



# Web Applications Development

---

## UNIT 4

# FILE TRANSFER PROTOCOL



# U4: File Transfer Protocol

---

## 1. INTRODUCTION



# U4: File Transfer Protocol

---

- General file transfer protocols perform one main function: allowing files to be copied from one computer to another.
- While many people now use electronic mail or Web browsers to perform the functions formerly performed exclusively using general file transfer, these older protocols are still very important and widely used, and important to understand.



# U4: File Transfer Protocol

---

## 1. File Transfer Protocol (FTP)



# U4: File Transfer Protocol

---

- The first FTP standard was RFC 114, published in April 1971, before TCP and IP even existed
- The first standard to define FTP operation over modern TCP/IP was in 1980
- It is based on a client/server model, with an FTP client on a user machine creating a connection to an FTP server to send and retrieve files to and from the server



# U4: File Transfer Protocol

---

- It was designed to allow the efficient transfer of files between any two devices on a TCP/IP internetwork.
- It automatically takes care of the details of how files are moved, provides a rich command syntax to allow various supporting file operations to be performed (such as navigating the directory structure and deleting files)
- To ensure that files are sent and received without loss of data that could corrupt them it operates using the TCP transport service for reliability.



# U4: File Transfer Protocol

---

- FTP is designed to automatically deal with many of the issues that can potentially arise due to format differences in files stored on differing systems.
- An authentication system is used to ensure that only authorized clients are allowed to access a server.
- At the same time, a feature sometimes called anonymous FTP allows an organization that wishes it to set up a general information server to provide files to anyone who might want to retrieve them.



# U4: File Transfer Protocol

---

## FTP Control Connection and Data Connection





# U4: File Transfer Protocol

---

- **FTP does not use just one TCP connection for all communication the way most protocols do.**
- **The FTP model is designed around two logical channels of communication between the server and user FTP processes:**
  - **Control Connection**
  - **Data Connection**



# U4: File Transfer Protocol

---

## FTP Control Connection and Data (cont)

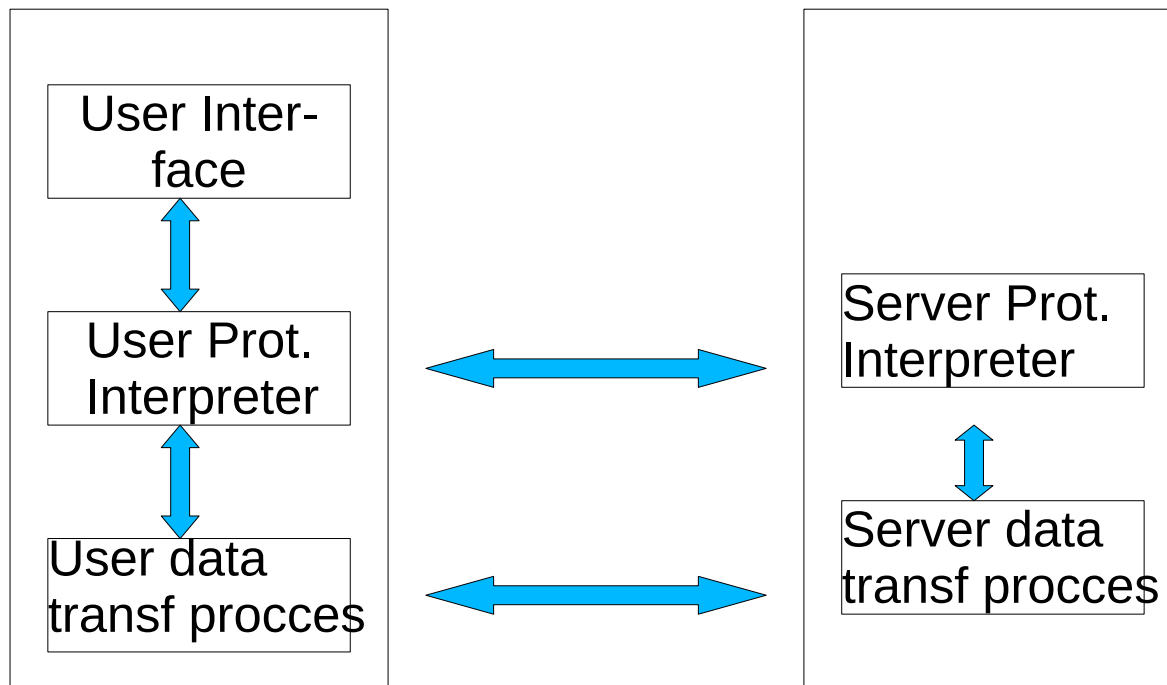
- Control Connection: This is the main logical TCP connection that is created when an FTP session is established. It is maintained throughout the FTP session and is used only for passing control information, such as FTP commands and replies. It is not used to send files.
- Data Connection: Each time that data is sent from the server to the client or vice-versa, a distinct TCP data connection is established between them. Data is transferred over this connection. When the file transfer is complete, the connection is terminated.



# U4: File Transfer Protocol

## FTP Control Connection and Data (cont)

- FTP model divides the software on each device into two logical protocol components that are responsible for each channel (control- data).



Web Applications Development

Web Applications Deployment



# U4: File Transfer Protocol

---

The Server-FTP Process contains these two protocol elements:

- **Server Protocol Interpreter (Server-PI):** is the responsible for managing the control connection on the server. It listens on the main reserved FTP port for incoming connection requests from users (clients). Once a connection is established, it receives commands from the User-PI, sends back replies, and manages the server data transfer process.
- **Server Data Transfer Process (Server-DTP):** used to send or receive data to or from the User-DTP. The Server-DTP may either establish a data connection or listen for a data connection coming from the user. It interacts with the server's local file system to read and write files.



# U4: File Transfer Protocol

---

The User-FTP Process contains these three protocol elements:

- **User Protocol Interpreter (User-PI):** it's responsible for managing the control connection on the client. It initiates the FTP session by issuing a request to the Server-PI. Once a connection is established, it processes commands received from the user interface, sends them to the Server-PI, and receives back replies. It also manages the user data transfer process.
- **User Data Transfer Process (User-DTP):** it sends or receives data to or from the Server-DTP. The User-DTP may either establish a data connection or listen for a data connection coming from the server. It interacts with the client device's local file system.



# U4: File Transfer Protocol

---

The User-FTP Process contains these three protocol elements (cont):

- **User Interface:** it provides a more “friendly” FTP interface to a human user. It allows simpler user-oriented commands to be used for FTP functions, and also allows results and information to be conveyed back to the person operating the FTP session.



# U4: File Transfer Protocol

---

## **FTP Control Connection Establishment, User Authentication and Anonymous FTP Access**



# U4: File Transfer Protocol

---

- The FTP server assumes a passive role in the control connection process.
- The server protocol interpreter (Server-PI) “listens” on the special well-known TCP port reserved for FTP control connections: port **21**.
- The User-PI uses an ephemeral port number as its source port in the TCP connection.
- The first thing to do after the channel is operating is user authentication, which the FTP standard calls the login sequence. There are two purposes for this process:
  - Access Control: it allows access to the server to be restricted to only authorized users.
  - Resource Selection: the FTP server can make decisions about what resources to make available to the user.





# U4: File Transfer Protocol

## FTP Login Sequence and Authentication

- An FTP session begins with the establishment of a TCP connection between the client and server.
- After that, the client sends the user name and password to authenticate with the server.
- The server checks the user name and password against its user database, to verify that the connecting user has valid authority to access the server.
  - If the information is valid, the server sends back a greeting to the client to indicate that the session is opened.
  - If the user improperly authenticates, the server will request that the user attempt authorization again. After a number of invalid authorization tries, the server may time out and terminate the connection.
- There are FTP security extensions



# U4: File Transfer Protocol

---

- Anonymous FTP

- It allows a guest who has no account on the server to have limited access to server resources.
- This is often used by organizations that wish to make files available to the public for purposes such as technical support, customer support, or distribution.



# U4: File Transfer Protocol

---

## **FTP Data Connection Management, Normal (Active) and Passive Data Connections and Port Usage**



# U4: File Transfer Protocol

---

- The FTP standard specifies two different ways of creating a data connection,
- The two methods differ primarily in which device initiates the connection:
  - the server (normal or active)
  - the client (passive)



# U4: File Transfer Protocol

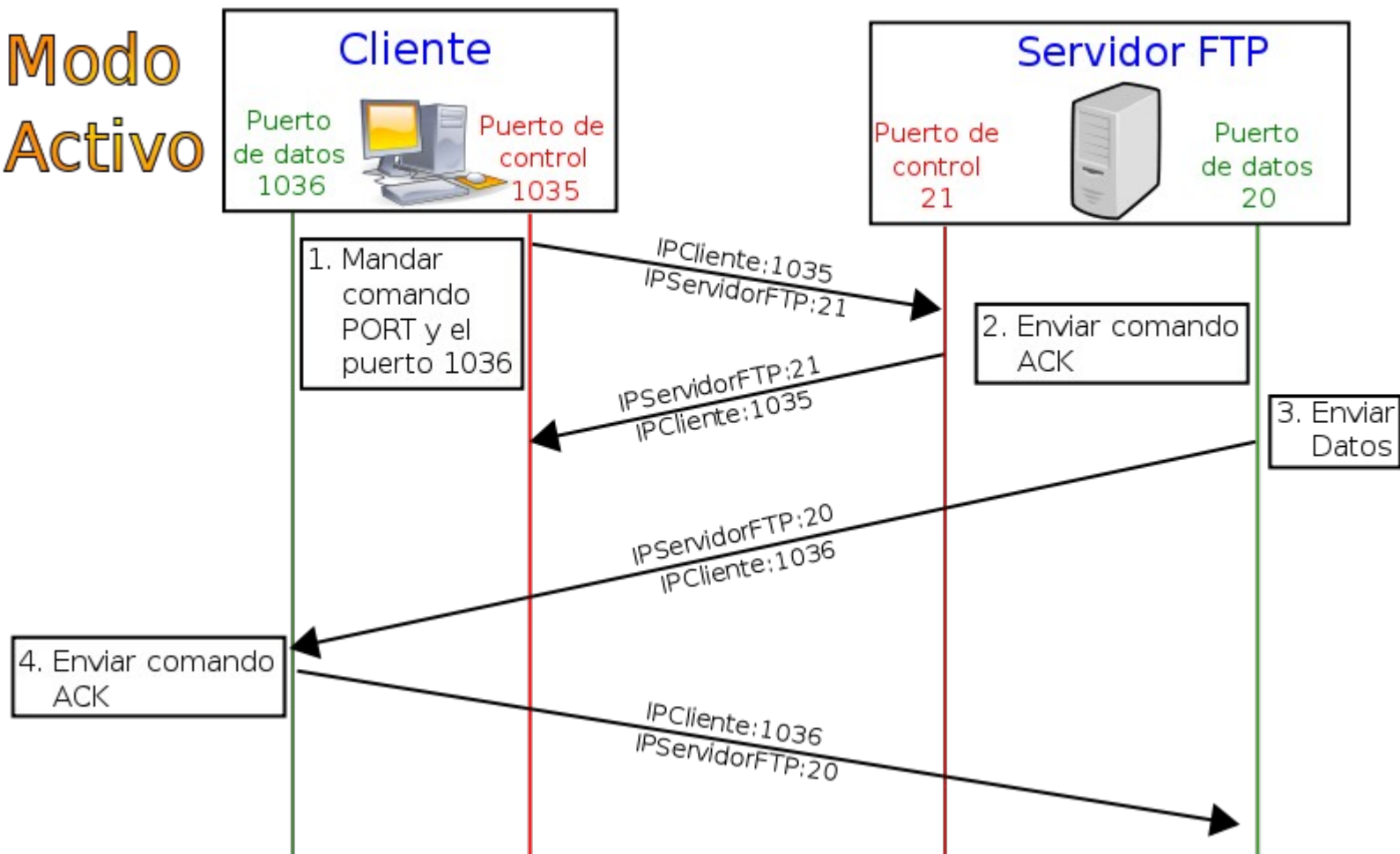
---

- Normal (or active) Data Connections,
  - It is the default method
  - The Server-DTP initiates the data channel by opening a TCP connection to the User-DTP.
  - The server uses the special reserved port number 20 for the data connection.
  - The client specify a different port number using the PORT command sending it through the control connection. It can be used the same ephemeral port number used for the control connection, but this complicates the operation of FTP and can lead to some tricky problems.



# U4: File Transfer Protocol

## Modo Activo





# U4: File Transfer Protocol

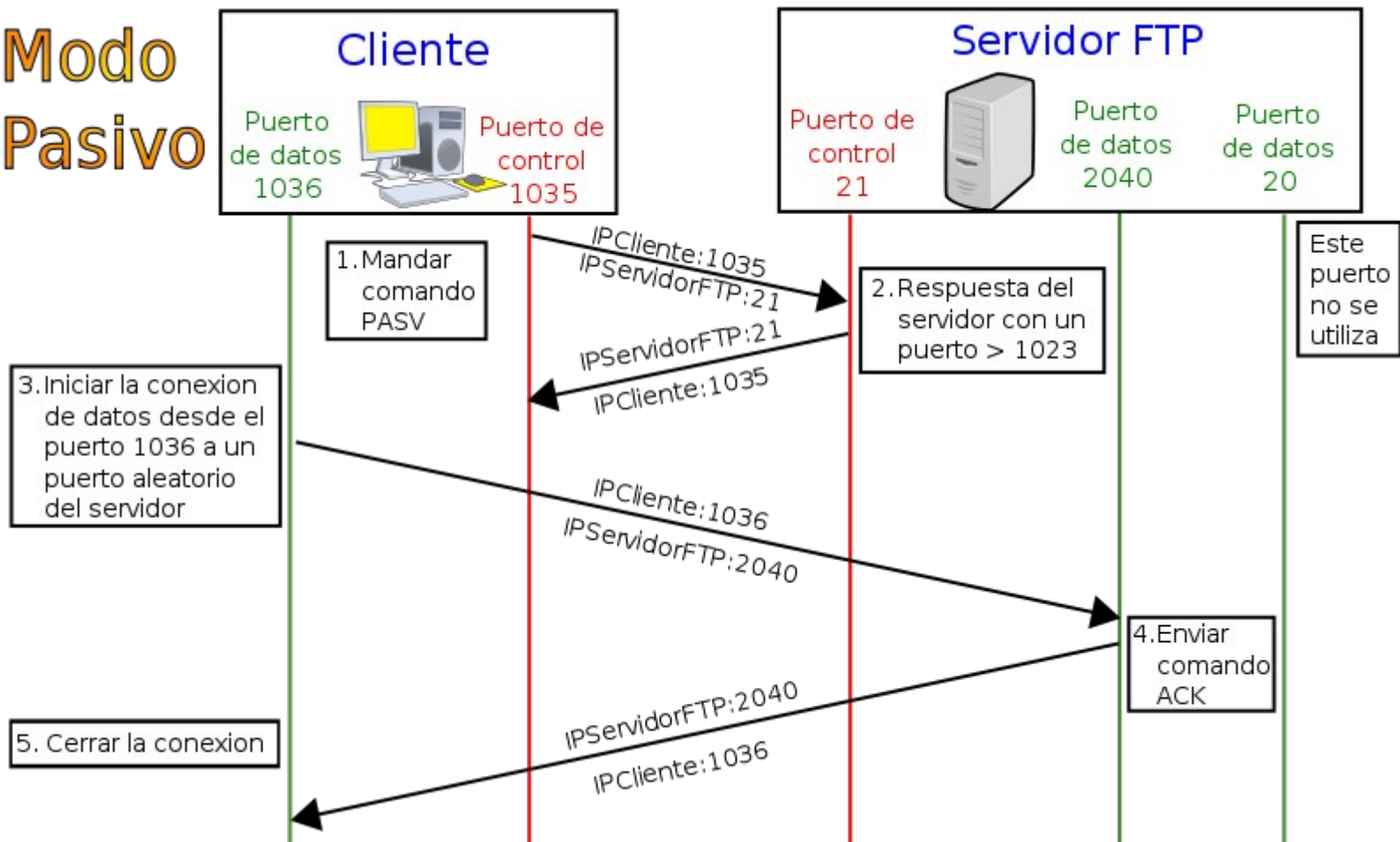
## ■ Passive Data Connections

- The client uses the PASV command to tell the server to wait for the client to establish the data connection (through the control connection).
- The server responds, telling the client what port it should use on the server for the data transmission
- The client then opens the data connection using that port number on the server and a client port number of its own choosing.
- Passive mode is often used because many client devices today are not able to accept incoming connections from servers.



# U4: File Transfer Protocol

## Modo Pasivo







# U4: File Transfer Protocol

---

## FTP General Data Communication and Transmission Modes



# U4: File Transfer Protocol

---

- FTP defines three different transmission modes or transfer mode:
  - stream mode,
  - block mode, and
  - compressed mode.
- They specify exactly how data is sent from one device to another over an opened data channel.



# U4: File Transfer Protocol

---

## • Stream Mode

- Data is sent simply as a continuous stream of unstructured bytes.
- No message format with distinct header fields is used,
- The end of the file is indicated simply by the sending device closing the data connection when it is done.



# U4: File Transfer Protocol

---

## • Stream Mode (cont)

- It's by far the most widely used in real FTP implementations. There are likely three reasons for this:
  - It is the default and also the simplest method,
  - It's the most general, because it treats all files as simple streams of byte without paying attention to their content.
  - It is the most efficient method because no bytes are wasted on “overhead” such as headers.



# U4: File Transfer Protocol

---

## ■ Block Mode

- Data is broken into data blocks and encapsulated into individual FTP blocks, or records.
- Each record has a three-byte header that indicates its length and contains information about the data blocks being sent.
- A special algorithm is used to keep track of the transmitted data and to detect and restart an interrupted transfer.



# U4: File Transfer Protocol

---

## ● Compressed Mode

- A relatively simple compression technique called run-length encoding is used to detect repeated patterns in the data being sent, and then represent them in such a way that the overall message takes fewer bytes.
- The compressed information is then sent in a way similar to block mode, using a header+payload record format.



# U4: File Transfer Protocol

---

## ● Compressed Mode

- Compressed mode seems on the surface to be useful. However, compression is often implemented in other places in a typical networking software stack, making it unnecessary in FTP.
- Data is broken into data blocks and encapsulated into individual FTP blocks, or records.
- Each record has a three-byte header that indicates its length and contains information about the data blocks being sent.
- A special algorithm is used to keep track of the transmitted data and to detect and restart an interrupted transfer.



# U4: File Transfer Protocol

---

## FTP User Interface and User Commands





# U4: File Transfer Protocol

---

- The FTP user interface:
  - Presents FTP to the human user in a way that is easier and simpler to use than issuing protocol commands
  - The command used to perform a particular function can be customized based on common parlance in the networking industry, without requiring changes to be made to the FTP protocol itself
  - A single user command can be made to issue multiple FTP protocol commands, hiding internal FTP details and making the protocol easier to use



# U4: File Transfer Protocol

---

- There are two types of FTP user interfaces
  - Command-line interface that understands English-like user commands
  - Graphical interface, where mouse clicks and other graphical operations are translated into FTP commands.



# U4: File Transfer Protocol

## COMMANDS

- **ascii:** Switch to ASCII mode. ASCII mode is the default mode; use it for transferring text files.
- **binary:** Switch to binary mode. Use to transfer binary files, including files ending in .zip, .tar, .Z, and .gz, executable programs, and graphics files.
- **bye (or quit):** Close the connection to the remote computer and exit FTP.
- **cd:** Change the directory on the remote computer.
- **close:** Close the connection to the remote computer.
- **del:** Delete files from the remote computer.
- **dir (or ls):** List the files in the current directory on the remote computer.



# U4: File Transfer Protocol

---

- **get:** Copy a file from the remote computer to the local computer.
- **help (or ?):** Lists or provides help on the use of FTP commands.
- **lcd:** Change the directory on the local computer.
- **lpwd:** Show the current directory (present working directory) on the local computer. This command is not available in all FTP versions. On Unix systems, try !pwd if lpwd doesn't work.
- **mdel:** Delete multiple files on the remote computer.
- **mget:** Copy multiple files from the remote computer to the local computer.
- **mkdir:** Create a directory on the remote host.



# U4: File Transfer Protocol

---

- **mput:** Copy multiple files from the local computer to the remote computer.
- **open:** Open a connection to a remote computer.
- **put:** Copy a file from the local computer to the remote computer.
- **pwd:** Show the current directory (present working directory) on the remote computer.
- **rmdir:** Remove a directory on the remote host (the directory usually has to be empty).
- **user :** Log into the remote computer to which you are currently connected. FTP will ask for a login name and possibly a password.



# U4: File Transfer Protocol

---

- **! :** In Unix, exit to the shell prompt, where you can enter commands. Enter exit to get back to FTP. If you follow ! with a command (e.g., !pwd), FTP will execute the command without dropping you to the Unix prompt.



# U4: File Transfer Protocol

---

## Trivial File Transfer Protocol (TFTP)



# U4: File Transfer Protocol

---

- For situations where the full File Transfer Protocol (FTP) is either unnecessary or impractical, the simpler Trivial File Transfer Protocol (TFTP) was developed.
  - Transport: TFTP uses UDP. The TFTP server listens continuously for requests on well-known UDP port number 69
  - Limited Command Set: TFTP only allows files to be sent and received.
  - Limited Data Representations: TFTP does not include some of FTP's fancy data representation options; it allows only simple ASCII or binary file transfers.
  - Lack of Authentication: UDP uses no login mechanism or other means of authentication.