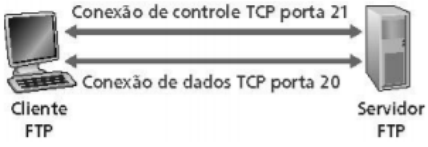
FTP: the file transfer protocol

- Transferring files to and from the remote computer
- Client server model
- Client: side that initiates the transfer (either from or to the remote side)
- Server: remote host
- FTP: RFC 959
- FTP server: port 21

FTP: separate control, data connections



- FTP and HTTP protocols use TCP. However, FTP uses two parallel TCP connections to transfer a file:
  - A Control Connection
  - A Data Connection

- FTP client contacts the FTP server on port 21 specifying TCP as the transport protocol.

- Client obtains authorization over control connection.

- Client searches the remote directory by sending commands over the control connection.

- When the server receives a command for a file transfer, it opens a TCP data connection to the client.

- After transferring a file, the server closes the connection.

- Server opens a second TCP data connection to transfer another file.

- Control connection: "out of band".

- FTP server keeps "state": current directory, previous authentication.

• Thus, with FTP, the control connection remains open for the entire user session, but a new data connection is created for each file transferred within a session.

FTP commands, responses

• Send ASCII text on control channel
  - **USER username**
  - **PASS password**
  - **LIST** returns listing of file in current server directory
  - **RETR filename** extracts the file from the server. Activates the server to open a data connection and send the file requested by this connection.
  - **STOR filename** stores the file on the remote host

Each command is followed by a response, which is sent from the server to the client.

Return Code Examples
• Status code and phrase (as in HTTP)
• **331 Username OK, password required**
• **125 Data connection already open; starting transfer**
• **425 Cannot open data connection**
• 452 Error writing file

SMTP

Transfers messages from sending mail servers to receiving mail servers.

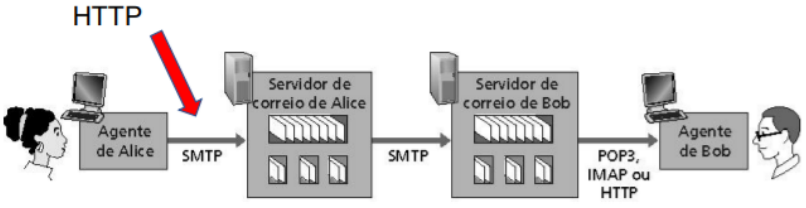
Three main components:
• User agents
• Mail servers
• Simple mail transfer protocol: SMTP
RFC 5321

User agent
• "mail reader"
• Composition, editing, reading of
mail messages
• E.g.: gmail

Mail servers
• Mailbox contains messages that have arrived (not yet read) for the user
• Message queue contains the mail messages to be sent

SMTP protocol allows mail servers to exchange messages with each other
• Client: sending mail server
• “Server”: receiving mail server

Scenario: Alice sends message to Bob



1) Alice uses the user agent (UA) to compose the message and send "to" bob@someschool.edu
2) Her user agent sends the message to her mail server; the message is placed on the message queue.
3) The client side of SMTP opens a TCP connection to Bob's mail server.
4) SMTP client sends Alice's message over TCP connection.
5) Bob's mail server puts the message in Bob's mailbox.
6) Bob invokes his user agent to read the message.

It is important to note that SMTP does not use intermediary mail servers to send the message.

Email: SMTP [RFC 2821]

- Uses TCP for reliable transfer of mail messages from client to server, port 25
- Direct transfer: server that sends to server that receives
- Three phases of transfer
- Handshaking (presentation)
- Message transfer
- Closure

• Command/Response Interaction
• Commands: ASCII text
• Answer: status code and phrase
• Messages must be formatted in 7-bit ASCII code

SMTP is a push protocol, not used to get data.

SMTP interaction example

“C”: lines that the client sends into its TCP socket.
“S”: lines that the server sends into its TCP socket.

S: 220 hamburger.edu
C: HELO crepes.fr
S: 250 Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Voce gosta de ketchup?
C: Que tal pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection

When an email message is sent, a header containing peripheral information can be attached.

This peripheral information is contained in a series of header lines defined in RFC 5322-------MINE Commands.
A typical header looks like this:

From: alice@crepes.fr
To: bob@hamburger.edu
Subject: envio de dados.

It is important to note that these header lines are different from SMTP commands, even though they contain some common words.

SMTP commands are part of the protocol; the header lines examined in this section are part of the message itself.~

Mail access protocols

- SMTP: delivers and stores on destination server
- Access protocols: **TRANSFER MESSAGES FROM THE MAIL SERVER TO THE LOCAL PC**
- **POP3**: Post Office Protocol [RFC 1939]
    - Authorization (agent <-->server) and download
    - Port 110
- **IMAP**: Internet Mail Access Protocol [RFC 3501]
    - More features (more complex)
    - Handling of messages stored on the server
- **HTTP**: Hotmail, Yahoo! Mail etc.

POP3

- Uses "read & delete" and "read & save" mode
- Does not allow manipulating folders on the mail server.

IMAP

- Keeps all messages in one place: the server
- Allows the user to organize messages into folders

HTTP

- Means of sending and accessing the most used email.
- Released by Hotmail in the mid-1990s.