

The Application Layer: FTP and email

Smith College, CSC 249
February 7, 2008

many slides from J.F. Kurose and K.W. Ross, All Rights Reserved



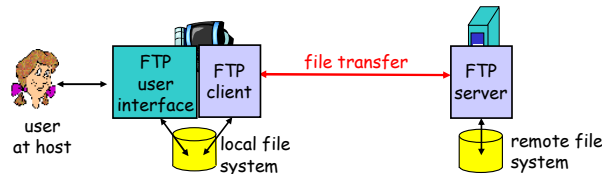
4-1

Chapter 2: Application layer

- 2.1 Principles of network applications
- 2.2 Web and HTTP
- 2.3 FTP
- 2.4 Electronic Mail
 - ❖ SMTP, POP3, IMAP
- 2.5 DNS
- 2.6 P2P file sharing
- 2.7 Socket programming with TCP
- 2.8 Socket programming with UDP
- 2.9 Building a Web server

4-2

FTP: the file transfer protocol



The Objective:

- To transfer files to and from a remote host

The Details:

- Uses the client/server model
 - ❖ *client*: side that initiates communication (file transfer)
 - ❖ *server*: remote host
- Port: ftp server port is 21, for the control connection

4-3

FTP: the file transfer protocol

- A common protocol for exchanging files over any network that supports the TCP/IP protocol
- The client
 - ❖ Initiates a connection to an FTP server by using FTP client software
- The server
 - ❖ "Listens" on the network for connection requests from other computers.
- The client can
 - ❖ upload files to the server,
 - ❖ download files from the server,
 - ❖ rename or delete files on the server, etc

4-4

New Aspect with FTP

□ HTTP

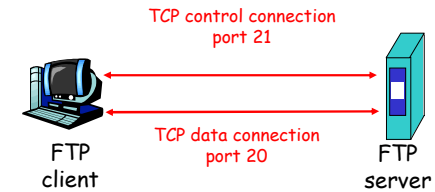
- ❖ One connection established between client and server
- ❖ Could be persistent or non-persistent
- ❖ (A web browser may be designed to open multiple parallel connections)

□ FTP

- ❖ Two connections opened
 - The **control connection** opened by the client
 - The **data connection** opened by the server

4-5

FTP: separate control & data connections

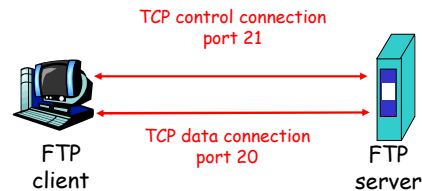


The FTP Client:

- Contacts the FTP server at port 21
- Specifies TCP as transport protocol
- Obtains authorization over control connection, with a username and password
- Browses remote directory by sending commands over control connection (unix commands)
 - ❖ List/Change directory
 - ❖ Request to send or receive files ...

4-6

FTP: separate control & data connections

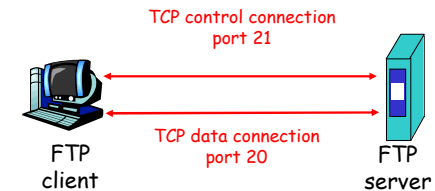


The Server:

- Listens on port 21 for an incoming connection request
- When server receives a command for a file transfer, the server opens a TCP **data connection** to client
- After transferring one file, server closes connection.

4-7

FTP: separate control & data connections



- The server opens a second TCP data connection to transfer another file.
- The control connection remains open
- The control connection is "out of band"
- **FTP server maintains "state"**: current directory, earlier authentication, which limits the total number of FTP connections possible at a time

4-8

FTP commands & responses

Sample commands:

- ❑ sent as ASCII text over control channel
 - ❖ 4 uppercase chars
- ❑ **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)
 - ❖ 3 digit numbers
- ❑ 331 Username OK, password required
- ❑ 125 data connection already open; transfer starting
- ❑ 425 Can't open data connection
- ❑ 452 Error writing file

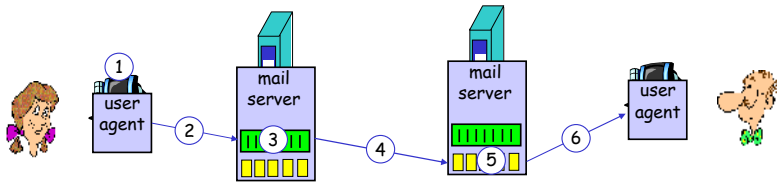
4-9

Chapter 2: Application layer

- ❑ 2.1 Principles of network applications
- ❑ 2.2 Web and HTTP
- ❑ 2.3 FTP
- ❑ **2.4 Electronic Mail**
 - ❖ SMTP, POP3, IMAP
- ❑ 2.5 DNS
- ❑ 2.6 P2P file sharing
- ❑ 2.7 Socket programming with TCP
- ❑ 2.8 Socket programming with UDP
- ❑ 2.9 Building a Web server

4-10

Scenario: Alice sends message to Bob

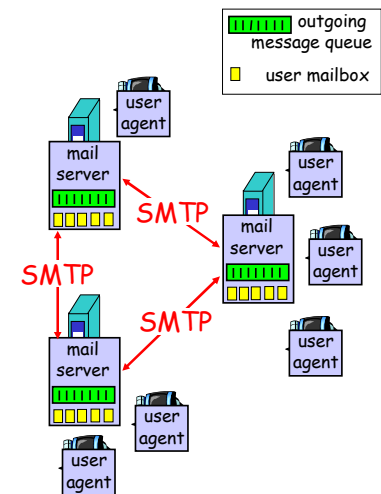


4-11

Electronic Mail

Three major components:

- ❑ user agents
- ❑ mail servers
- ❑ simple mail transfer protocol: SMTP



4-12

Electronic Mail: SMTP

- There are three phases of transfer
 - ❖ handshaking (greeting)
 - ❖ transfer of messages
 - ❖ closure
- command/response interaction
 - ❖ **commands**: ASCII text
 - ❖ **response**: status code and phrase
- client and server sides of SMTP run on every mail server
 - ❖ Use persistent TCP connections (reliable transfer)
 - ❖ Use port 25
- messages must be in ASCII
 - ❖ No binary data can be send - meaning what!?

4-13

Sample SMTP interaction

- In the following interaction with SMTP, which lines are
 - ❖ Handshaking
 - ❖ Transfer of message
 - ❖ Closure

4-14

Sample SMTP interaction

```
telnet cornell.edu 25
S: 220 cornell.edu
C: HELO edf.ch
S: 250 Hello edf.ch, pleased to meet you
C: MAIL FROM: <alice@edf.ch>
S: 250 alice@edf.ch... Sender ok
C: RCPT TO: <bob@cornell.edu>
S: 250 bob@cornell.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: to: bob@cornell.edu
C: from: alice@edf.ch
C: subject: winter weather
C:
C: Do you like snow?
C: How about blizzards?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 cornell.edu closing connection
```

4-15

Mail message format

- * Example of the actual message - NOT part of the SMTP handshaking process

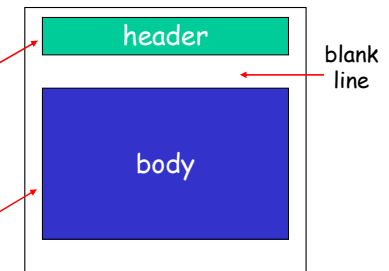
- header lines, e.g.,

- ❖ To:
- ❖ From:
- ❖ Subject:

different from SMTP commands!

- body

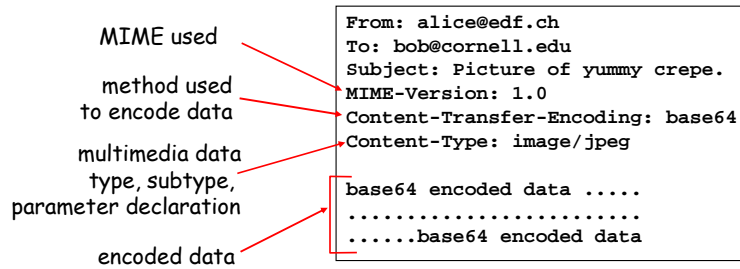
- ❖ the "message", ASCII characters only



4-16

Message format: multimedia extensions

- MIME: multipurpose internet mail extension
- additional lines in msg header declare MIME content - only version 1.0 exists now



4-17

Base64 Encoding

- Original (binary) bit stream
100110111010001011101001
100110 .. 111010 .. 001011 .. 101001
- Which corresponds to the 6-bit values
38, 58, 11 and 41
- Which are encoded as
m6Lp

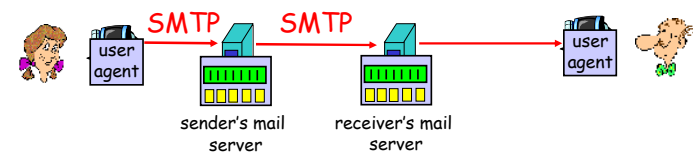
4-18

Base64 Encoding

0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

4-19

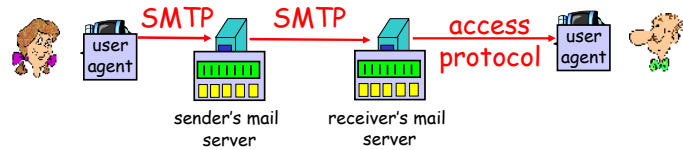
Mail access protocols



- SMTP is a 'PUSH' protocol
- So how do we 'PULL' messages off the mail server?

4-20

Mail access protocols



- ❑ SMTP: delivery/storage to receiver's server
- ❑ Mail access protocol: retrieval from server
 - ❖ POP: Post Office Protocol
 - authorization (agent <-->server) and download
 - ❖ IMAP: Internet Mail Access Protocol
 - more features (more complex)
 - manipulation of stored msgs on server
 - ❖ HTTP: gmail, Hotmail, Yahoo! Mail, etc.

4-21

SMTP: compared to HTTP

Comparison with HTTP:

- ❑ HTTP: pull
- ❑ SMTP: push

- ❑ both have ASCII command/response interaction, status codes

- ❑ SMTP: multiple objects sent in one message, using encoding as needed
 - ❖ SMTP requires message (header & body) to be in ASCII

- ❑ HTTP: each object encapsulated in its own response msg

4-22

Summary

- ❑ New protocols
 - ❖ FTP - file transfer
 - ❖ SMTP - email
- ❑ Using telnet to fake being
 - ❖ an ftp agent
 - ❖ an email agent

4-23