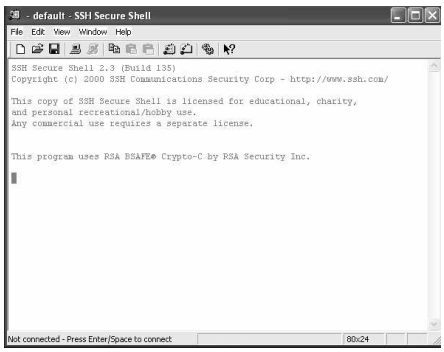
Chapter 2 outline

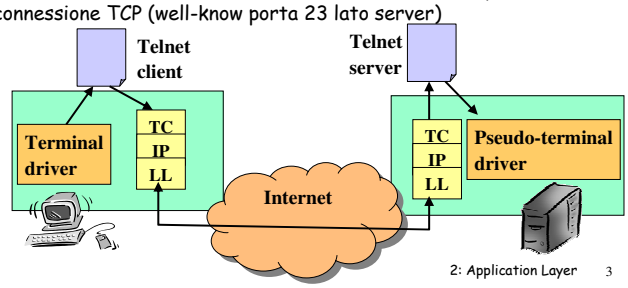
- 2.1 Principles of app layer protocols
  - clients and servers
  - app requirements
- 2.2 Web and HTTP
- 2.3 FTP
- 2.4 Electronic Mail
  - SMTP, POP3, IMAP
- 2.5 DNS
- 2.9 Content distribution
  - Network Web caching
  - Content distribution networks
  - P2P file sharing
- Telnet

TELNET (TerminaL NETwork)



TELNET (TerminaL NETwork)

- E' un semplice applicativo che consente ad un programma su un host (TELNET client) di accedere alle risorse su un host remoto (TELNET server) come se il client fosse un terminale (testuale) attaccato all'host remoto
- Un applicativo TELNET agisce da client o server a seconda che contatti o sia contattato
- Al contrario di un terminale locale i comandi sono trasferiti su una connessione TCP (well-know porta 23 lato server)



TELNET: Basic Ideas

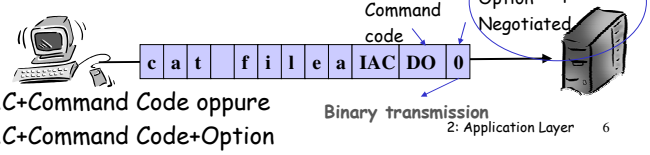
- Network Virtual Terminal
  - Imaginary device having a basic structure common to a wide range of real terminals→guarantees the colloquium follows clear rules and formats independent of the real terminal features
    - Data represented in 7 bit ASCII transmitted in 8-bit bytes
    - NVT half-duplex device operating in line-buffered mode (the characters of a line are buffered and a complete line is transmitted together)
    - Local echo provided
  - NTV made of: a printer (or display) and a keyboard. The keyboard produces the data transmitted over the TCP connection; the printer receives and displays the data.
- Negotiation of terminal options
  - Two hosts can negotiate different options to extend the NVT capabilities to reflect the capabilities of the real HW in use (at the beginning of the colloquium or dynamically during it)
- Symmetric view of terminals and processes
  - Symmetric protocol for option negotiation
  - Being a client or server may only depend on who is contacing whom

Terminal options

- Terminal options: kind of transmission (e.g. binary), window size, terminal type, line width, etc.
- How to negotiate them?
  - Usage of DO/DON't/WILL/WON'T followed by option code
  - Es (CLIENT). DO Transmit binary (requests that the other party use binary transmission)
    Reply: WILL Transmit binary (OK),
    WON'T Transmit binary (NO)
    Es2 (CLIENT) WILL Transmit binary (desire to use)
    Reply: DO Transmit binary(OK, I expect you to do so)

TELNET (TerminaL NETwork)

- TELNET trasferisce caratteri
  - Caratteri dati:
    - Sono caratteri ASCII con il primo bit pari a 0
    - Si possono trasferire anche caratteri ASCII con il primo bit pari a 1 se li si fa precedere da un byte di controllo speciale
  - Caratteri di controllo:
    - Sono comandi codificati in sequenze di 8 bit con il primo pari ad 1
    - Tra questi
      - IAC (255): interpreta il prossimo come carattere di controllo
      - WILL (251), WON'T (252), DO (253), DON'T (254)
      - EC (247)erase character etc.



IAC+Command Code oppure
IAC+Command Code+Option

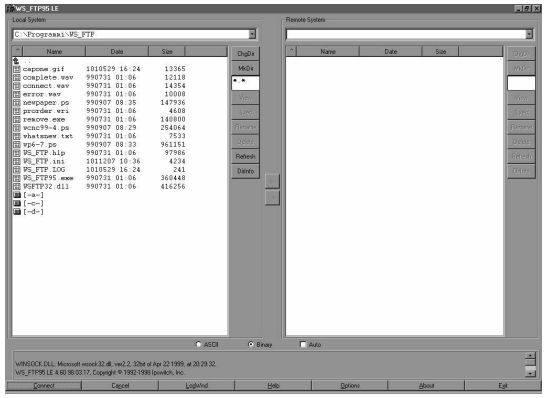
Come leggere uno standard

- Lo standard RFC di TELNET disponibile sul sito (in blu descrizione protocollo quindi vari livelli di dettaglio crescente: arancione, fucsia, nero)
- Vari livelli di lettura: primo pass per capire il funzionamento (negli standard anche molte info di dettaglio sin dall'inizio che non potete capire/che non vi interessano). NON come un libro: cio' che non capite potrebbe essere spiegato chiaramente pagine dopo. Piu' pass per una comprensione totale.
- Distinguere le info necessarie per una prima comprensione da info di dettaglio che fanno esempi/specificano meglio alcuni aspetti da info di estremo dettaglio (implementazioni in alcuni sistemi, formati etc.) che potrebbero non interessarvi affatto!!
- Dopo aver compreso con una lettura delle parti importanti il funzionamento tornate sugli argomenti di dettaglio e li rileggete con piu' attenzione se e solo se vi serve quel livello di dettaglio (un unico standard risponde a diverse esigenze: studente, implementatore, etc.!!)

Chapter 2 outline

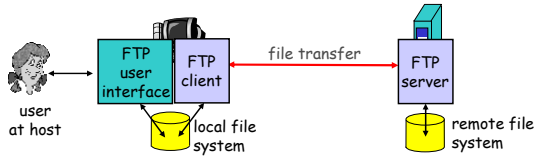
- 2.1 Principles of app layer protocols
  - clients and servers
  - app requirements
- 2.2 Web and HTTP
- 2.3 FTP
- 2.4 Electronic Mail
  - SMTP, POP3, IMAP
- 2.5 DNS
- 2.9 Content distribution
  - Network Web caching
  - Content distribution networks
  - P2P file sharing

I servizi di IP: il File Tranfer



FTP: the file transfer protocol

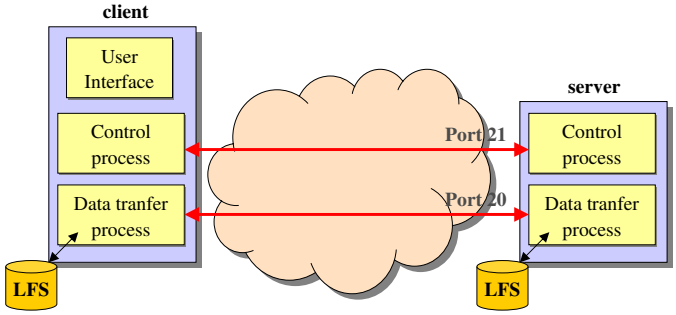
• "File Transfer Protocol", RFC 959, October 1985.



- transfer file to/from remote host
- client/server model
  - *client:* side that initiates transfer (either to/from remote)
  - *server:* remote host
- ftp: RFC 959
- ftp server: port 21

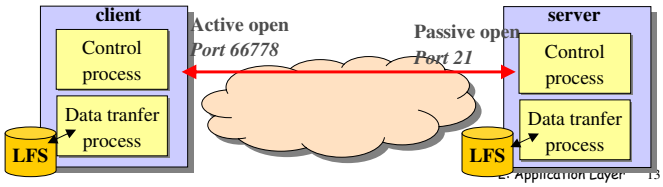
File Transfer Protocol (FTP)

- Fa uso di TCP per il trasporto
- Due connessioni sono aperte per dati e controllo



FTP: connessione di controllo

- La connessione di controllo viene aperta in modo simile alle altre applicazioni
  - Il server lancia un passive open per la porta 21 e rimane in attesa di richieste di connessione
  - Il client lancia una active open ogni volta deve iniziare una sessione di trasferimento file e fa partire una richiesta di connessione TCP usando una porta dinamica
- La connessione di controllo è *persistent* , ovvero rimane aperta per tutta la durata della sessione di traferimento e può essere usata per molti file da trasferire



FTP: connessione dati

- Le connessioni dati sono *non-persistent* ovvero sono aperte solo per trasferire un file o altre informazioni e poi sono immediatamente chiuse
- Per aprire un connessione dati:

  Metodo 1:
  - Il client che desidera iniziare un trasferimento dati effettua un passive open su una porta di sua scelta
  - Il client comunica la porta al server sulla connessione di controllo mediante il comando PORT
  - Il server fa un active open verso la porta del client usando il suo numero di porta noto 20

  Metodo 2:
  - Il client invia il comando di PASV al server
  - Il server sceglie un numero di porta, fa una passive open e comunica il numero di porta al client nella risposta
  - Il client fa un active open verso la porta comunicata dal server

FTP: connessione dati

- Il trasferimento di dati può avvenire con diverse modalità e formati:
- File type:
  - ASCII file: file di caratteri
  - Binary: formato generale per tutti i file non testuali
- Transmission mode:
  - Stream mode: il file viene trasferito al TCP come una sequenza non strutturata di byte
  - Block mode: il file viene trasferito in blocchi di cui i primi tre byte rappresentano l'header

FTP: comandi

- Il trasferimento di comandi di FTP è testuale (ASCII)

Comandi di accesso

```
USER username
PASS password
QUIT log out
```

Gestione file

```
CWD  change directory
DELE delete file
LIST list files
RETR retrive file
STOR store file
```

Modalità di trasferimento

```
TYPE file type
MODE transfer mode
```

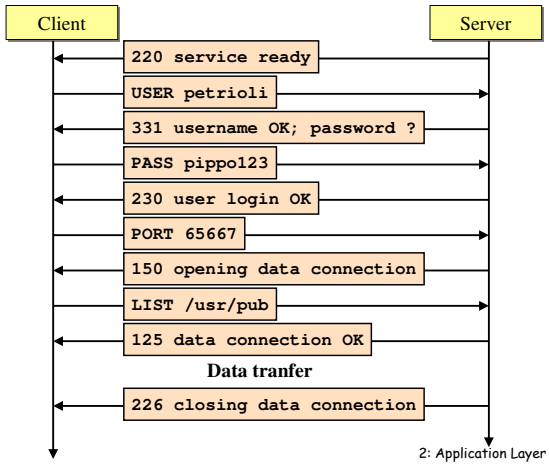
Gestione delle porte

```
PORT client port
PASV server choose port
```


FTP: risposte

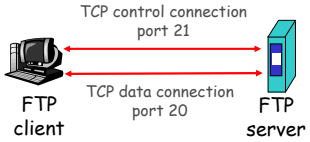
```
125 Data connection already open; transfer starting
200 Command OK
225 Data connection open
226 Closing data connection
227 Entering passive mode; srv. sends Ip_add.,port
230 User login OK
331 Username OK, password required
425 Can't open data connection
426 Connection closed; tranfer aborted
452 Error writing file
500 Syntax error; unrecognized command
501 Syntax error in parameters or arguments
502 Command not implemented
```


FTP: esempio trasferimento file



FTP: separate control, data connections

- FTP client contacts FTP server at port 21, specifying TCP as transport protocol
- Client obtains authorization over control connection
- Client browses remote directory by sending commands over control connection.
- 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.



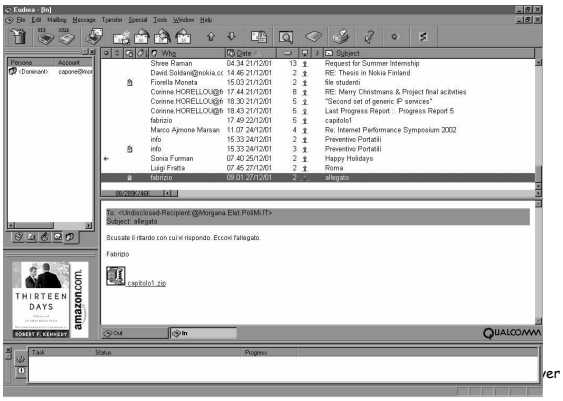
- Server opens a second TCP data connection to transfer another file.
- Control connection: "out of band"
- *FTP server maintains "state": current directory, earlier authentication*

Chapter 2 outline

- 2.1 Principles of app layer protocols
  - clients and servers
  - app requirements
- 2.2 Web and HTTP
- 2.3 FTP
- 2.4 Electronic Mail
  - SMTP, POP3, IMAP
- 2.5 DNS
- 2.9 Content distribution
  - Network Web caching
  - Content distribution networks
  - P2P file sharing

I servizi di IP: La Posta Elettronica

- L'e-mail è un applicazione che consente di inviare in modo asincrono messaggi testuali



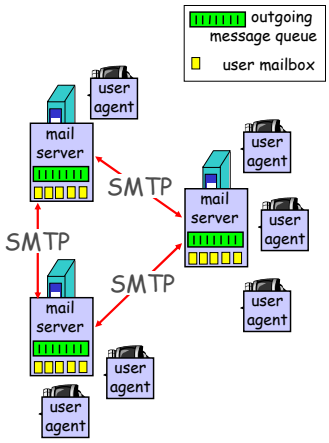
Electronic Mail

Three major components:

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

User Agent

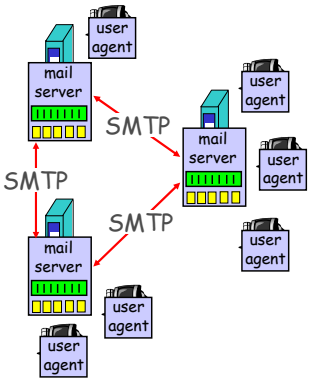
- a.k.a. "mail reader"
- composing, editing, reading mail messages
- e.g., Eudora, Outlook, elm, Netscape Messenger
- outgoing, incoming messages stored on server



Electronic Mail: mail servers

Mail Servers

- mailbox contains incoming messages for user
- message queue of outgoing (to be sent) mail messages
- SMTP protocol between mail servers to send email messages
  - client: sending mail server
  - "server": receiving mail server



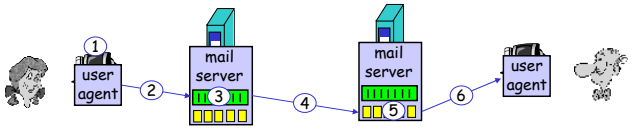
SMTP

J.B. Postel, "Simple Mail Transfer Protocol," RFC 821, August 1982.

- E' un protocollo testuale
- richiede che anche il corpo dei messaggi sia ASCII
  - i documenti binari devono essere convertiti in ASCII
- quando un server riceve un messaggio da un user agent
  - mette il messaggio in una coda
  - apre una connessione TCP con la porta 25 del server del destinatario
  - trasferisce il messaggio

Scenario: Alice sends message to Bob

1) Alice uses UA to compose message and "to" `bob@someschool.edu`
2) Alice's UA sends message to her mail server; message placed in message queue
3) Client side of SMTP opens TCP connection with Bob's mail server
4) SMTP client sends Alice's message over the TCP connection
5) Bob's mail server places the message in Bob's mailbox
6) Bob invokes his user agent to read message



Electronic Mail: SMTP [RFC 2821]

- direct transfer: sending server to receiving server
- three phases of transfer
  - handshaking (greeting)
  - transfer of messages
  - closure
- command/response interaction
  - commands: ASCII text
  - response: status code and phrase

Colloquio tra client e server SMTP

```
S: 220 hamburger.edu          ← Service ready
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: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection
```


SMTP: final words

- SMTP uses persistent connections
- SMTP requires message (header & body) to be in 7-bit ASCII
- SMTP server uses `CRLF.CRLF` to determine end of message

Comparison with HTTP:

- HTTP: pull
- SMTP: push
- both have ASCII command/response interaction, status codes
- HTTP: each object encapsulated in its own response msg
- SMTP: multiple objects sent in multipart msg

Formato dei messaggi

D.H. Crocker, "Standard for the Format of ARPA Internet Text Messages," RFC 822, August 1982.

- Il formato dei messaggi inviati (comando DATA) è specificato
- Alcuni header standard precedono il corpo del messaggio vero e proprio

```
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Request of information
<black line>
<Body>
.
```


Multipurpose Internet Mail Extensions (MIME)

•"Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies," RFC 2045, Nov. 1996.
•"Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types," RFC 2046, Nov. 1996.

- Estensione MIME al RFC 822 ha come scopo principale quello di consentire il trasferimento di messaggi non ASCII

MIME version

method used to encode data

multimedia data type, subtype, parameter declaration

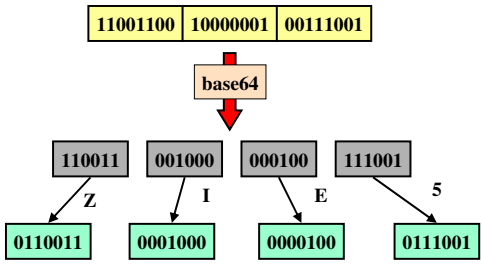
encoded data

```
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
```

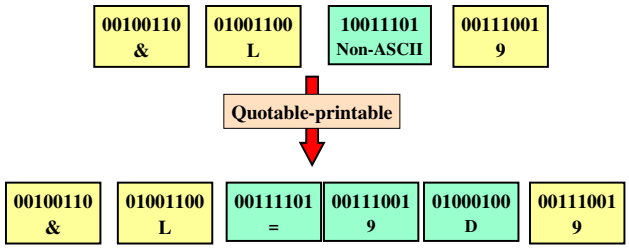
Multipurpose Internet Mail Extensions (MIME)

- Codifiche:
  - Base64:
    - Le sequenze di bit da trasferire sono divise in gruppi di 24 bit
    - Ogni gruppo è in diviso in 4 sotto-gruppi di 6 bit
    - Ad ogni gruppo viene aggiunto uno zero in testa e vengono trasferiti come caratteri ASCII



Multipurpose Internet Mail Extensions (MIME)

- Quoted-printable
  - Le sequenze di bit da trasferire sono divise in gruppi di 8 bit
  - Se una sequenza corrisponde ad un carattere ASCII è inviata così com'è
  - Altrimenti viene inviata come tre caratteri , "=" seguito dalla rappresentazione esadecimale del byte



Multipurpose Internet Mail Extensions (MIME)

- MIME consente anche il trasferimento di più oggetti come parti di uno stesso messaggio:

```
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Picture of yummy crepe with commentary
MIME-Version: 1.0
Content-Type: multipart/mixed; Boundary=StartOfNextPart
--StartOfNextPart
Dear Bob,
Please find a picture of an absolutely scrumptious crepe.

--StartOfNextPart
Content-Transfer-Encoding: base64
Content-Type: image/jpeg
base64 encoded data .....

--StartOfNextPart
Let me know if you would like the recipe.
.
```


MIME types

`Content-Type: type/subtype; parameters`

Text
- example subtypes: `plain`, `html`

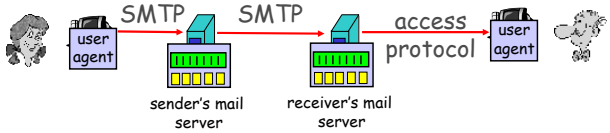
Image
- example subtypes: `jpeg`, `gif`

Audio
- exampe subtypes: `basic` (8-bit mu-law encoded), `32kadpcm` (32 kbps coding)

Video
- example subtypes: `mpeg`, `quicktime`

Application
- other data that must be processed by reader before "viewable"
- example subtypes: `msword`, `octet-stream`

Mail access protocols



- SMTP: delivery/storage to receiver's server
- Mail access protocol: retrieval from server
  - POP: Post Office Protocol [RFC 1939]
    - authorization (agent <-->server) and download
  - IMAP: Internet Mail Access Protocol [RFC 1730]
    - more features (more complex)
    - manipulation of stored msgs on server
  - HTTP: Hotmail , Yahoo! Mail, etc.

POP3

- Il server POP3 ascolta sulla porta TCP 110. Il client che vuole usufruire del servizio di posta si connette con il mail server su questa porta.
- Fasi:
  - Greetings
  - Scambio di comandi/risposte (status keywords additional info)
    - AUTHORIZATION STATE (il client deve identificarsi con il POP3 server inviando una username e password)
    - TRANSACTION STATE (il client richiede azioni da parte del server, e.g.RETR retrieve message, DELE delete message)
    - UPDATE STATE (dopo una QUIT il server POP3 rilascia eventuali risorse acquisite durante lo stato di transaction –e.g. lock-e dice goodbye)
  - Chiusura della connessione

Per inviare messaggi al mail server: SMTP!!

POP3 protocol

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
```


POP3 (more) and IMAP

More about POP3

- Previous example uses "download and delete" mode.
- Bob cannot re-read e-mail if he changes client
- "Download-and-keep": copies of messages on different clients
- POP3 is stateless across sessions

IMAP

- Keep all messages in one place: the server
- Allows user to organize messages in folders
- IMAP keeps user state across sessions:
  - names of folders and mappings between message IDs and folder name