Internet infrastructure

Prof. dr. ir. André Mariën

File Transfer Protocol

- The objectives:
  - sharing of files to shield a user from variations in file storage systems among hosts
  - to transfer data reliably and efficiently.
- History
  - First proposed file transfer mechanisms in 1971
  - TCP as the underlying protocol: RFC 765
    - specification of FTP for use on TCP
  - Now: RFC 959

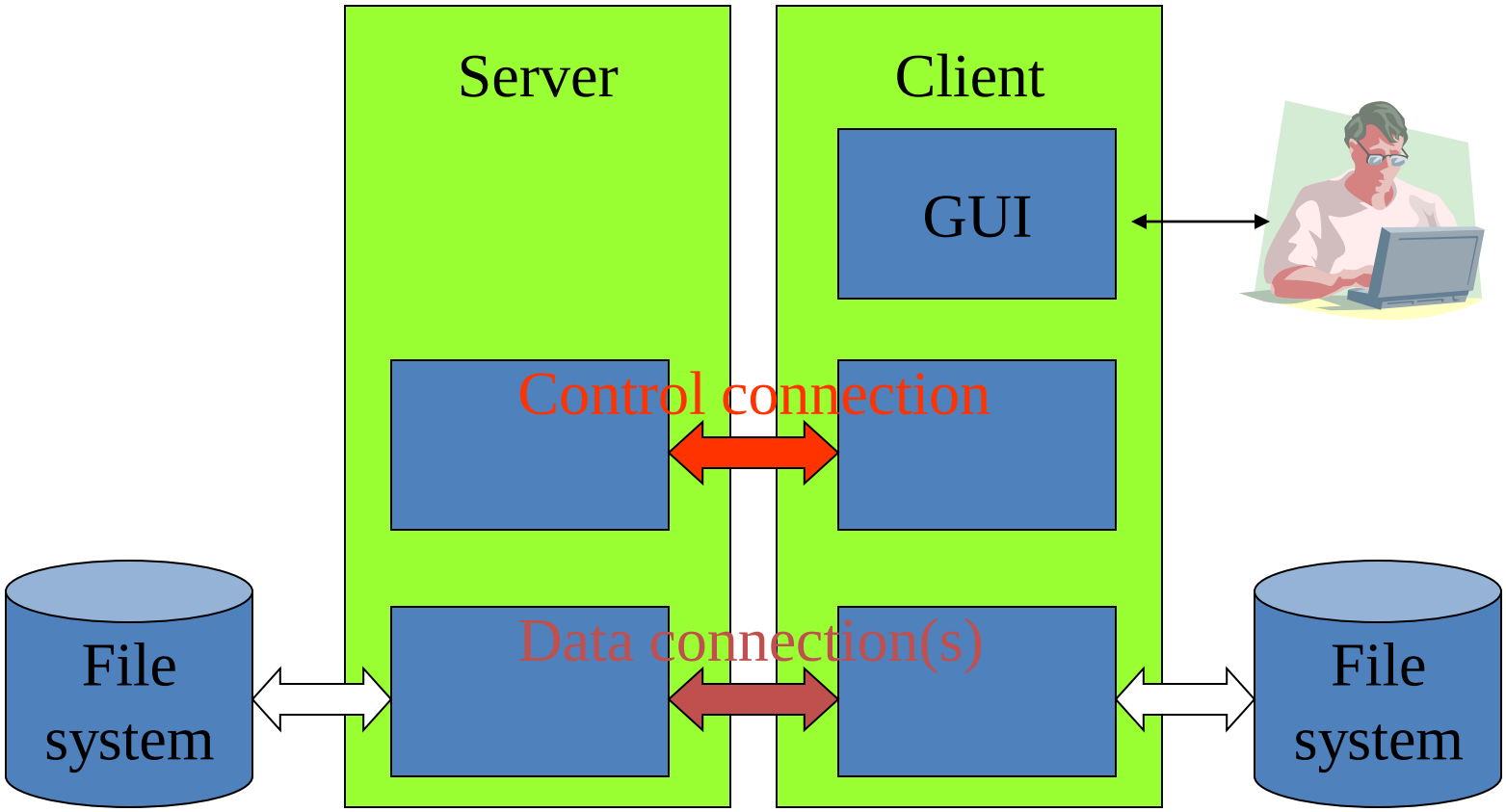
Connections

- Control connection
  - This connection follows the Telnet Protocol.
- Data connection
  - A full duplex connection over which data is transferred, in a specified mode and type.
  - The data transferred may be a part of a file, an entire file or a number of files.
  - The path may be between a server and a user, or between two servers.

Data transfer functions

- Files are transferred only via the data connection.
- The control connection is used for the transfer of commands, which describe the functions to be performed, and the replies to these commands

The FTP model



Notes

- The data connection may be used in either direction
- The data connection need not exist all of the time

Mechanics of transferring data

- set up the data connection to the appropriate ports
- choose the parameters for transfer

- client and server have a default data port
- The user-process default data port is the same as the control connection port
- The server-process default data port is the port adjacent to the control connection port (20).The transfer byte size is 8-bit bytes

Passive data transfer process

- "listen" on the data port prior to sending a transfer request command
- The FTP request command determines the direction of the data transfer (GET/PUT)

- The server initiates the data connection to the port
- When the connection is established, the data transfer begins
- The server sends a confirming reply to the client

Port usage

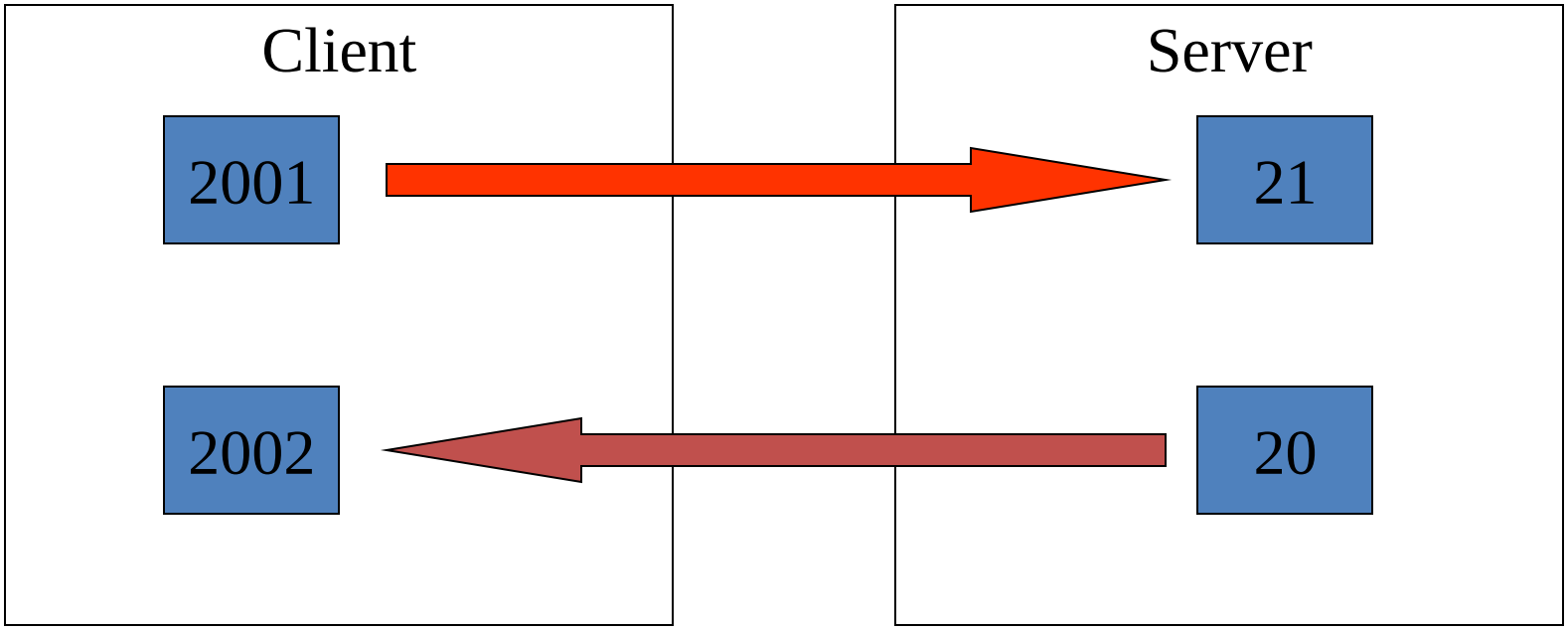
- the use of the default data ports must be supported

- only the client can initiate a change to non-default ports.
- client: request alternate data port via the PORT command
- client: request a non-default server side data port with the PASV command

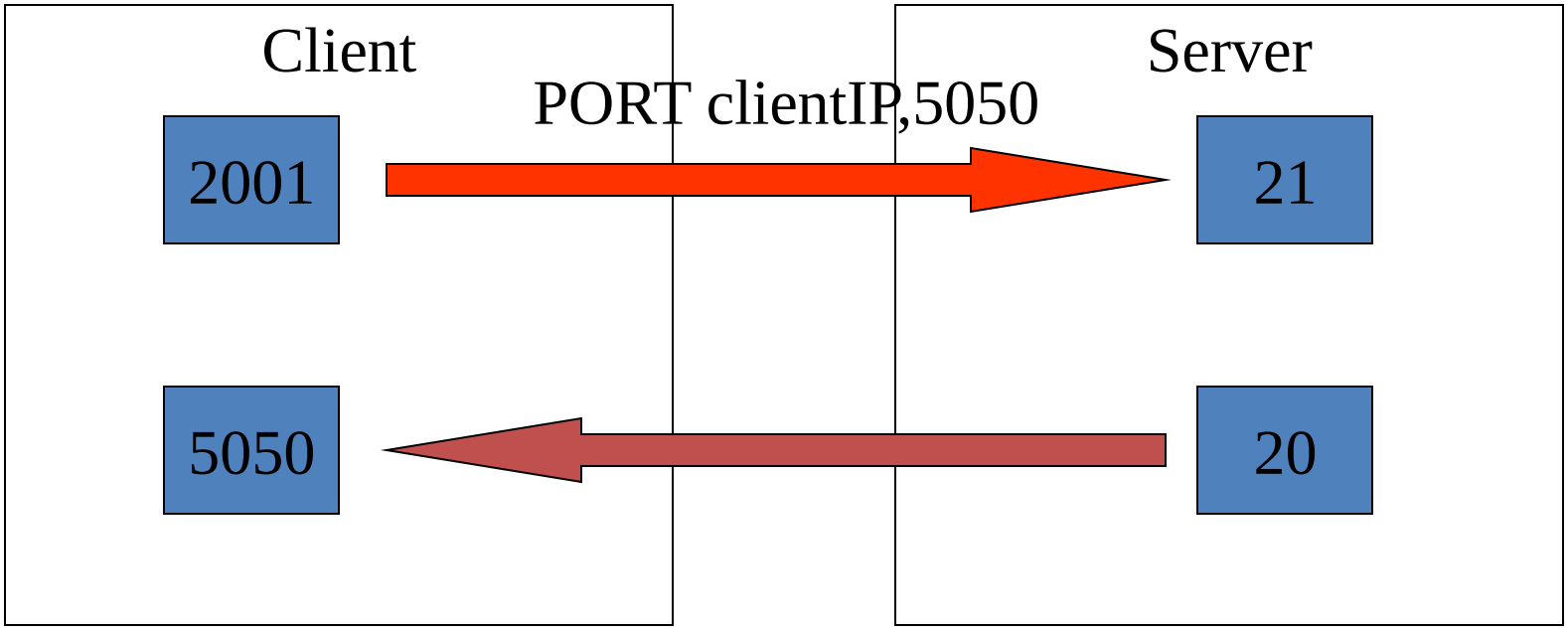
Reuse of the Data Connection

- When using the stream mode of data transfer the end of the file must be indicated by closing the connection
- problem: multiple files transferred in one session

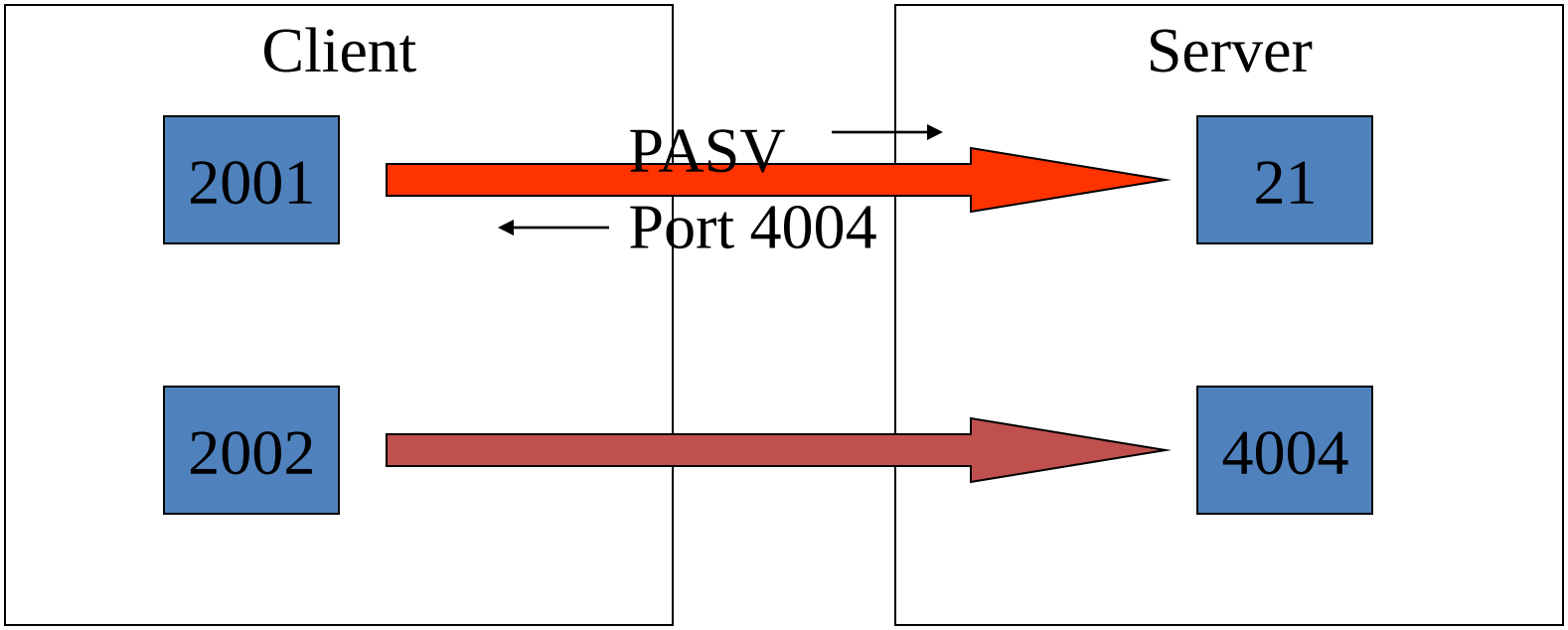
Default ports



PORT command



PASV command



Character set conversions

- convert characters into the standard NVT-ASCII representation when transmitting text between dissimilar systems
- The sending and receiving sites perform the necessary transformations between the standard representation and their respective internal representations.

Transmitting binary data

- FTP: very limited data type representations
- Transformations desired beyond this: performed by the user directly
- representations
  - ASCII: the default type
    - must be accepted by all FTP implementations
    - intended primarily for the transfer of text files
  - EBCDIC

Transmitting binary data

– IMAGE
  - sent as contiguous bits packed into the 8-bit transfer bytes
  - The receiving site must store the data as contiguous bits.
  - intended for the efficient storage and retrieval of files
  - for the transfer of binary data
  - It is recommended that this type be accepted by all FTP implementations.

File structure

- FTP allows the structure of a file to be specified.
- Three file structures are defined
  - “file” structure
    - no internal structure; the file is considered to be a continuous sequence of data bytes
    - the default structure
  - “record” structure
    - the file is made up of sequential records
  - “page” structure
    - the file is made up of independent indexed pages.

Three transmission modes

- block: formats the data and allows for restart procedures
- Compressed: also compresses the data for efficient transfer
- stream: passes the data with little or no processing

Block transmission mode

- The file is transmitted as a series of data blocks
- data blocks are preceded by one or more header bytes
- The header bytes: a count field (16 bits) + descriptor code (8 bits)
- The descriptor code defines:
  - last block in the file (EOF)
  - last block in the record (EOR)
  - restart marker
  - (flags in byte)

File transfer functions

- FTP COMMANDS

Commands: access control

- USER NAME (USER) <name>
- PASSWORD (PASS) <password>
- CHANGE WORKING DIRECTORY (CWD) <dir>
- CHANGE TO PARENT DIRECTORY (CDUP)
- REINITIALIZE (REIN)
- LOGOUT (QUIT)

Commands: Transfer parameters

- DATA PORT (PORT) <ip1>,<ip2>,<ip3>,<ip4>,<p1>,<p2>
- PASSIVE (PASV)
- REPRESENTATION TYPE (TYPE) ("A"|"E"|"I")
  - A: ASCII
  - I: Image
  - E: EBCDIC
  - default is "A"

Commands: Transfer parameters

- FILE STRUCTURE (STRU) ("F"|"R")
  - F: File (no record structure)
  - R: Record structureThe default structure is File.
- TRANSFER MODE (MODE) ("S"|"B")
  - S: Stream
  - B: Block
  - The default is Stream.

Commands: service

- RETRIEVE (RETR) <pathname>
- STORE (STOR) <pathname>
- STORE UNIQUE (STOU)
  - file is to be created in the current directory under a name unique to that directory.
  - 250 response: must include the name generated
- APPEND (with create) (APPE) <pathname>
- ALLOCATE (ALLO) <nrbytes>

Commands: service

- RENAME FROM (RNFR) <oldpathname>
- RENAME TO (RNTO) <newpathname>
  - Belong together
- ABORT (ABOR)
- DELETE (DELE) <pathname>
- REMOVE DIRECTORY (RMD) <pathname>
- MAKE DIRECTORY (MKD) <pathname>
- PRINT WORKING DIRECTORY (PWD)

C

- LIST (LIST) (<filepathname>|<dirpathname>)
- NAME LIST (NLST) (<dirpathname>)
- SITE PARAMETERS (SITE)
  - services specific to his system
  - specification of their syntax: do “HELP SITE”
- SYSTEM (SYST)
  - type of operating system at the server

Commands: service

- STATUS (STAT)
- HELP (HELP) [<comand name>]
- NOOP (NOOP)

MINIMUM IMPLEMENTATION

- TYPE: ASCII, Non-print
- MODE: Stream
- STRUCTURE: File, Record
- COMMANDS: USER, QUIT, PORT, TYPE, MODE, STRU, RETR, STOR, NOOP

NNTP

- RFC 977: Network News Transfer Protocol
- NNTP specifies a protocol for
  - the distribution
  - inquiry
  - retrieval
  - posting
- TCP port 119

NNTP characteristics

- news articles are stored in a central database
  - mail: individual databases
- subscription based
- Supports:
  - Indexing
  - cross-referencing
  - expiration of aged messages

News versus other options

- bulletin boards
- mailing lists

internet Mailing Lists problems:

- subscription
- performance
- centralization

News

- multiple hosts
- synchronization
  - remote
  - local
- centralized storage and archiving for LAN users

article format: RFC 850

- Compare
- Mail:
  - SMTP
  - RFC822
- NEWS:
  - NNTP
  - RFC850

News Distribution

- = netnews system for synchronization
- method of exchanging articles between cooperating hosts.
- may be propagated from host to host by flooding
  - send all you have
  - let receiver either
    - discard doubles
    - clean and store

NNTP: interactive mechanism

- for deciding which articles are to be transmitted
- changes:
  - new news groups
    (see NEWGROUPS command)
  - new articles list
    (see NEWNEWS command)
- article content
- transfer of posts from client to server if not yet there
- 7-bit ascii (like SMTP)

Command/response operation

- two kinds
  - textual
    - after status respons
    - ended with "." (see SMTP)
  - status
- state during operation:
  - current article pointer (CAP)

Response codes

- general coding system
- The first digit of the response code
  - 1xx - Informative message
  - 2xx - Command ok
  - 3xx - Command ok so far, send the rest of it.
  - 4xx - Command was correct, but couldn't be performed for some reason.
  - 5xx - Command unimplemented, or incorrect, or a serious program error occurred.

The second digit of the response code

- x0x - Connection, setup, and miscellaneous messages
- x1x - Newsgroup selection
- x2x - Article selection
- x3x - Distribution functions
- x4x - Posting
- x8x - Nonstandard (private implementation) extensions
- x9x - Debugging output

ARTICLE, BODY, HEAD, STAT commands

- same parameters, return different portions of article
  - ARTICLE: all
  - BODY: only body
  - HEAD: only head
  - STAT: only reply code
- forms:
  - ordinal number
  - article ID: "<...>“
  - Based on CAP (no parameter)

ARTICLE command

- article <...>: full article
- article nnn: full article, changes CAP

- response:
- nnn <id>
- ...
- .

ARTICLE,...: Responses

- 220 n <a> article retrieved - head and body follow
- 221 n <a> article retrieved - head follows
- 222 n <a> article retrieved - body follows
- 223 n <a> article retrieved - request text separately
- 412 no newsgroup has been selected
- 420 no current article has been selected
- 423 no such article number in this group
- 430 no such article found

GROUP command

- GROUP ggg.ggg.ggg
- Responses: if 2xx: CAP set to first article
- 211 n f l s group selected
  - n = estimated number of articles in group
  - f = first article number in the group
  - l = last article number in the group
  - s = name of the group
- 411 no such news group

IHAVE command

- IHAVE <messageid>
- Used for synchronization

IHAVE: difference with POST

- intended for use in transferring already-posted articles between hosts
- Responses
  - 235 article transferred ok
  - 335 send article to be transferred. End with <CR-LF>.<CR-LF>
  - 435 article not wanted - do not send it
  - 436 transfer failed - try again later
  - 437 article rejected - do not try again

LIST command

- LIST
  - 215 list of newsgroups follows
  - group last first y/n
  - …
  - group last first y/n
  - .
- y/n: posting allowed (in theory)

NEWGROUPS command

- NEWGROUPS date time [GMT] [<distributions>]
- date format: YYMMDD
- time format: HHMMSS in the server's timezone
- distributions: prefix, like net for net.*, alt for alt.*
  - example: <net,alt,bin>

NEWNEWS command

- NEWNEWS newsgroups date time [GMT] [<distribution>]
- newsgroups:
  - wildcard expansion
  - match negation (!alt.binaries.*)
- response:
  - 230 list of new articles by message-id follows...
  - .

NEXT command

- NEXT
- Side effect: CAP++
- Responses223 n a article retrieved - request text separately
- 412 no newsgroup selected
- 420 no current article has been selected
- 421 no next article in this group

POST command

- POST response code
  - 340: start posting
  - 440: posting prohibited
- data: format specified by RFC850
- Responses (ok)
  - 240 article posted ok
  - 340 send article to be posted. End with <CR-LF>.<CR-LF>

POST: failures

- Response codes
  - 440 posting not allowed
  - 441 posting failed
- Note: server initially tells posting status
  - 200 server ready - posting allowed
  - 201 server ready - no posting allowed

QUIT command

- QUIT

Next: examples

- Sample Conversations, straight from RFC

Example - relative access with NEXT

- [connection]
- (client asks for a current newsgroup list)
- C:LIST
- S:215 list of newsgroups follows
- S:net.wombats 00543 00501 y
- S:net.unix-wizards 10125 10011 y
- (more information here)
- S:net.idiots 00100 00001 n
- S:.

example

- (client selects a newsgroup)
- C:GROUP net.unix-wizards
- S:211 104 10011 10125 net.unix-wizards group selected
- (there are 104 articles on file, from 10011 to 10125)

example

- (client selects an article to read)
- C:STAT 10110
- S:223 10110 <23445@sdcsvax.ARPA> article retrieved - statistics only
- (article 10110 selected, its message-id is <23445@sdcsvax.ARPA>)

example

- (client examines the header)
- C:HEAD
- S:221 10110 <23445@sdcsvax.ARPA> article retrieved - head follows
- (text of the header appears here)
- S:.

Example: client wants to see the text body of the article

- C:BODY
- S:222 10110 <23445@sdcsvax.ARPA> article retrieved – body follows
- (body text here)
- S:.

Example: client selects next article in group

- C:NEXT
- S:223 10113 <21495@nudebch.uucp> article retrieved - statistics only
- (article 10113 was next in group)

Example: absolute article access with ARTICLE

- C:GROUP msgs
- S:211 103 402 504 msgs Your new group is msgs
- (there are 103 articles, from 402 to 504)
- C:ARTICLE 401
- S:423 No such article in this newsgroup
- C:ARTICLE 402
- S:220 402 <4105@ucbvax.ARPA> Article retrieved, text follows
- S:(article header and body follow)
- S: .

Example: HEAD

- C:HEAD 403
- S:221 403 <3108@mcvax.UUCP> Article retrieved, header follows
- S:(article header follows)
- S:.
- C:QUIT
- S:205 VAX news server closing connection. Goodbye.

Example - NEWGROUPS command

- (client asks for new newsgroups since April 3, 1985)
- C:NEWGROUPS 850403 020000
- S:231 New newsgroups since 03/04/85 02:00:00 follow
- S:net.music.gdead
- S:net.games.sources
- S:.

Example

- C:GROUP net.music.gdead
- S:211 0 1 1 net.music.gdead Newsgroup selected
- (there are no articles in that newsgroup, and the first and last article numbers should be ignored)
- C:QUIT
- S:205 Imaginary Institute news server ceasing service. Bye!

Example - posting a news article

- S:200 VAX news server ready, posting allowed.
- C:POST
- S:340 Continue posting; Period on a line by itself to end
- C:(transmits news article in RFC850 format)
- C:.
- S:240 Article posted successfully.

Example – news distribution

- (client asks for new newsgroups since 2 am, May 15, 1985)
- C:NEWGROUPS 850515 020000
- S:235 New newsgroups since 850515 follow
- S:net.fluff
- S:net.lint
- S:.

Example: client asks for new news

- Articles since 2 am, May 15, 1985:
  - C:NEWNEWS * 850515 020000
  - S:230 New news since 850515 020000 follows
  - S:<1772@foo.UUCP>
  - S:<87623@baz.UUCP>
  - S:<17872@GOLD.CSNET>
  - S:.

Example: article by ID

- client asks for article 1772@foo.UUCP:
  - C:ARTICLE <1772@foo.UUCP>
  - S:220 <1772@foo.UUCP> All of article follows
  - S:(sends entire message)
  - S:.
- client asks for article 87623@baz.UUCP:
- C:ARTICLE <87623@baz.UUCP>
  - S:220 <87623@baz.UUCP> All of article follows
  - S:(sends entire message)
  - S:.

Example: news transfer

- client offers an article it has received: recently
  - C:IHAVE <4105@ucbvax.ARPA>
  - S:435 Already seen that one, where you been?

Example: news transfer

- client offers another article:
  - C:IHAVE <4106@ucbvax.ARPA>
  - S:335 News to me! <CRLF.CRLF> to end.
  - C:(sends article)
  - C:.
  - S:235 Article transferred successfully. Thanks.
- (or)
  - S:436 Transfer failed.