Email

How much is there to say about Email?
email: The killer app of the seventies

• In 1971 Ray Tomlinson sent the first successful email from one computer, to another sitting right next to it – through ARPANET.

• He used the @ sign to delineate what host the user receiving the message was on.

• Tomlinson can’t remember his first message but stated it was “QWERTYUIOP or something similar.”

You’ve got mail*

• Like ordinary mail (and unlike either the HTTP or FTP), e-mail is asynchronous.

• It has three major components:
  – user agents
  – mail servers
  – Simple Mail Transfer Protocol (SMTP)

* Please tell me you get that reference!
Electronic mail in the Internet

User agents allow users to read, reply, forward, save, and compose messages.

Currently, our browser acts as a user agent. Formerly, we used a proprietary agent from FirstClass.

Servers form the core

Each recipient has a mailbox located in one of the mail servers.
SMTP* 

SMTP (Simple Mail Transfer Protocol, is the principal application-layer protocol for Internet electronic mail. Like HTTP, SMTP has a client and server side. Both sides run on every server.

*Something of a legacy technology (RFC 1982), SMTP requires body of all mail messages to be in simple seven-bit ASCII.

Alice sends Bob a message

Key:
- Message queue
- User mailbox
Alice sends Bob a message

Alice invokes her user agent, provides Bob's address, composes message, and instructs user agent to send.

Alice's user agent sends the message to her mail server, where it is placed in message queue.
Alice sends a message to Bob. This involves the following steps:

1. Client side of SMTP, running on Alice's server, sees the message and opens a TCP connection to the SMTP server on Bob's mail server.
2. After initial SMTP handshaking, the SMTP client sends Alice's message into the TCP connection.

Diagram:
- Alice's agent
- Alice's mail server
- Bob's mail server
- Bob's agent

Key:
- Message queue
- User mailbox
Alice sends Bob a message

At Bob’s mail server, the server side of SMTP receives message and puts it into Bob’s mailbox.

Bob invokes his user agent to read the message.
Try SMTP interaction for yourself!

- `telnet servername 25`
- see 220 reply from server
- enter HELO, MAIL FROM, RCPT TO, DATA, QUIT commands

above lets you send email without using email client (reader)

Comparing HTTP & SMTP

- Both transfer files using persistent connections on top of TCP.
- HTTP is mainly a pull protocol, while SMTP is a push protocol.
- SMTP requires each message to be 7-bit ASCII.
- HTTP encapsulates each object in its own HTTP message. SMTP places all objects in one message.
Mail message format

- **SMTP**: protocol for exchanging email msgs
- RFC 822: standard for text message format:
  - Header lines, e.g.,
    - To:
    - From:
    - Subject: different from SMTP MAIL FROM, RCPT TO: commands!
- Body: the “message”
  - ASCII characters only

Problem: How do we send attachments?

From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Picture of yummy crepe.
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg
base64 encoded data ..... 
..........................
......base64 encoded data

*Multipurpose Internet Mail Extensions.*
Unanswered questions

- Alice user agent uses SMTP to push e-mail to her mail server.
- Mail server relays e-mail to Bob. Why the two step?
- How does Bob’s user agent obtain his message from Alice?

Mail access protocols

- Since SMTP is a PUSH protocol and Bob wants to PULL his messages, he needs another application-layer protocol.

- There are several to choose from: Post Office Protocol (POP3); Internet Mail Access Protocol (IMAP); and HTTP.
POP3 protocol

• POP3 is an extremely simple mail access protocol with rather limited functionality, ...

• ... but it does get the job done.

authorization phase

• client commands:
  – user: declare username
  – pass: password

• server responses
  – +OK
  – -ERR

transaction phase, client:

• list: list message numbers
• retr: retrieve message by number
• dele: delete
• quit

S: +OK POP3 server ready
C: user bob
S: +OK
C: pass hungry
S: +OK user successfully logged on

C: list
S: 1 498
S: 2 912
S: .

C: retr 1
S: <message 1 contents>
S: .

C: dele 1
C: retr 2
S: <message 1 contents>
S: .

C: dele 2
C: quit
S: +OK POP3 server signing off
IMAP and HTTP

• POP3 is a no frills access protocol.

• IMAP is somewhat more luxurious.
  – keeps all messages on server
  – allows users to organize messages into folders
  – keeps user state across sessions (like FC).

• Alternatively, we could build an e-mail application on top of HTTP (Gmail, FC, … ).

The File Transfer Protocol
The original specification for the File Transfer Protocol was written by Abhay Bhushan (RFC 114, April 1971).

It ran on NCP until 1980* when it was replaced by a TCP/IP version (RFC 114).

*Flag day.

Like HTTP, FTP is a file transfer protocol and shares many common characteristics. However, there are some important differences.

*Fetch on the Mac and Winsock on the PC.
FTP and HTTP

- FTP, like HTTP, runs on top of TCP.
- However, unlike HTTP, FTP uses two parallel TCP connections to transfer a file, a control connection and a data connection. We say FTP sends its control information out-of-band.

- Also unlike HTTP, FTP maintains state. In particular, FTP remembers the current directory and earlier authentication.

Control and data connection

- When a user starts an FTP session, the client side of FTP initiates a control TCP connection with the server.
- When the server receives a command for a file transfer, it initiates a TCP data connection to the client side.
- The control connect stays open. The data connection stays open for one file only.
FTP commands*, responses**

Sample commands: sent as ASCII text over control channel
USER username
PASS password
LIST return list of file in current directory
RETR filename retrieves (gets) file
STOR filename stores (puts) file onto remote host

Sample return codes status code and phrase (as in HTTP)
331 Username OK, password required
125 data connection already open; transfer starting
425 Can't open data connection
452 Error writing file

*Four uppercase 7-bit ASCII characters followed by some optional arguments.
**Replies are three-digit numbers with optional message.

Web browser support

- Most web browser can retrieve files hosted on FTP servers. FTP URL syntax is described in RFC1738:

  ftp://[<username>[:<password>@]<host>[:<port>]/<url-path>

- For example,
Innocence abroad

- A host that provides an FTP service may provide anonymous FTP access.

- Users log into the service with an ‘anonymous’ account with prompted for a user name.

- Users may be asked to send their email address instead of a password, but no verification is performed.

Danger …

- FTP was not designed to be a secure protocol and it isn't.
- It is not able to encrypt its traffic; all transmissions are in clear text, and usernames, passwords, commands and data can be easily read by anyone able to perform packet capture.
- Exploits include: Offline password cracking, spoof attacks; and bounce attacks.
• Explicit SFTP is an extension to the FTP standard that allows clients to request that the FTP session be encrypted. This is done by sending the "AUTH TLS" command.

• SFTP, SSH* File Transfer Protocol, is not related to FTP except that it also transfers files and has a similar command set for users.