divided into pages, with each page having a page number and a page header.

Example


Legend
Control process (port 21)
Data tranfer process (port 20)
Command
Response
Data transfer


Client
Server
220 (Service ready)
1
2
USER forouzan
331 (User name OK. Password?)
3
4
PASS xxxxxx
230 (User login OK)
5
6
PORT 1267
150 (Data connection opens shortly)
7
8
TYPE EBCDIC
200 (OK)
9
10
STRU R
200 (OK)
11
12
RETR/usr/user/forouzan/reports/file1
250 (OK)
13
Records of file .....
14
Records of file .....
19
226 (Closing data connection)
20
21
QUIT
221 (Service closing)
22

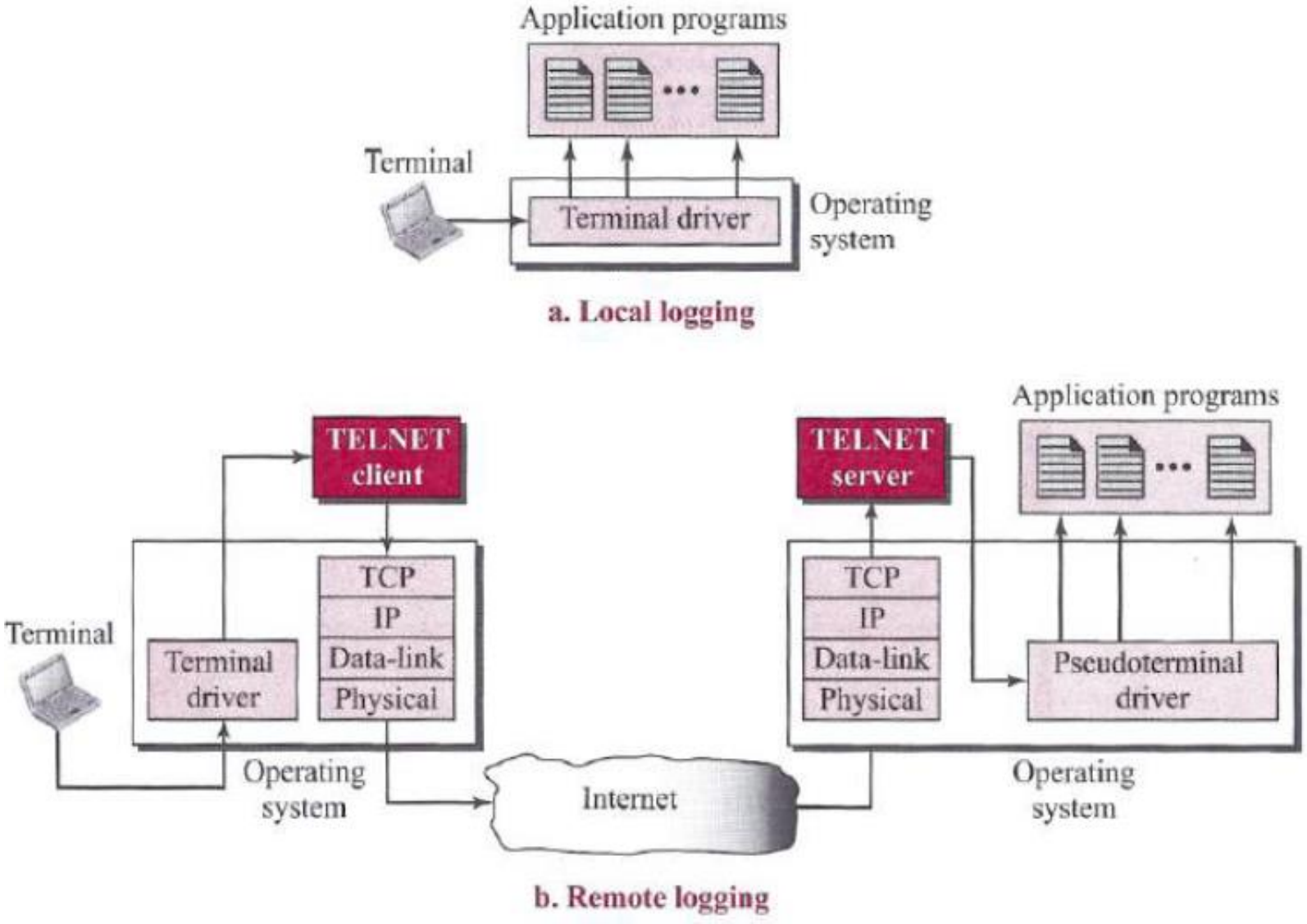
Security for FTP

- The FTP protocol was designed when security was not a big issue.
- Although FTP requires a password, the password is sent in plaintext (unencrypted)
- To be secure, one can add a Secure Socket Layer between the FTP application layer and the TCP layer.
- In this case FTP is called SSL-FTP

TELNET

- Many cases we need to have some generic client/server programs that allow a user on the client site to log into the computer at the server site and use the services available there.
  - E.g., Java Compiler is installed in server. No need in client PC.
- We refer to these generic client/server pairs as *remote logging* applications.
- One of the original remote logging protocols is TELNET, which is an abbreviation for *TErminaL NETwork.*
- It is vulnerable to hacking because it sends all data including the password in plaintext (not encrypted).
- TELNET is almost replaced by Secured Shell (SSH) because of its vulnerability

Local versus Remote Logging

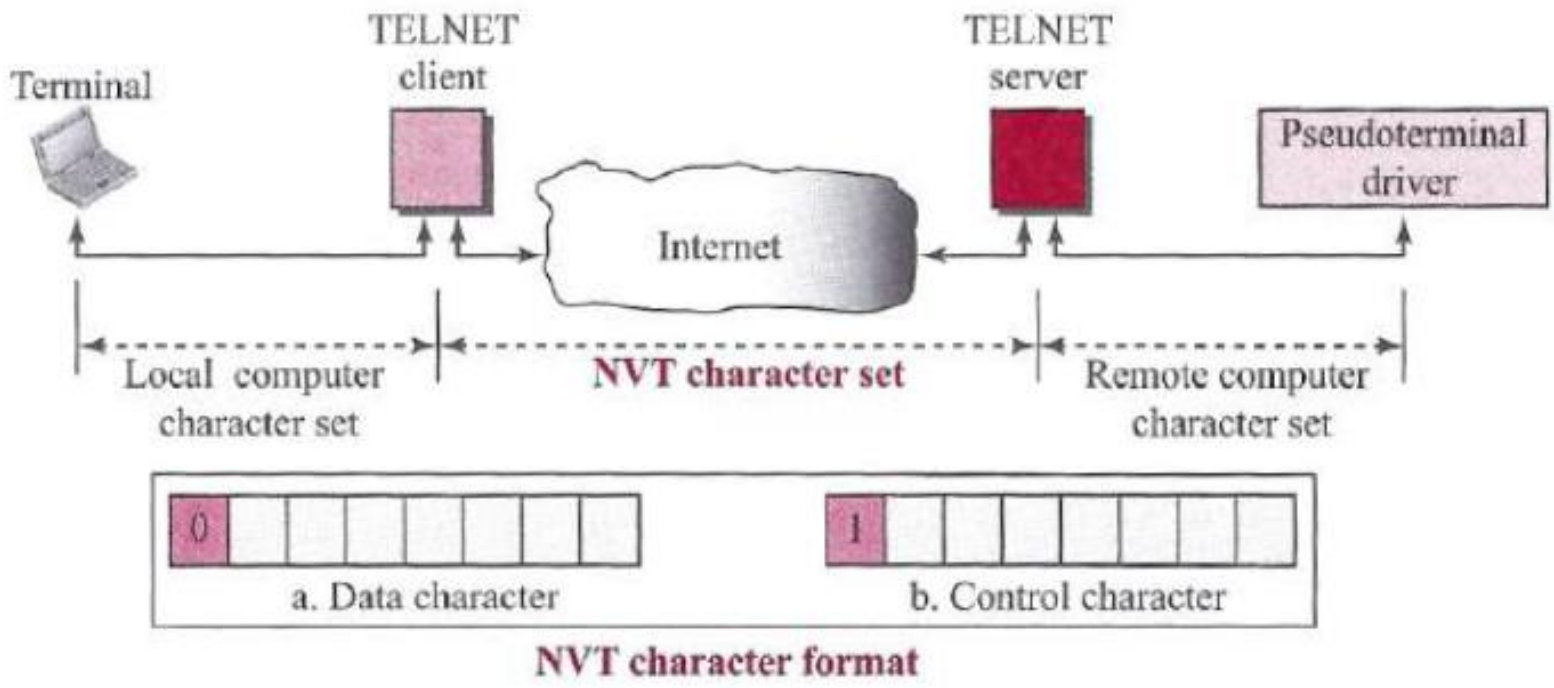


Cont…

- 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 into a universal character set called *Network Virtual Terminal* (NVT) characters and delivers them to the local TCPI/IP stack.
- The commands or text, in NVT form, travel through the Internet and arrive at the TCPI/IP stack at the remote machine.
- However, the characters cannot be passed directly to the operating system in the remote machine.
- The characters are sent to a *pseudoterminal driver (*which pretends that the characters are coming from a terminal to the OS in remote machine).
- The operating system then passes the characters to the appropriate application program.

Cont…

- We are dealing with heterogeneous systems.
- TELNET solves this problem of heterogeneity by defining a universal interface called the *Network Virtual Terminal (NVT)* character set.
- NVT uses two sets of characters, one for data and one for control. Both are 8-bit bytes.



NVT character format

Options & Unser Interface

- Options are extra features available to a user with a more sophisticated terminal.

  E.g., -4: IPv4; -6: IPv6; -E: disable escape char

- The operating system (UNIX, for example) defines an interface with user-friendly commands.

Table 26.11 *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

SECURE SHELL (SSH)

- SSH is a secure application program
-
- Applications
  - SSH for Remote Logging (PuTTy)
  - SSH for File Transfer (sftp)
  - SSH for Secure Copy (scp)

- It has three components:
  - *SSH Transport-Layer Protocol (SSH-TRANS)*
  - *SSH Authentication Protocol (SSH-AUTH)*
  - *SSH Connection Protocol (SSH-CONN)*

SSH-TRANS

- SSH Transport-Layer Protocol
  - It creates a secured channel on top of the TCP
  - the client and server first use the TCP protocol to establish an insecure connection.
  - Then they exchange several security parameters to establish a secure channel on top of the TCP.

- Few services provided by this protocol:
  - Privacy or confidentiality
  - Data integrity
  - Server authentication
  - Compression of the messages

SSH-AUTH

- *SSH Authentication Protocol*
  - Authenticate the client for the server.
  - Authentication starts with the client, which sends a request message to the server.
  - The request includes the user name, server name, the method of authentication, and the required data.
  - The server responds with either a success message, or a failed message,

SSH-CONN

- *SSH Connection Protocol*
  - It provides few more services such as multiplexing
  - SSH-CONN takes the secure channel established by the two previous protocols and lets the client create multiple logical channels over it.
  - Each channel can be used for a different purpose, such as remote logging, file transfer, and so on.

Thanks!