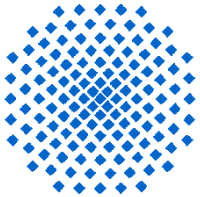


Universität Stuttgart

Institute of Parallel and
Distributed Systems (IPVS)

Universitätsstraße 38
D-70569 Stuttgart

Net-based Applications: Protocols

Adnan Tariq

(Based on the slides of Dr. Frank Dürr)

Protocols

- Telnet Protocol
- File Transfer Protocol (FTP)
- Simple Mail Transfer Protocol (SMTP)
- Multipurpose Internet Mail Extensions (MIME)
- Post Office Protocol (POP)

TELNET Protocol

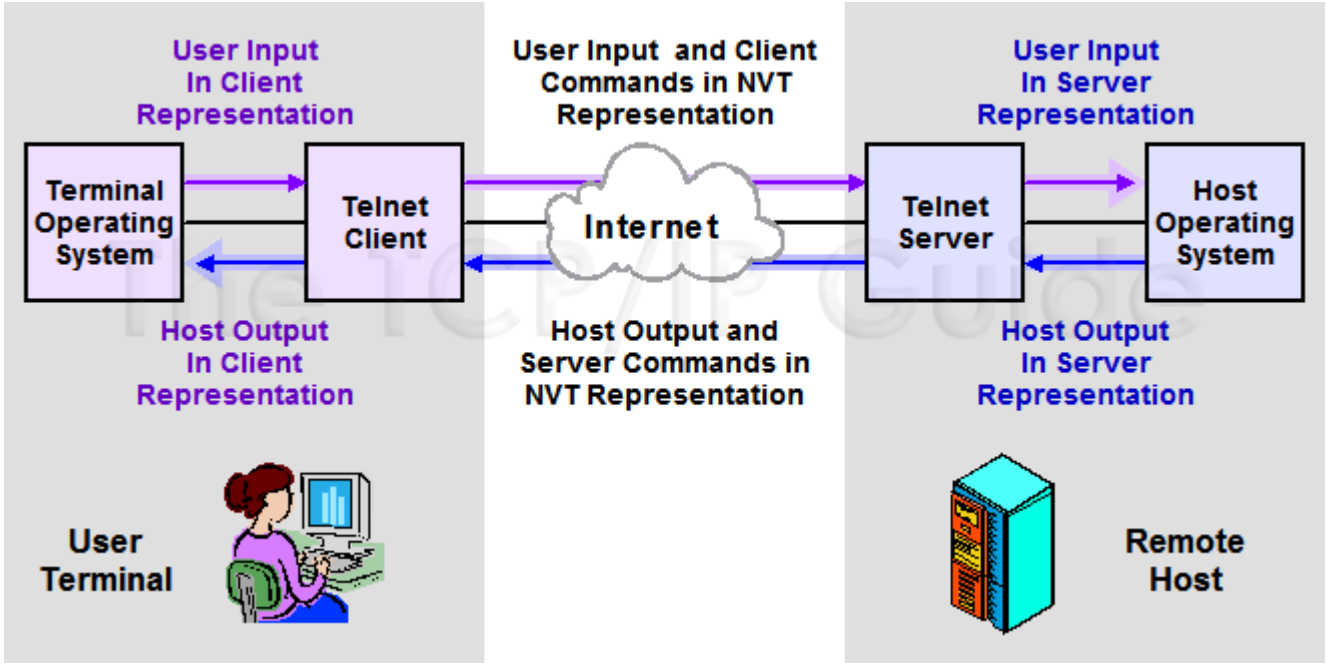
The purpose of the TELNET Protocol is to provide a fairly general, bidirectional, eight-bit byte oriented communications facility

- Standard described in RFC 854 (1983)
- Additional RFCs describe options
- TCP connection is used to transmit data with interspersed TELNET control information
- Main purpose: Remote logins via Internet
- Main ideas:
  - Network Virtual Terminal (NVT) concept
  - Negotiated options

TELNET - Network Virtual Terminal (NVT)

- An NVT is an imaginary device which provides a standard, network-wide, intermediate representation of a canonical terminal
- NVT uses 7-bit ASCII codes for characters
- Characters are transmitted as 8 bit bytes, MSB set to zero
- Hosts are responsible for translating characters to/from NVT
- Initially, each end of the connection is assumed to originate and terminate a NVT
- Option negotiation then allows for different terminals to be used in order to take advantage of all their features

TELNET - Network Virtual Terminal (NVT)



TELNET - Commands

- Commands (option negotiation commands, for example) are incorporated into the data stream
- Most significant bit is set to distinguish commands from regular 7-bit ASCII characters (NVT)
- Commands always start with the IAC (Interpret As Command) character (decimal 255) – necessary because usage of data formats other than 7-bit ASCII characters can be negotiated

Telnet Synch Mechanism:
Some control functions such as AO (Abort Output) and IP (Interrupt) require immediate action. This may cause difficulties if data is held in buffers awaiting input requests at a (possibly misbehaving) remote process. To overcome this problem, a DM (Data Mark) character is sent in a TCP Urgent segment. This tells the receiver to examine the data stream for "interesting" characters such as IP, AO and AYT (Are You There) up to DM.

TELNET - Negotiated Options (1)

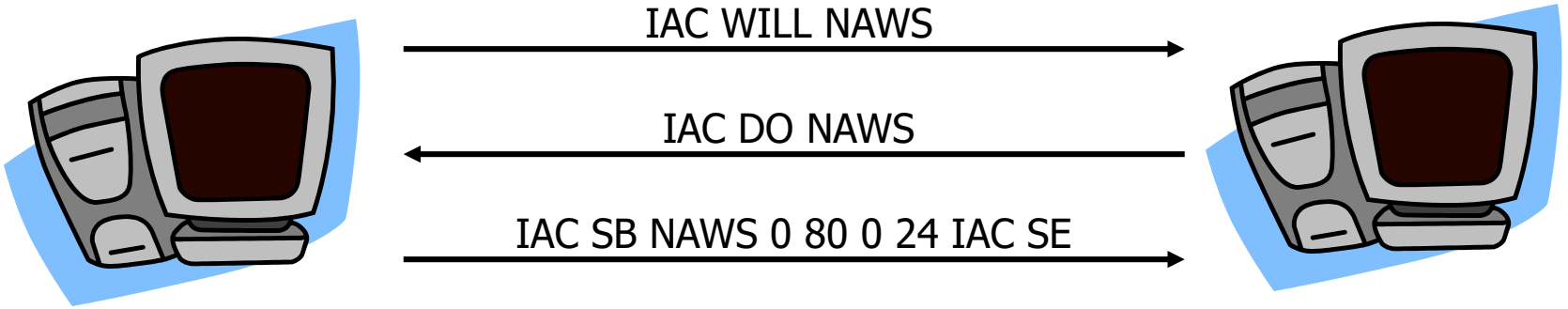
- Negotiate additional options (more sophisticated or just different conventions for the specific connection)
- Options are described in separate RFCs
- Negotiation takes place at the beginning of the connection and can also take place at any time during the connection

Some options:

Code	Option	RFC
1	echo	857
3	suppress go ahead (for full-duplex connections)	858
24	terminal type	1091
31	window size	1073

TELNET - Negotiated Options (2)

NAWS = Negotiate About Window Size



	Meaning
WILL	Sender wants to do something
DO	Sender wants receiver to do something
WON'T	Sender doesn't want to do something
DON'T	Sender doesn't want receiver to do something

*) only valid response

Sent	Received	Result
WILL	DO	✓
WILL	DON'T	×
DO	WILL	✓
DO	WON'T	×
WON'T	DON'T*	×
DON'T	WON'T*	×

TELNET - Sample Session

```
babaai@marvin ~ >telnet
telnet> toggle options
Will show option processing.
telnet> open trick
Trying 129.69.212.1...
Connected to trick.
Escape character is '^]'.
SENT DO SUPPRESS GO AHEAD
SENT WILL TERMINAL TYPE
SENT WILL NAWS
SENT WILL TSPEED
SENT WILL LFLOW
SENT WILL LINEMODE
SENT WILL NEW-ENVIRON
SENT DO STATUS
RCVD DO TERMINAL TYPE
RCVD DO NAWS
SENT IAC SB NAWS 0 80 (80) 0 24 (24)
RCVD DO XDISPLOC
SENT WONT XDISPLOC
RCVD DO NEW-ENVIRON
```

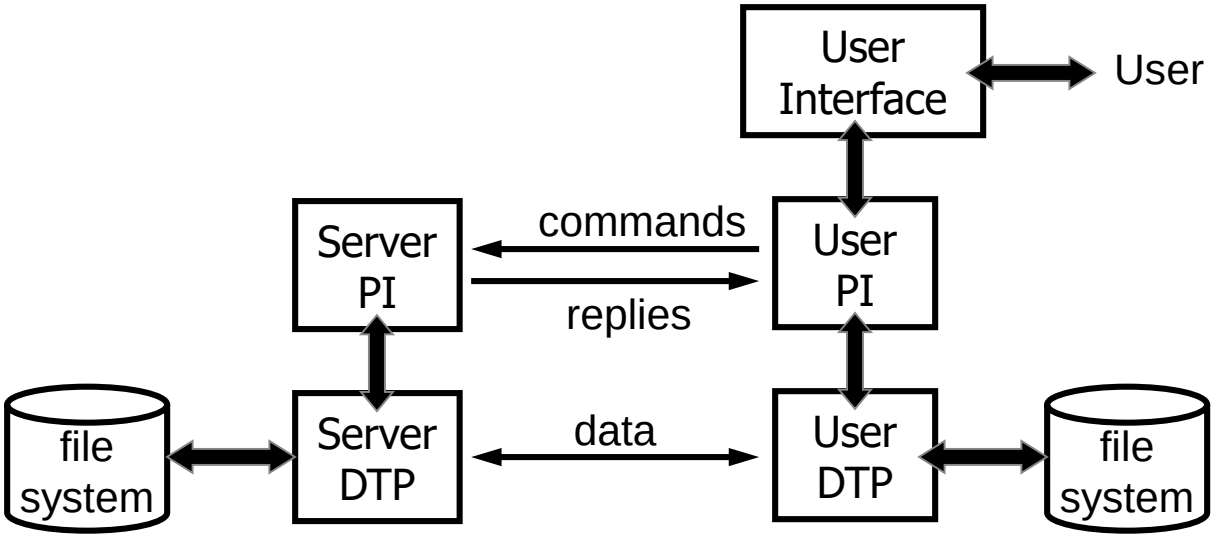
```
RCVD DO OLD-ENVIRON
SENT WONT OLD-ENVIRON
RCVD WILL SUPPRESS GO AHEAD
RCVD DONT TSPEED
RCVD DONT LFLOW
RCVD DONT LINEMODE
RCVD WONT STATUS
RCVD DONT XDISPLOC
RCVD DONT OLD-ENVIRON
RCVD IAC SB TERMINAL-TYPE SEND
SENT IAC SB TERMINAL-TYPE IS "VT100"
RCVD IAC SB NEW-ENVIRON SEND
SENT IAC SB NEW-ENVIRON IS


SunOS 5.6

RCVD WILL ECHO
SENT DO ECHO
RCVD DO ECHO
SENT WONT ECHO
RCVD DONT ECHO
login:
```

FTP (File Transfer Protocol)

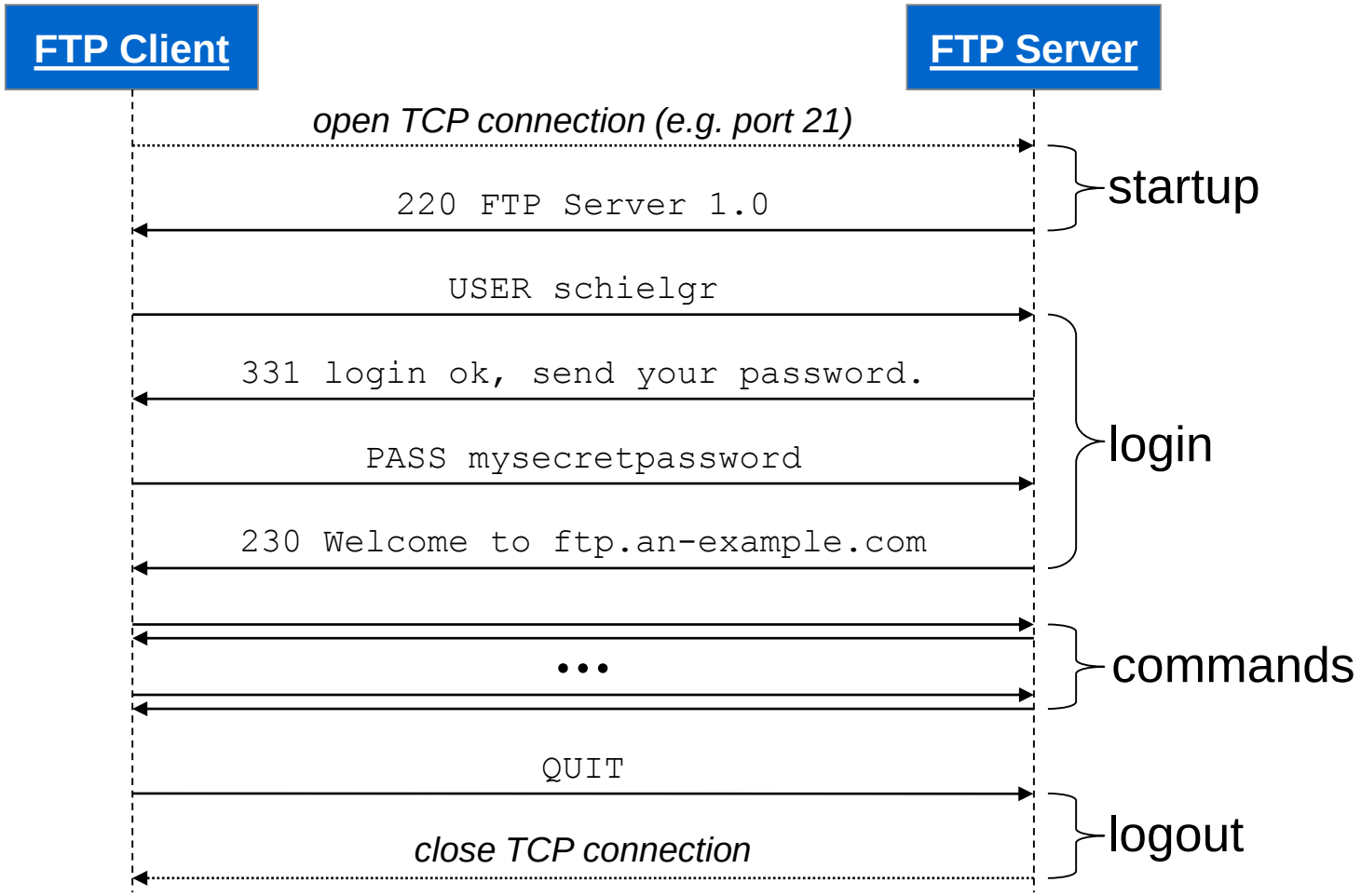
- FTP standard is defined in RFC 959 (1985)
- FTP uses two separate connections (control and data)
  - ✓ Different QoS can be used for data and control connection
  - ✓ No need for embedding commands into the data stream



FTP - Control Connection (1)

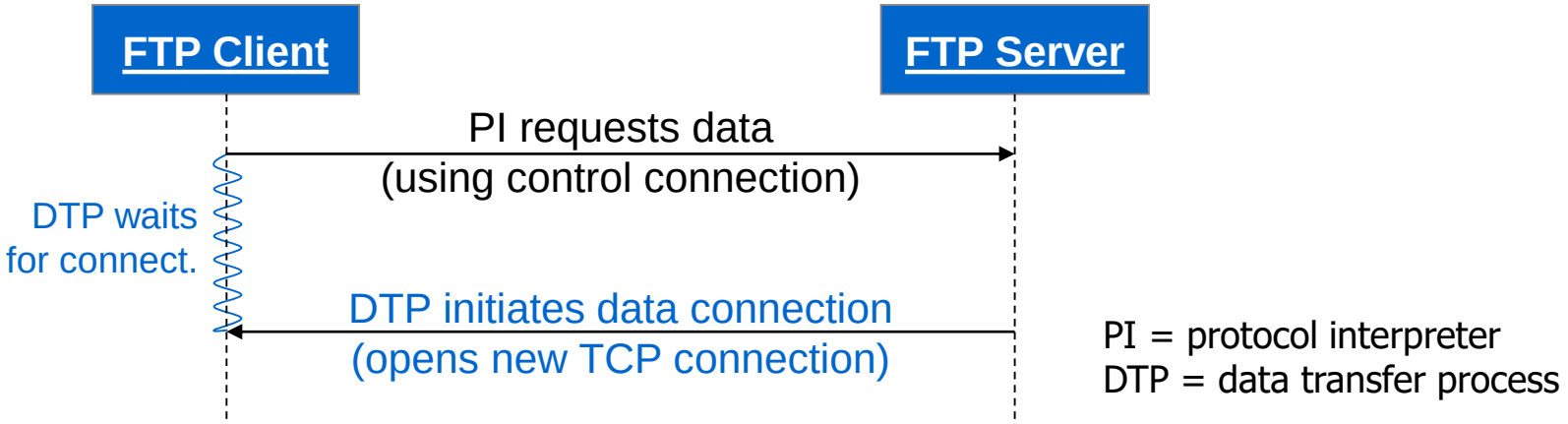
- TELNET used for control connection
- FTP commands specify
  - **Parameters for data connection** (data port, transfer mode, representation type, and structure)
  - **Nature of file system operation** (store, retrieve, append, delete, etc.)
- Commands transmitted as NVT ASCII strings starting with three or four upper case NVT ASCII characters followed by optional arguments and a CR/LF pair at the end (`CWD /pub`)
- Replies start with 3 digit NVT ASCII numbers with an optional message (`250 CWD command successful`)

FTP - Control Connection (2)



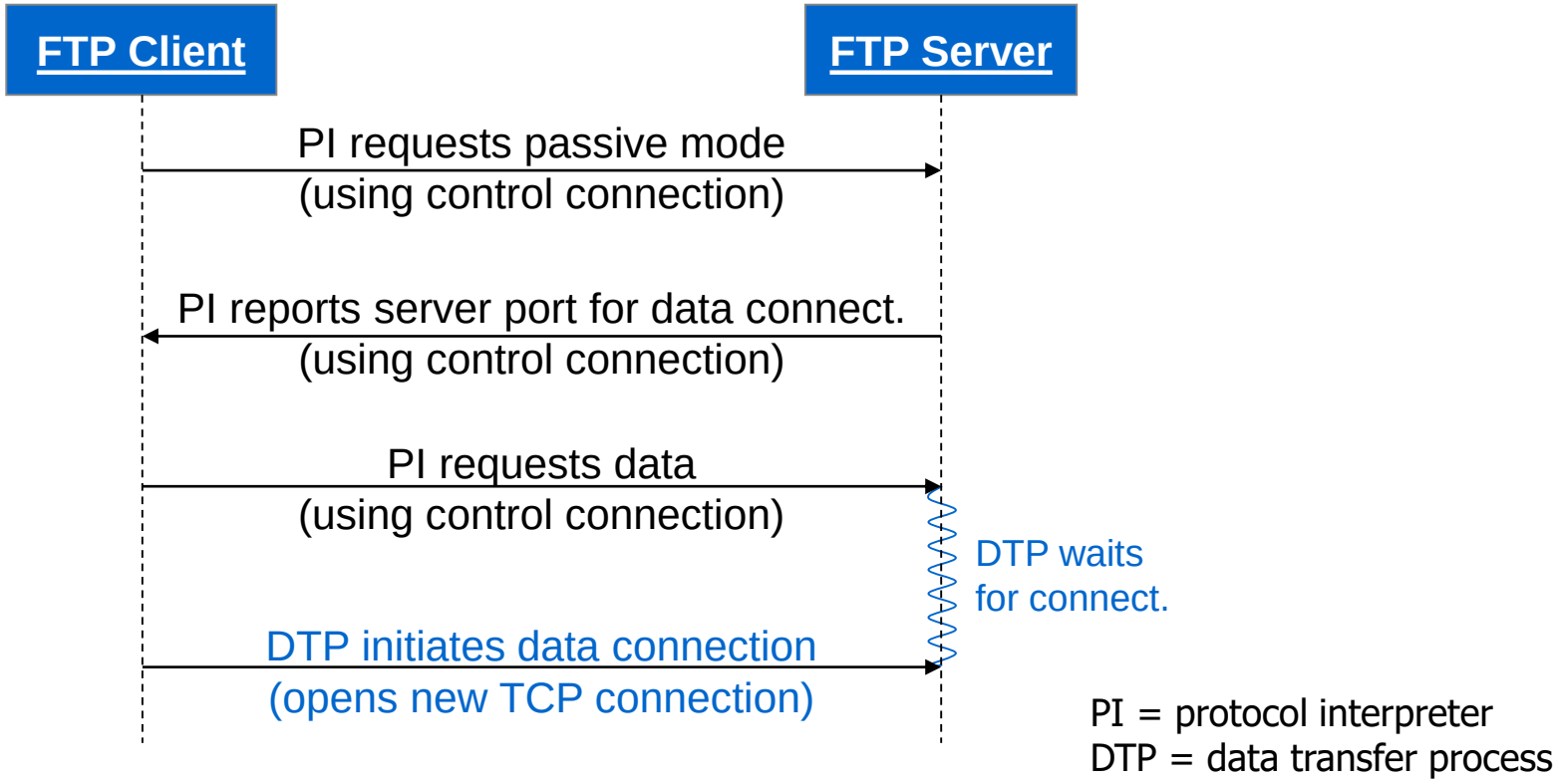
FTP - Data Connection: Active Mode

- Used to transfer files and directory listings
- Needs to be open only during data transfer →open/close as needed
- File type can either be ASCII or image (binary) – RFC defines additional data types, but these are not likely to be encountered in practice
- **Opening data connection (active mode):**



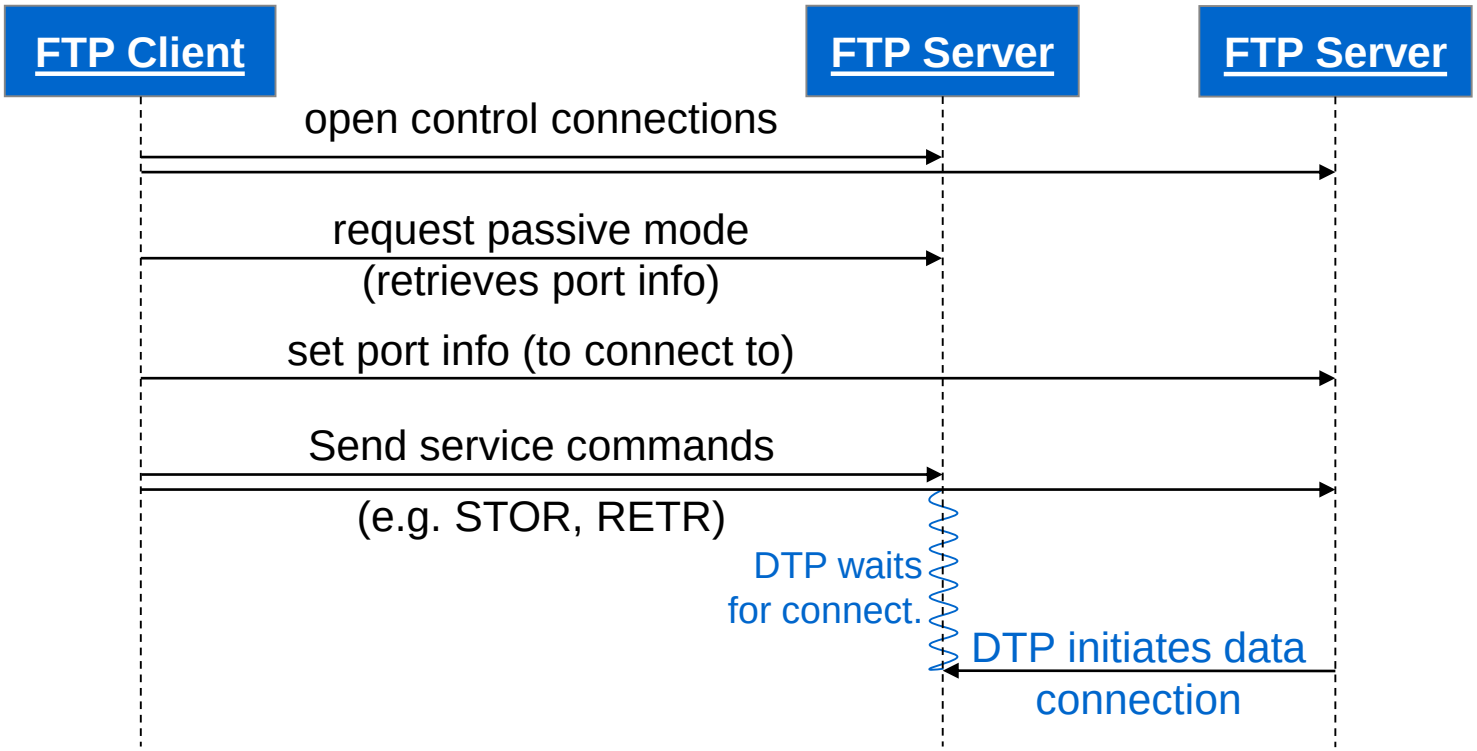
FTP - Data Connection: Passive Mode

- Problem: client-side firewall prevents server to open connection to client
- **Opening data connection (passive mode):**



FTP - Server To Server Data Transmissions

- Until now: **client/server** transmissions
- FTP additionally allows **server/server** transmissions



FTP - Sample Session (1)

```
babaai@marvin ~ >ftp ftp.uni-stuttgart.de
Connected to ftp.uni-stuttgart.de (129.69.2.131).
220-
220-                    Welcome to ftp.uni-stuttgart.de, the
220-                ******  I N F O and S O F T  Server  ******
220-           Computational Center, University of Stuttgart, Germany
220-
220 info6 FTP server (Version wu-2.6.1(1) Wed Nov 28 04:02:31 GMT 2001) ready.
Name (ftp.uni-stuttgart.de:babaai): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password:
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd /pub/misc
250 CWD command successful.
ftp> pwd
257 "/pub/misc" is current directory.
ftp>
```

FTP - Sample Session (2)

```
ftp> ls
227 Entering Passive Mode (129,69,2,131,92,150)
150 Opening ASCII mode data connection for /bin/ls.
total 624
-rw-rw-r--   1 rusiadm  info       567875 May 13 05:20 FILES
-rw-rw-r--   1 rusiadm  info        69543 May 13 05:20 FILES.gz
drwxrwsr-x   4 rusfrank info           25 Apr  2 07:56 games
drwxrwsr-x   2 rusiadm  info            6 Dec 11 13:59 rec
226 Transfer complete.
ftp> get FILES
local: FILES remote: FILES
227 Entering Passive Mode (129,69,2,131,156,211)
150 Opening BINARY mode data connection for FILES (567875 bytes).
226 Transfer complete.
567875 bytes received in 0.241 secs (2.3e+03 Kbytes/sec)
ftp> binary
200 Type set to I.
ftp>
```

Electronic Mail

Ali.Baba@studi.informatik.uni-stuttgart.de

local part — hostname part

Mail Transfer Agent (MTA):

- Programs that run on mail servers and listen for incoming mail from local users and MTAs on other servers (usually on port 25)
- MTA decides what to do with incoming mail:
  - Save locally for pickup by users (*mail spool*)
  - Forward to another MTA
- Intermediate MTAs are known as *relays*
- *sendmail* and *exim* are well-known MTAs

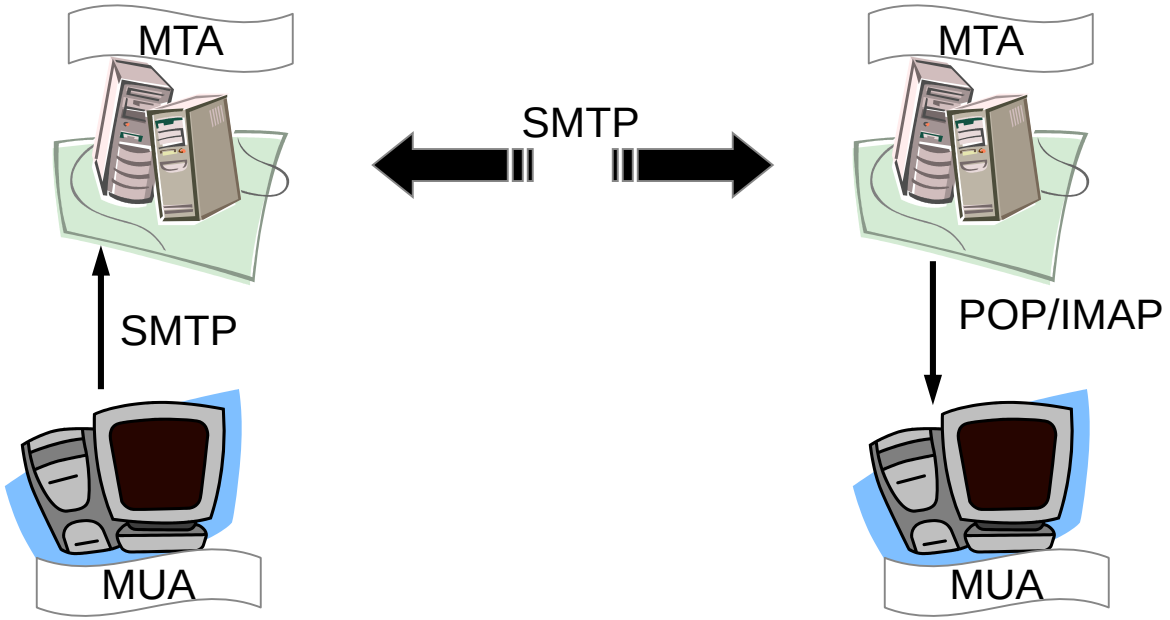
Mail User Agents (MUA):

- Mail software used to read and write mail (Mutt, Pine, Outlook, …)

SMTP (Simple Mail Transfer Protocol)

The objective of Simple Mail Transfer Protocol (SMTP) is to transfer mail reliable and efficiently.

- Standard defined in RFC 821
- Format (Internet Message Format) defined in RFC 822



SMTP - Message Format

- Envelope (used by MTAs)
- Headers (used by MUAs)

To:	Recipient’s address
Cc:	Carbon copy
Bcc:	Blind carbon copy
From:	Sender’s address
Reply-To:	Address to reply to
Subject:	Subject of the message

- Additional headers may be used (X-…)
- Empty line
- Message text (7-bit ASCII)

SMTP - Sample Session

```
babaai@marvin ~ >telnet studi 25
Trying 129.69.212.4...
Connected to studi.
Escape character is '^]'.
220 studi.informatik.uni-stuttgart.de ESMTP Exim 3.22 #7 Mon, 13 May 2002
  14:13:38 +0200
HELO marvin
250 studi.informatik.uni-stuttgart.de Hello babaai at
  marvin.informatik.uni-stuttgart.de [129.69.212.19]
MAIL FROM: babaai@studi.informatik.uni-stuttgart.de
250 <babaai@studi.informatik.uni-stuttgart.de> is syntactically correct
RCPT TO: Ali.Baba@studserv.uni-stuttgart.de
250 <Ali.Baba@studserv.uni-stuttgart.de> verified
DATA
354 Enter message, ending with "." on a line by itself
Subject: Hello World!

Hello World.
.
250 OK id=177Eix-0004ZY-00
```

SMTP - Problems & Extensions

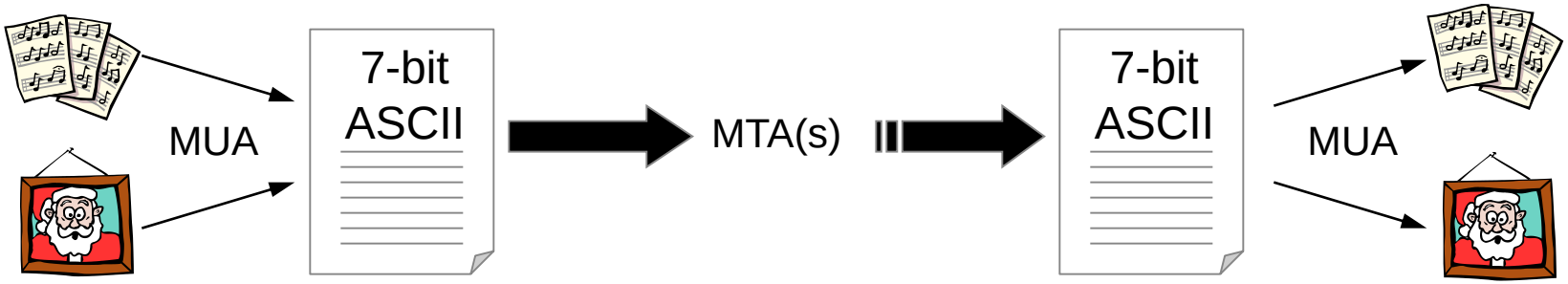
- Message length limited to 64 kB
- Only 7-bit ASCII characters may be used
- No authentication
- Mailstorms (infinite loops)
- Timeouts

Improvements:

- ESMTP (Extended SMTP)
- SMTP AUTH
- MIME (Multipurpose Internet Mail Extensions)
- …

MIME (Multipurpose Internet Mail Extensions)

- Defined in RFCs 2045-2049
- Provides support for international languages, multimedia messages, multipart messages, …
- MIME provide a structure and rules for the encoding of non-ASCII messages into the usual RFC 822 format
- Only the MUAs have to be modified, not the MTAs
- MIME can be extended to support new contents



MIME - Header Fields

Field	Example
MIME-Version:	1.0
Content-Description:	Mail message body
Content-Transfer-Encoding:	QUOTED-PRINTABLE
Content-Disposition:	inline
Content-Type:	TEXT/PLAIN; charset=US-ASCII

MIME - Encodings (1)

Quoted-printable

- Intended to represent data that largely consists of octets that correspond to printable characters in the US-ASCII character set
- 7 bit ASCII
- All characters > 127 are represented by their hexadecimal value after escape character “=“
  - Example: Frank D=FCrr
  - “=“ character in text is replaced by “=3D”
- Encoded form of the data remains largely recognizable by humans

MIME - Encodings (2)

Base64

- Information is considered a data stream
- 65-character subset of US-ASCII is used*
- 6 bits can be represented per printable character
- Extra 65th character "=" is used for padding
- 24-bit groups of input bits are represented as output strings of 4 encoded characters
  - Each character sent as one byte → 33% more space required
- Line breaks are ignored

*) Subset is carefully chosen – it is represented identically in all versions of ISO 646, including US-ASCII, and all characters in the subset are also represented identically in all versions of EBCDIC ⇨ better portability)

Base64 Alphabet (Code)

Value	Char	Value	Char	Value	Char	Value	Char
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	/

- Character “=“ used as pad-character

MIME - Content Types (1)

Type	Subtype	Description
text	plain	plain, unformatted text
	richtext	formatted text (defined in RFCs 1341/1896)
image	jpeg	image in JPEG format
	gif	image in GIF format
audio	basic	audio
video	mpeg	MPEG encoded video
application	octet-stream	uninterpreted binary data or information
	PostScript	data in PostScript format

- New media types can be introduced as described in RFC 2048
- Examples: text/html, application/pdf, application/msword, audio/mpeg

MIME - Content Types (2)

Composite content types:

Type	Subtype	Description
multipart	mixed	generic mixed set of parts
	alternative	same data in multiple formats
	parallel	parts intended to be viewed simultaneously
	digest	each part is a complete RFC822 message (parts have default type of message/rfc822)
message	rfc822	content is a RFC822 message
	external-body	body is referenced from an external source
	partial	message has been fragmented for transmission

MIME - Sample Message

```
From: Ali Baba <babaai@studi.informatik.uni-stuttgart.de>
To: Ali.Baba@studserv.uni-stuttgart.de
Subject: Hello World
Message-ID: <20020520235147.A23513@studi.informatik.uni-stuttgart.de>
Mime-Version: 1.0
Content-Type: multipart/mixed; boundary="17pEHd4RhPHOinZp"
Content-Disposition: inline

--17pEHd4RhPHOinZp
Content-Type: text/plain; charset=us-ascii
Content-Disposition: inline

Hello World

--17pEHd4RhPHOinZp
Content-Type: application/powerpoint
Content-Disposition: attachment; filename="helloworld.ppt"
Content-Transfer-Encoding: base64

0M8R4KGxGuEAAAAAAAAAAAAAAAAAAAAAPgADAP7/CQAGAAAAAAAAAAAAAAANAAAA (...)
```

POP (Post Office Protocol)

Motivation:

- Recipient wants to read mail on local machine
- Local machine does not have necessary resources or network connection to permanently handle mail (mail server should be online 24/7)

⇨Mailbox and MUA are on different machines!

Solution:

- POP to transfer mail from server to local machine

POP:

- Current version: POP3 (defined in RFC 1225)
- POP3 server listens on port 110
- Provides authentication

POP3 - Sample Session

```
Server: +OK pop.myisp.de POP3 server ready
Client: USER babaai
Server: +OK
Client: PASS *
Server: +OK
Client: STAT
Server: +OK 2 1988
Client: LIST
Server: +OK
Server: 1 994
Server: 2 994
Server: .
Client: RETR 1
Server: +OK 994 octets
(... Server sends message ...)
Server:.
Client: DELE 1
Server: +OK
(...)
Client: QUIT
Server: +OK
```

2 messages of total size 1988 bytes in inbox

POP3 - Problems & Extensions

- POP3 is not intended to provide extensive manipulation operations of mail on the server
- Limited support for reading mail from different machines
- No encryption

⇨ POP/SSL (encryption)

⇨ IMAP (mail remains on server)