Vietnamese-German University

VGU

Computer Network

File Transfer Protocol

Presenter: *Nguyen Hoang Phuc* ***Vinh***

BIS2013

Agenda

1 - File Transfer Protocol

2 - Control channel

3 - Data channel

4 - Commands

5 - Status codes

6 - Transport binding

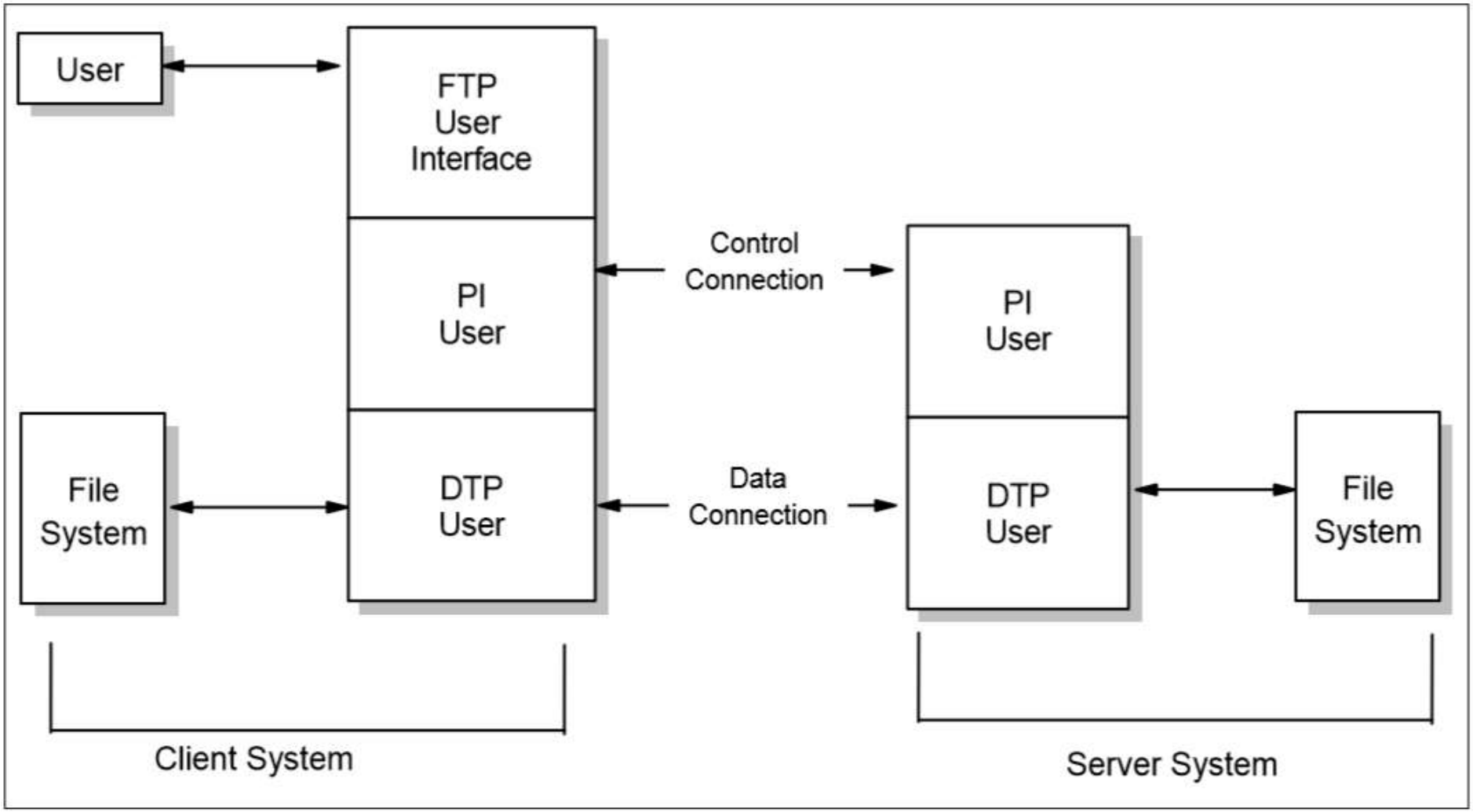
7 - Sercurity issues

1. File Transfer Protocol

- The File Transfer Protocol (FTP) is a standard network protocol.

- The objectives of FTP:
  1. Promote sharing of files (computer programs and/or data)
  2. Encourage indirect or implicit (via programs) use of remote computers
  3. Shield a user from variations in file storage systems among hosts
  4. Transfer data reliably and efficiently

1. FTP Overview



FTP model

2. Control Channel

- The FTP client initiates the first connection, referred to as the control connection, to well-known port 21
- This connection is used for all of the control commands a client user uses to log on to the server, manipulate files, and terminate a session.
- This is also the connection across which the FTP server will send messages to the client in response to these control commands.

3. Data Channel

- The second connection used by FTP is referred to as the data connection. Typically, the data connection is established on server port 20.
- It is across this connection that FTP transfers the data.
- FTP only opens a data connection when a client issues a command requiring a data transfer, such as a request to retrieve a file, or to view a list of the files available.

4. Commands

- When using FTP, the user performs some or all of the following operations
  - Connect to a remote host
  - Navigate and manipulate the directory structure.
  - List files available for transfer.
  - Define the transfer mode, transfer type, and data structure
  - Transfer data to or from the remote host
  - Disconnect from the remote host.

4. Commands

- Connect to a remote host: To execute a file transfer, the user begins by logging in to the remote host
  - **open**: Selects the remote host and initiates the login session.
  - **user**: Identifies the remote user ID.
  - **pass**: Authenticates the user.
  - **site**: Sends information to the foreign host that is used to provide services specific to that host.

4. Commands

- Navigate and manipulate the directory structure:
  - **cd**: Changes the directory on the remote host.
  - **lcd**: Changes the directory on the local host.
  - **ls**: Lists the contents of the remote directory. This command is intended to create output readable by human users.
  - **dir**: Lists the contents of the remote directory. Similar to the ls command, the list generated by dir is treated as data and requires the use of a data connection. This command is intended to create output readable by programs.

4. Commands

- Controlling how the data is transferred: The user has to decide on three aspects of the data handling

1. The way the bits will be moved from one place to another
2. The different representations of data on the system's architecture
3. The file structure in which the data is to be stored

- **mode**: Specifies whether the file is treated as having a record structure in a byte stream format
- **type**: Specifies the character sets used in translating and representing the data.
- **structure**: Specifies the structure of the file to be transferred.

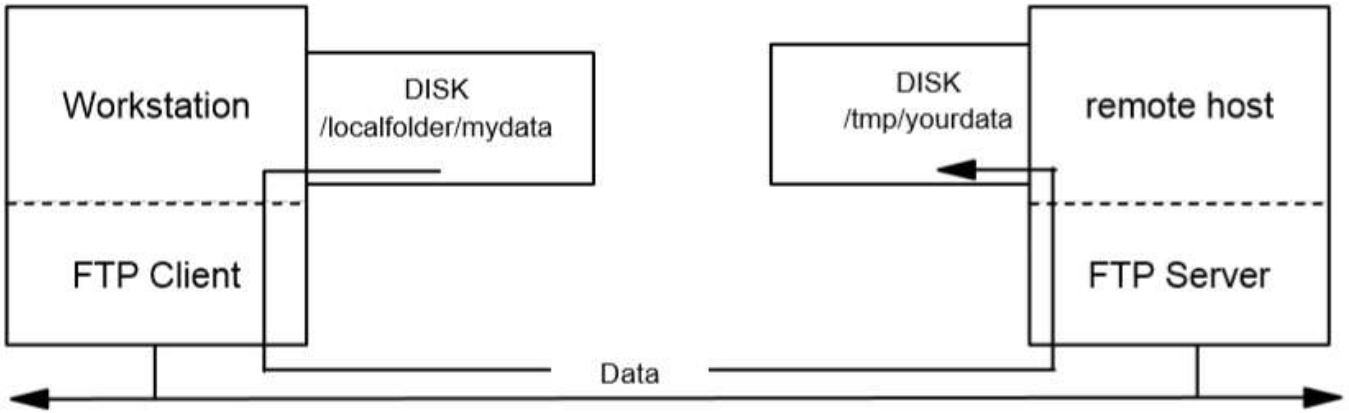
4. Commands

- Transferring files: commands can be used to copy files between FTP clients and servers
  - **get**: Copies a file from the remote host to the local host.
  - **mget**: Copies multiple files from the remote to the local host.
  - **put**: Copies a file from the local host to the remote host.
  - **mput**: Copies multiple files from the local host to the remote host.

4. Commands

- Terminating the FTP session: commands can be used to end an FTP session
  - **quit**: Disconnects from the remote host and terminates FTP. Some implementations use the BYE subcommand.
  - **close**: Disconnects from the remote host but leaves the FTP client running. An open command can be issued to establish a new control connection.

4. Commands



1)	Logon to the FTP server		`ftp`
			`user`
			`pass`
				`remote.host.yourcompany.com`
				`myUserId`
				`myPassword`

2)	Navigate to the correct remote and local folders		`cd` `l cd`	`/tmp` `/local folder`

3)	Specify the file attributes		`type`
			`mode`
			`stru`
				`l`
				`s`
				`f`

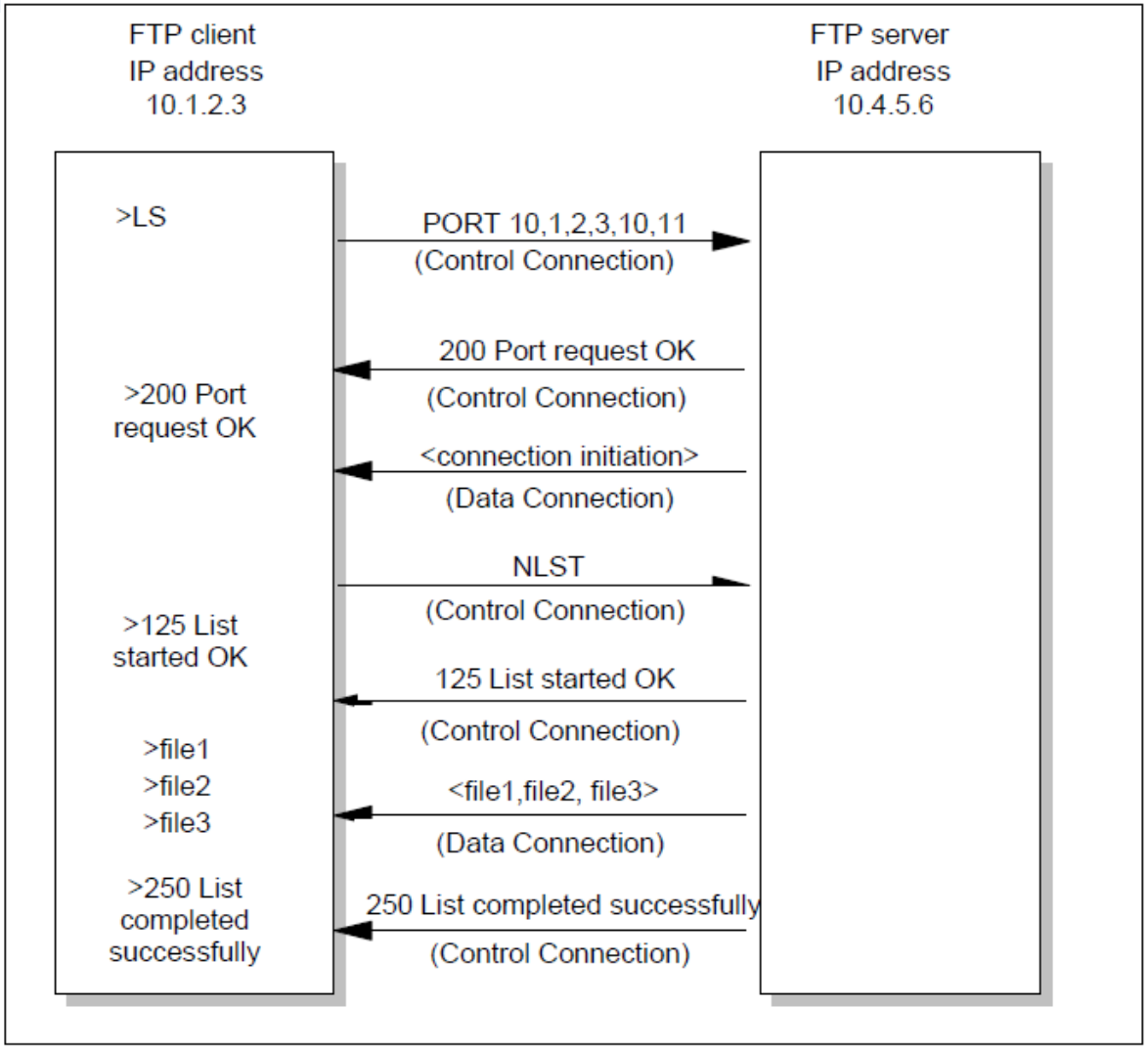
4)	Send the file		`put`	`mydata yourdata`
5)	Terminate the session		`quit`	

An example of FTP transfer

5. Status codes

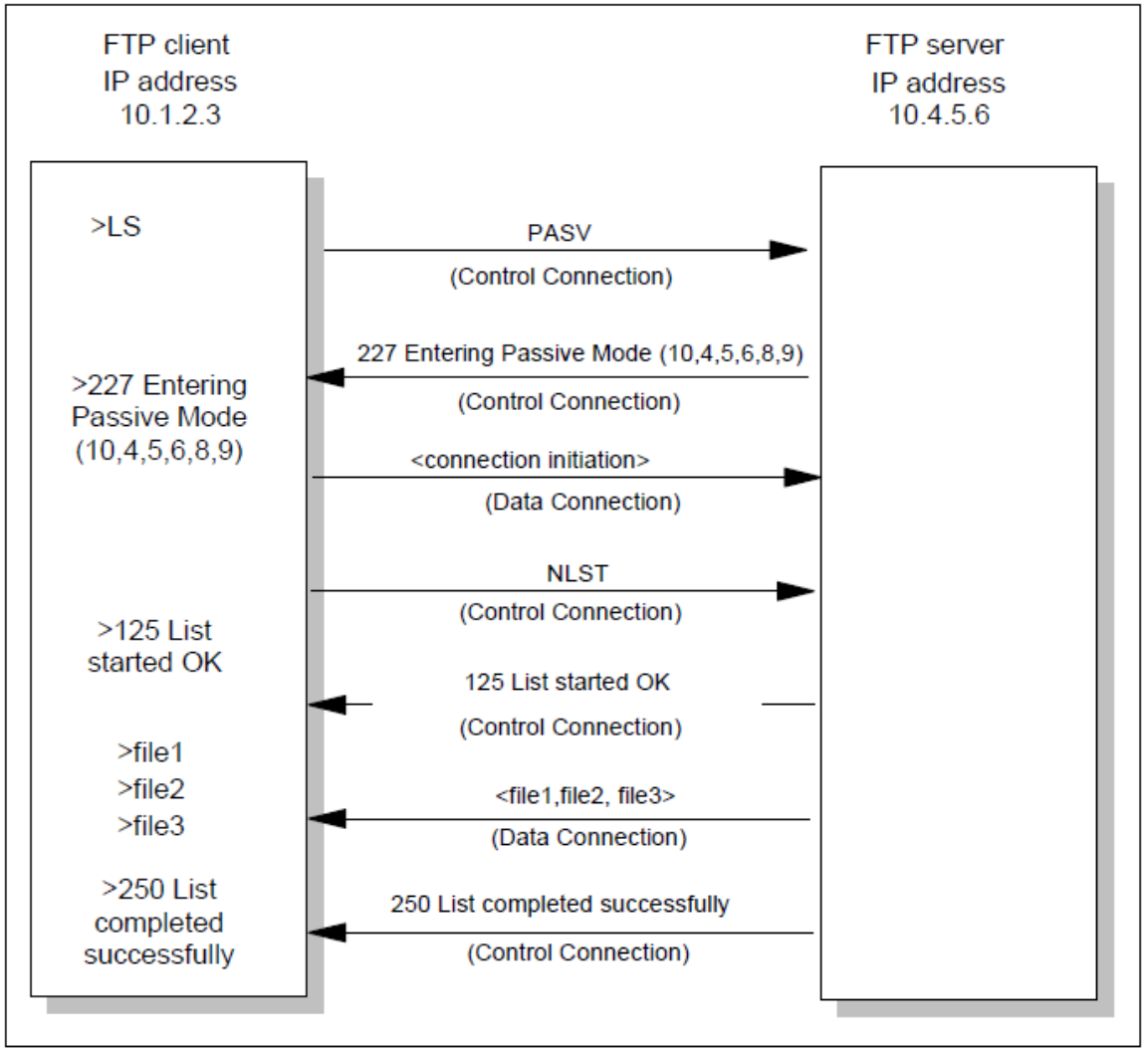
- An FTP reply consists of a three digit number followed by some text.
- The number is intended for use by automata to determine what state to enter next; the text is intended for the human user.
- The three digits of the reply each have a special significance.
  - The first digit denotes whether the response is good, bad or incomplete
  - A user-process that wants to know approximately what kind of error occurred (e.g. file system error, command syntax error) may examine the second digit
  - The third digit for the finest gradation of information

6. Transport Binding



Active data transfer

6. Transport Binding



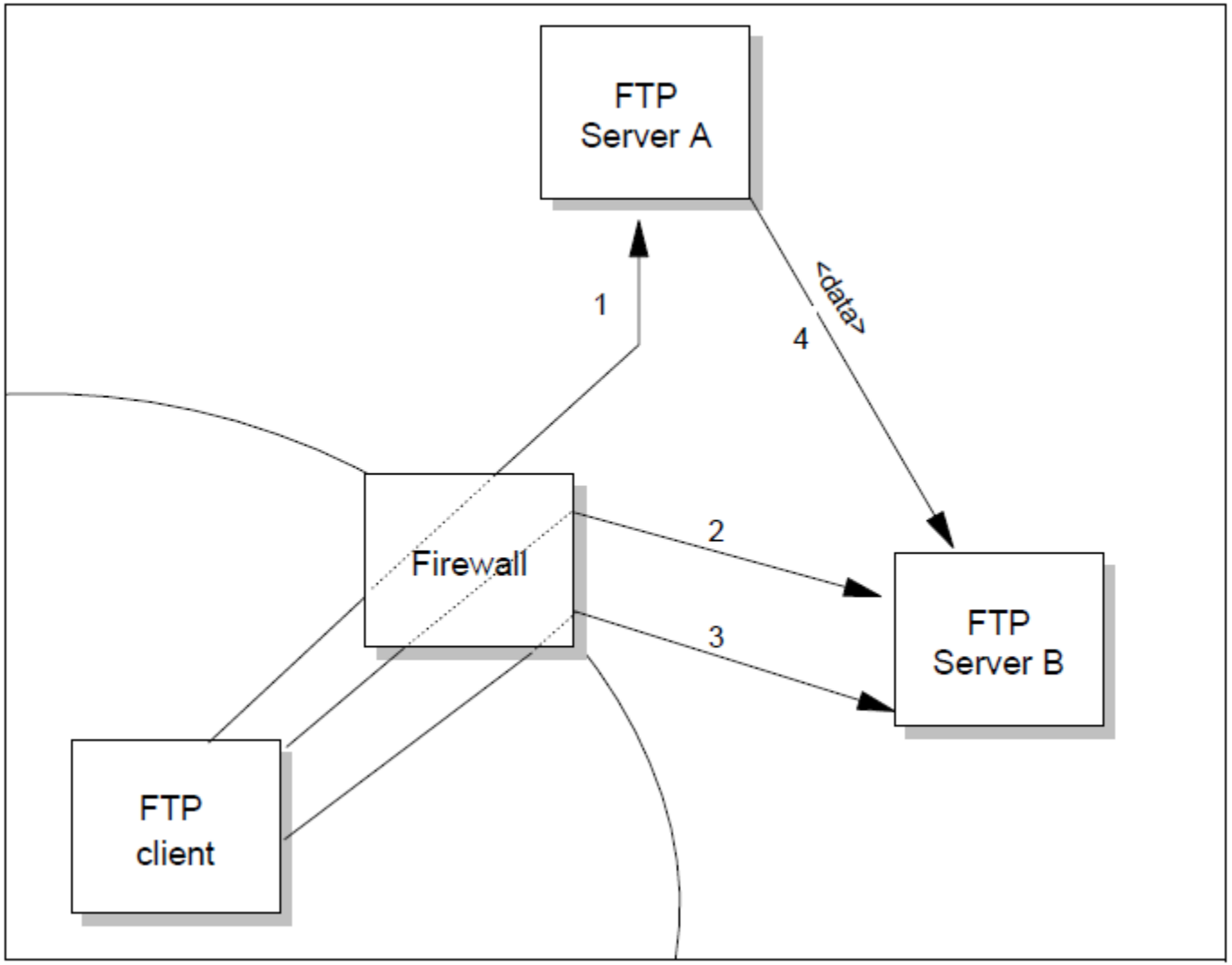
Passive data transfer

6. Transport Binding

- **FTP proxy transfer**
- FTP provides the ability for a client to have data transferred from one FTP server to another FTP server. Several justifications for such a transfer exist, including:

  - To transfer data from one host to another when direct access to the two hosts are not possible
  - To bypass a slow client connection
  - To bypass a firewall restriction
  - To reduce the amount of traffic within the client's network

6. Transport Binding



An FTP proxy transfer through a firewall

7. Security Issues

- When transferring data from one host to another, the data within the packets is sent in clear text.
- Therefore, network tools such as packet traces and sniffer devices can capture the packets and gain access to the transferred data.
- Additionally, the user ID and password used to log on to the server can be captured in these traces, giving a malicious user access to the system.
- To avoid this problem, the design of FTP has been enhanced to make use of Transport Layer Security (TLS).
- TLS defines a standard of data encryption between two hosts
- Applications only need to know how to invoke TLS.

References

- *http://www.w3.org/Protocols/rfc959/*
- *http://en.wikipedia.org/wiki/File_Transfer_Protocol*
- *"Computer Networking – A Top Down Approach" 6th Edition – Kurose Ross*
- *IBM Redbook: "TCP/IP Tutorial and Technical Overview" – 12/2006*

Vietnamese-German University

VGU