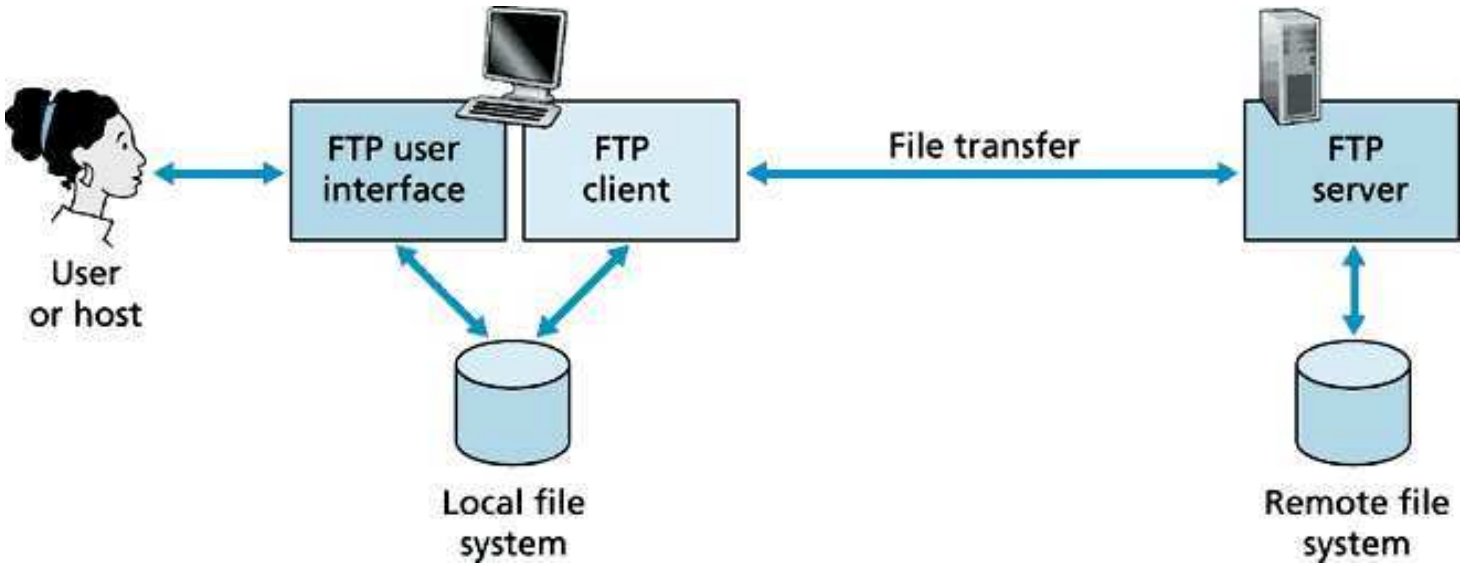
Plan

- Application layer
  - Principles of application layer protocols
  - The Web and HTTP
  - **File transfer: ftp**
  - Electronic mail in the Internet
  - DNS — The Internet's directory service
  - Socket programming with TCP and UDP
  - Content distribution

File Transfer Protocol (FTP)

The **File Transfer Protocol** (**ftp**) follows a client/server model, as HTTP.



The user wants to access a remote account in order to retrieve or send some files. Therefore he must possess an identification and a password on the remote machine. FTP runs on top of TCP.

File Transfer Protocol (FTP) (cont)

First, he enters the name of the remote host to the FTP client through the FTP user agent (under Unix, it is simply a shell command-line but in Windows it is a graphical interface).
Once the server has authorised the user, he can copy files in his local filesystem to the remote host or vice versa.
Both HTTP and FTP are file transfer protocols but there is an important technical difference: FTP uses two different TCP connections to transfer a file: a **control connection** and a **data connection**.

File Transfer Protocol (FTP) (cont)

The control connection is used for exchanging control information between the two hosts, like identification, password, commands to change remote directory, commands to get and put files etc.
The data connection is used to actually exchange the files.
Because FTP uses a separate control connection, it is said to send its control information **out-of-band**. Dually, HTTP sends its control information **in-band**.

File Transfer Protocol (FTP) (cont)

First, the user starts an FTP session with the remote host by initiating a control TCP connection on server port 21.
The FTP client sends user identification and password to the server over this control connection.
After authentication, the client then sends commands to change remote current directory.
When the server receives a command for file transfer to the client over the control connection, it opens a TCP data connection on client port 20.

File Transfer Protocol (FTP) (cont)

After sending exactly one file over the data connection, the server closes it.
So, during an FTP session, only the control connection remains open, one data connection is open for one file at a time, then closed, i.e. data connections are non-persistent.
Throughout a session, the FTP server must maintain a **state** about the user:

- the control connection must be associated with a specific user account;
- the server must keep track of a given user's current directory on its file-system.

By contrast, HTTP is a stateless protocol — it does not keep any user state.

File Transfer Protocol (FTP) (cont)

Common commands and replies

The commands from client to server, and replies, are sent over the control connection in ASCII (i.e. non-accentuated characters and digits encoded with seven bits). Hence, FTP commands, as HTTP ones, are readable by people (plain text).

One command lies on one line and it consists of four characters and some optional arguments. The more common are

- `USER username` is used to send the user identification to the server;
- `PASS password` is used to send the user's password to the server;
- `LIST` is used to ask the server the list of all files in the current directory; this list is sent back on a new and non-persistent data connection;

File Transfer Protocol (FTP) (cont)

- `RETR filename` is used to retrieve (i.e. `get`) a file from the current remote directory;
- `STOR filename` is used to store (i.e. `put`) a file into the current directory of the remote host.

There is typically a one-to-one correspondence between the commands the user enters (by means of the user agent) and the commands the FTP client sends, as exactly one client's `RETR` for one user's `get`.

File Transfer Protocol (FTP) (cont)

The replies consist in a three-digit number followed by an optional message — this is similar to the status code and phrase in the status line of the HTTP response messages.
Some typical replies are

- `331 Username OK, password required`
- `125 Data connection already open; transfer starting`
- `425 Can't open data connection`
- `452 Error writing file`