



Email Basics

BUPT/QMUL
2021-05-06



北京邮电大学

BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS

Electronic Engineering 



Agenda

- Brief introduction to email
- Components of email system
- Email Standards
- Summary



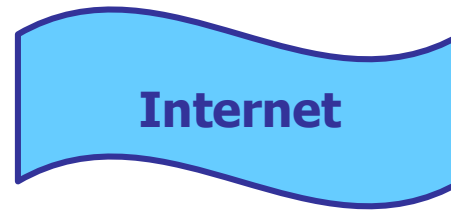
Brief Introduction To Email



What is Email?

- Electronic Mail (email, e-mail)
- Provides a means to send **electronic messages** from one person to another **asynchronously**
- One of the most popular applications on the Internet and one of the most important communication methods today
- Email service can be provided by
 - **ISPs**: @126.com, @163.com, @sina.com, @yahoo.cn, ...
 - Corporations and institutes: @baidu.com, @bupt.edu.cn, @ietf.org, ...
 - Bundled with other services: @139.com, @qq.com, ...

History Of Email



1971

1982

1984-1988

1988

1988

1993

1996

1996

...

the first
email in the
world

RFC 821(SMTP)
RFC 822

X.400
MOTIS

Eudora

POP
IMAP

MIME

Outlook Free emails ...

Ray Tomlinson

IETF

CCITT
ISO

Steve Dorner

IETF

IETF

Microsoft Hotmail etc. ...

Trends:

- Multiple data types other than text
- More Users
- More space, larger attachment

Newest development:

Oct. 2008

- RFC 5321 Simple Mail Transfer Protocol
- RFC 5322 Internet Message Format

What kind of methods for you to check email?

A

- Web

- Browser



B

- User Agent

- Foxmail
- Outlook
- Netease



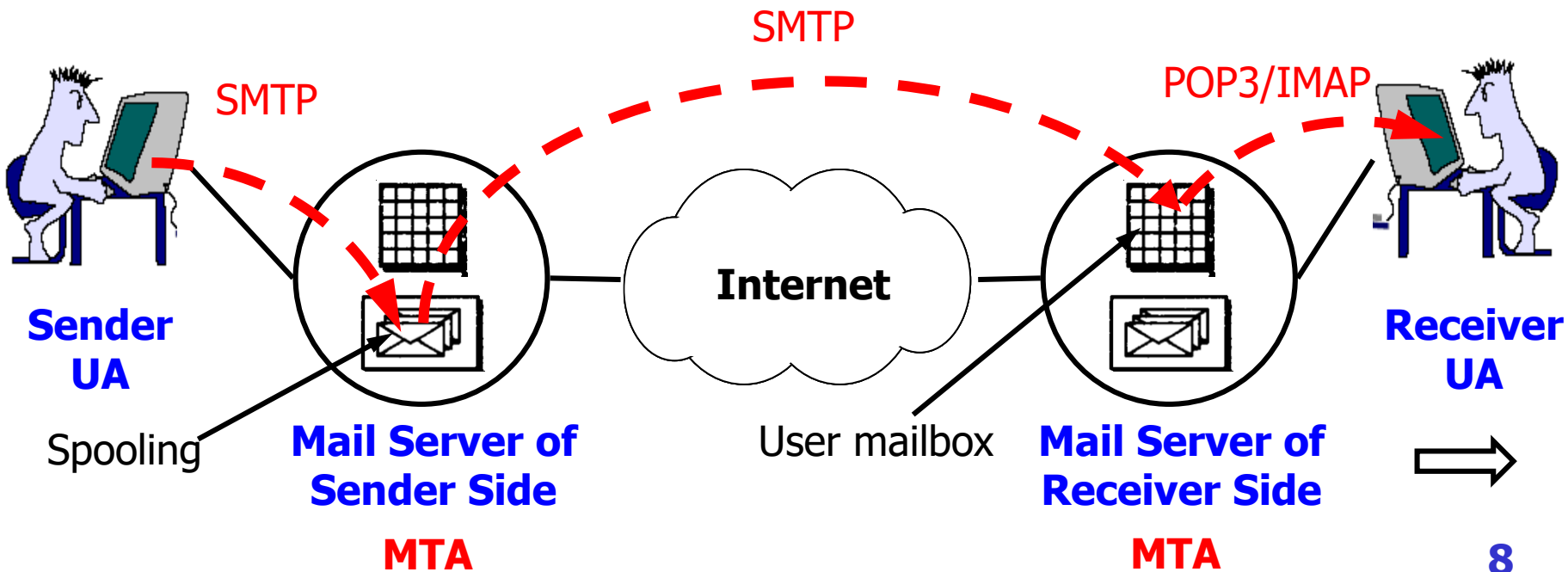
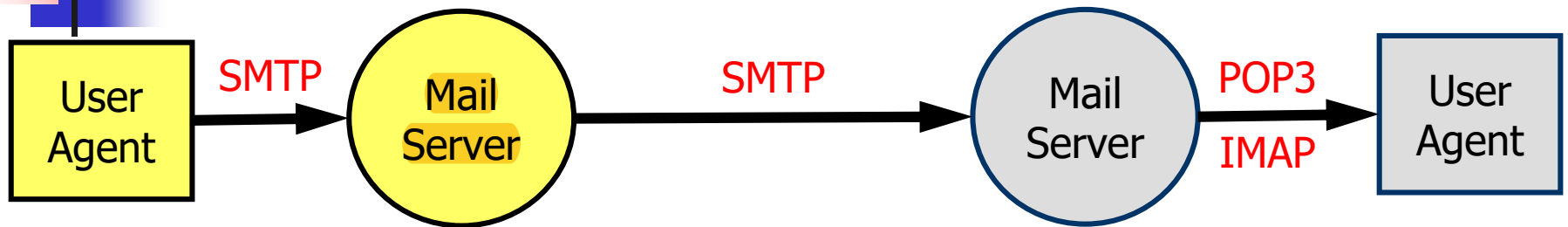
Outlook





Components Of Email System

Components Of Email System (1)



Components Of Email System (2)

- UA (User Agent)

- end-user mail program
- Interface between the end users and the email servers
- E.g., outlook, foxmail, ...



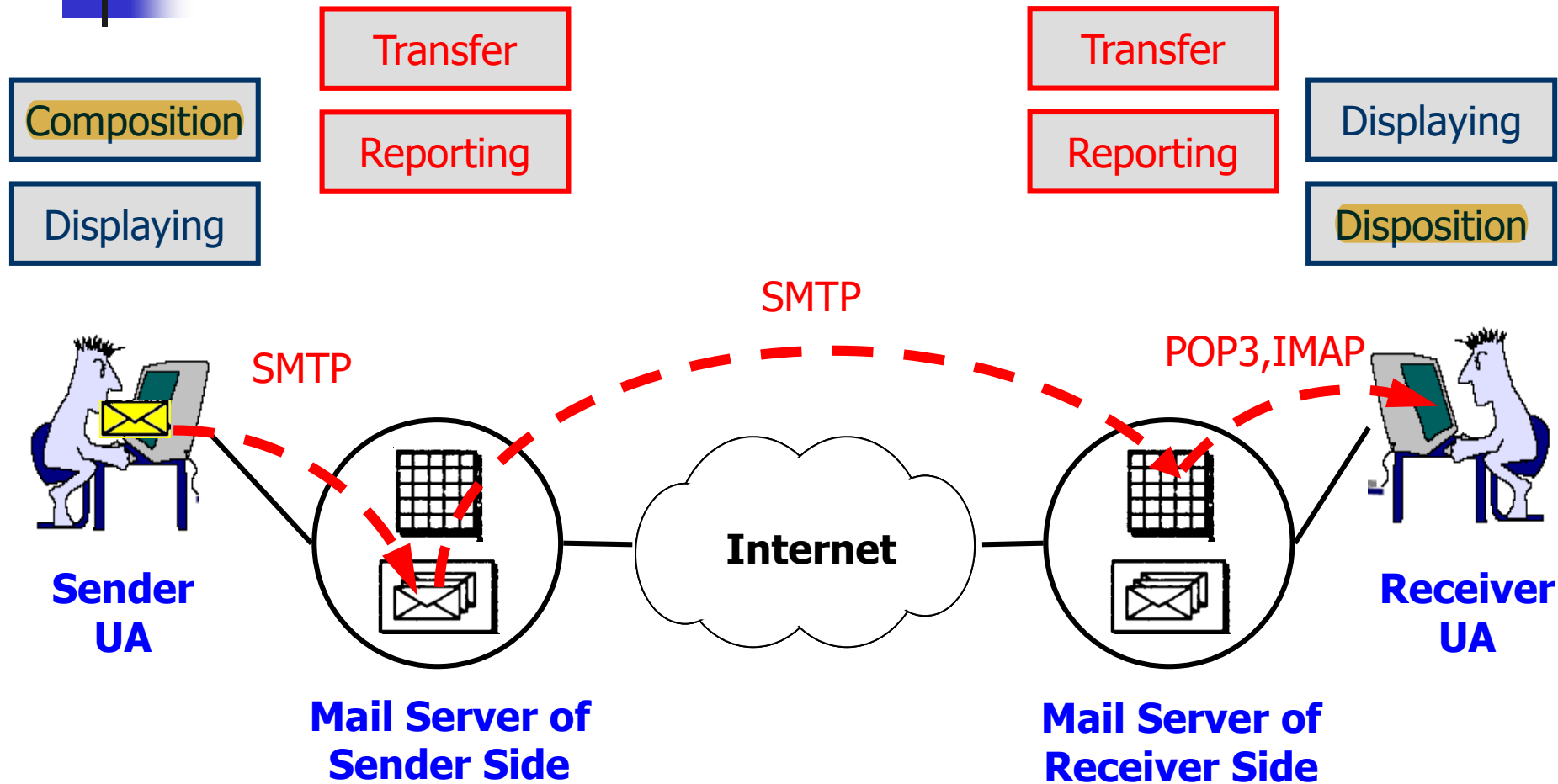
- Mail Server

- Responsible for transmitting/receiving emails and reporting status information about mail transferring to the mail sender
- **Both a client and a server**

- Email protocols

- SMTP: used for sending an email
- POP3/IMAP: used for receiving an email

Components Of Email System (3)





Basic Functions Of Email System

- *Composition* - refers to the process of creating messages and answers.
- *Transfer* - refers to moving messages from the originator to the recipient.
- *Reporting* - refers to the process of informing the originator what happened to the message. ■
- *Displaying* - means of showing the messages.
- *Disposition* - refers to what happened to the message after it has been read by the receiver.



Other Terminologies

- *Mailboxes* – created by the user to store incoming email.
- *Mailing lists* – means of sending identical emails to a group
- *MTA (Mail Transfer Agent)* – SMTP servers and clients provide a mail transport service





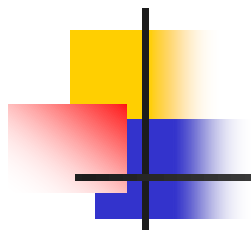
Email Sending

- To send an email message, a user must provide the **message**, the **destination address** and possibly some other parameters (e.g., security or security level)
- Email Address
 - Many user agents expect DNS addresses of the form ***mailboxname@domain***. (See RFC 5322)
 - Each email address is **unique** on the Internet because
 - ***Domain name*** is unique on the **Internet**
 - ***Mailboxname*** is unique in the **domain**



Email Reading

- Typically, when a user agent is started up, it will look at the user's mailbox for incoming email before displaying anything on the screen
- Then it may announce the number of messages in the mailbox or display a one-line summary of each one and wait for a command



Email Standards



Email Standards

- Internet Message Format
 - RFC 5322 etc.
- SMTP (Simple Mail Transfer Protocol)
 - RFC 5321 etc.
- POP (Post Office Protocol)
 - RFC 1939 etc.
- IMAP (Internet Message Access Protocol)
 - RFC 3501 etc.
- MIME (Multipurpose Internet Mail Extension)
 - RFC 2045-2049 etc.



Internet Message Format (1)

- Message envelop
 - contains whatever information is needed to accomplish transmission and delivery
- Message contents: comprise the object to be delivered to the recipient
 - Headers: from, to, subject, date, postmarks
 - Blank line
 - Body: actual message, may have many parts



Internet Message Format (2)

- Each header field consists of a single line of ASCII text containing the field name, a colon, and, for most fields, a value
 - eg. from:abc@gmail.com
- In normal usage, the User Agent builds a message and passes it to MTA
- The MTA then uses some of the header fields to construct the actual envelope
- User provides body & key headers, while mail system provides the rest

Internet Message Format

—Example RFC 5322 Message

- The **header** part is everything up to the blank line, and the **body** is everything after the blank line

Author of the message, required

Mailbox responsible for the actual transmission of the message, optional

the address(es) of the primary recipient(s) of the message

```
From: John Doe <jdoe@machine.example>
Sender: Michael Jones <mjones@machine.example>
To: Mary Smith <mary@example.net>
Subject: Saying Hello
Date: Fri, 21 Nov 1997 09:55:06 -0600
Message-ID: <1234@local.machine.example>
```






headers

```
-----
This is a message just to say hello.
So, "Hello".
```

body

Internet Message Format

—Field Definitions

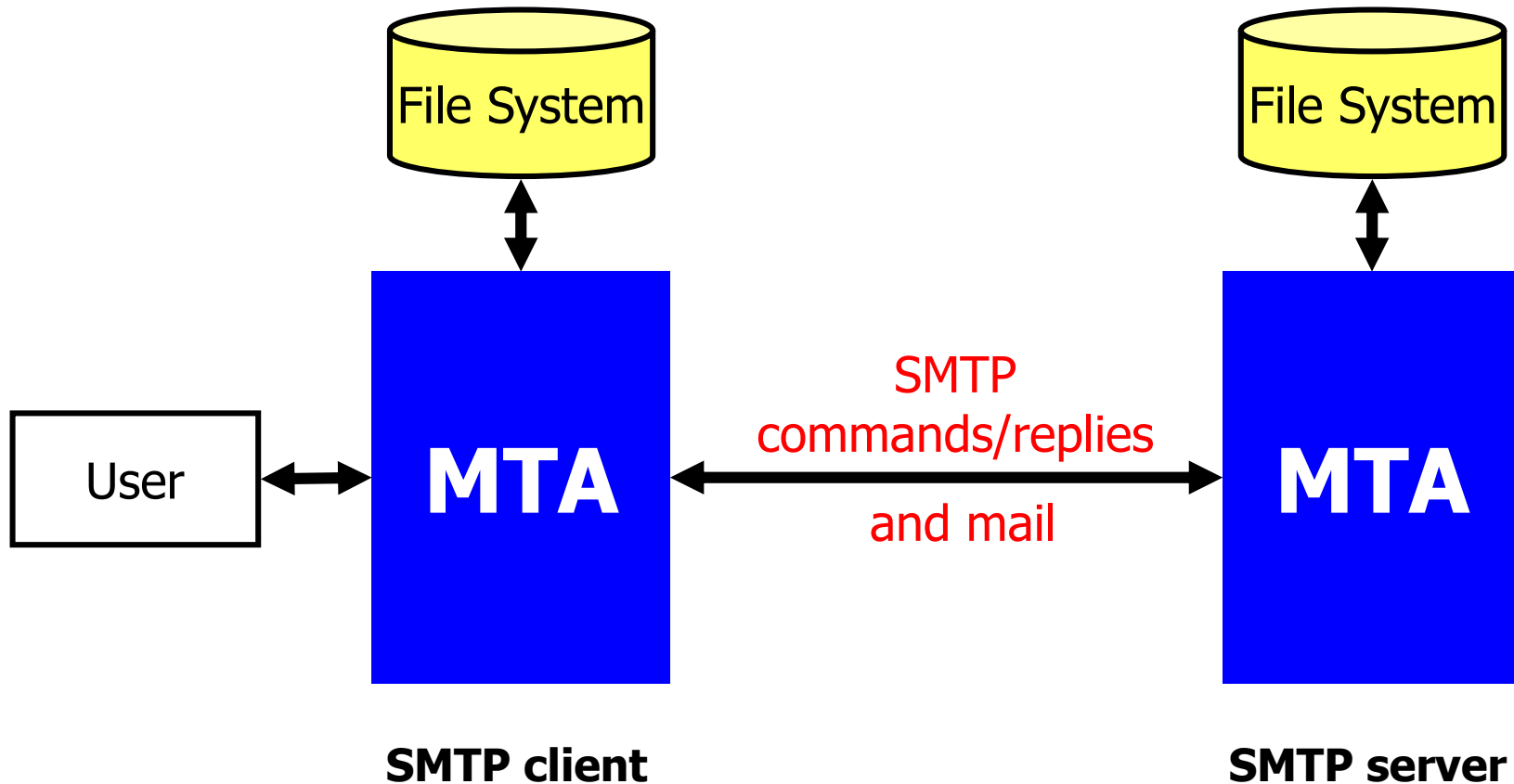
Categories	Header	Meaning
Originator fields	From:	Person or people who created the message
	Sender:	Email address of the actual sender
	Reply-to:	Email address to which replies should be sent
Destination address fields	To:	Email address(es) of primary recipient(s)
	Cc: 	Email address(es) of secondary recipient(s)
	Bcc: 	Email address(es) for blind carbon copies
The origination date field	Date:	The date and time the message was sent
Identification fields	Message-Id:	Unique number for referencing this message later
	References:	Other relevant Message-Ids
Information fields	Subject:	Short summary of the message for the one-line display
	Keywords:	User chosen keyword
Trace fields 	Received: 	Line added by each transfer agent along the route
	 Return-Path:	Can be used to identify a path back to the sender



SMTP

- The source machine establishes a **TCP** connection to Port **25** of the destination machine
- Listening to this port is a SMTP server
 - Accepts incoming connections
 - Receives messages over the connections
 - If the message cannot be delivered, an **error report containing the first part of the undeliverable message** is returned to the sender
- SMTP is a simple **ASCII** protocol
- After establishing the TCP connection to port 25, the sending machine, operating as a **client**, waits for the receiving machine operating as the **server to talk first**

SMTP Basic Model





SMTP Command Sequence – stages

- Connection establish
- Mail transfer
- Connection release



SMTP Commands: Basics

commands	description
HELO	■ identifies sender's Domain name
MAIL FROM:	■ starts a mail transaction and identifies the mail originator
RCPT TO:	■ identifies individual recipient. There may be multiple RCPT TO: commands
DATA	■ Sender ready to transmit body of the message. A series of lines of text, each ends with \r\n . A line containing only a period '.' indicates the end of the data
QUIT	■ close the connection



~~SMTP Commands: Extras~~

- **EHLO** (Extended Hello) – Same function with HELO.
But, in addition to domain name, address literal (eg. IPv4 address) is also supported
- VRFY - confirm that a name is a valid recipient
- EXPN - expand an alias (group email address)
- NOOP - send back a positive reply code
- RSET - abort current transaction
- HELP - cause the server to send helpful information to the client

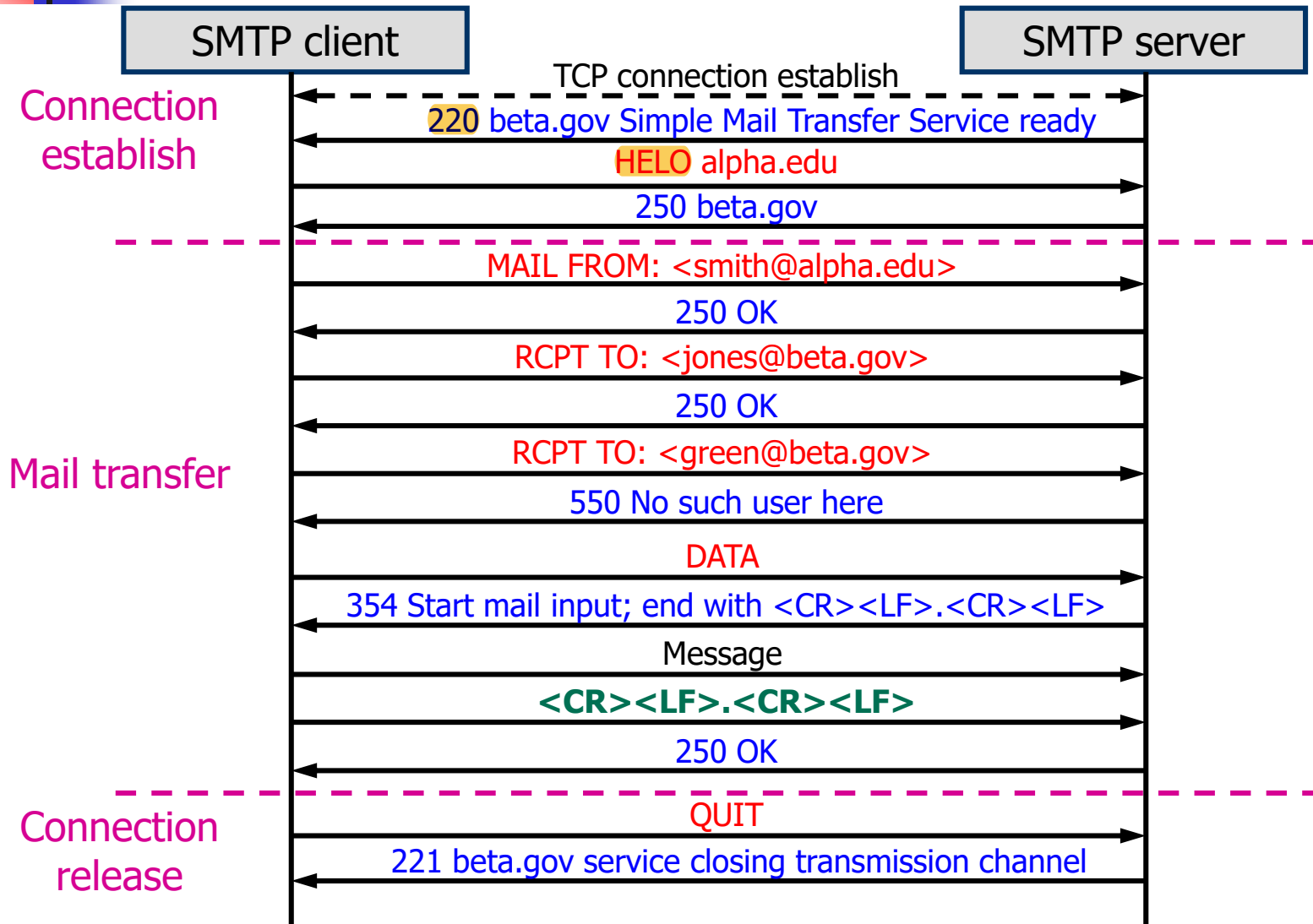


SMTP: Status Codes

- The Server responds with a 3 digit code that may be followed by text info
 - 2## -- Success
 - 3## -- Command can be accepted with more information
 - 4## -- Command was rejected, but error condition is temporary
 - 5## -- Command rejected, and the error condition is permanent, eg. Bad User!

SMTP Commands and Status codes – example

- smith@alpha.edu sends a mail to jones@beta.gov, green@beta.gov



Sending Email Through Telnet

```
C:\Documents and Settings\Administrator> telnet smtp.163.com 25
```

```
220 163.com Anti-spam GT for Coremail System (163com[20141201])
```

```
helo mail.163.com
```

```
250 OK
```

```
auth login
```

Base64 encoded "username:" and "Password:"

```
334 dXNlcm5hbWU6
```

```
Y2F0c2hpeQ==
```

Base64 encoded username – "catshiy"

```
334 UGFzc3dvcmQ6
```

Base64 encoded password – "123456"

```
MTIzNDU2
```

```
235 Authentication successful
```

```
mail from:<catshiy@163.com>
```

```
250 Mail OK
```

```
rcpt to:<catshiy@163.com>
```

```
250 Mail OK
```

```
data
```

```
354 Please start mail input.
```

```
subject:test email
```

Blank line: boundary between headers and body

```
this is only a test for sending email through telnet
```

```
.
```

Period: end of data

```
250 Mail queued for delivery.
```

```
quit
```

```
221 Bye.
```

```
失去了跟主机的连接。
```

```
C:\Documents and Settings\Administrator>
```

Online Base64 encoding and decoding:

http://tools.jb51.net/tools/base64_decode-gb2312.php



Limitations in SMTP

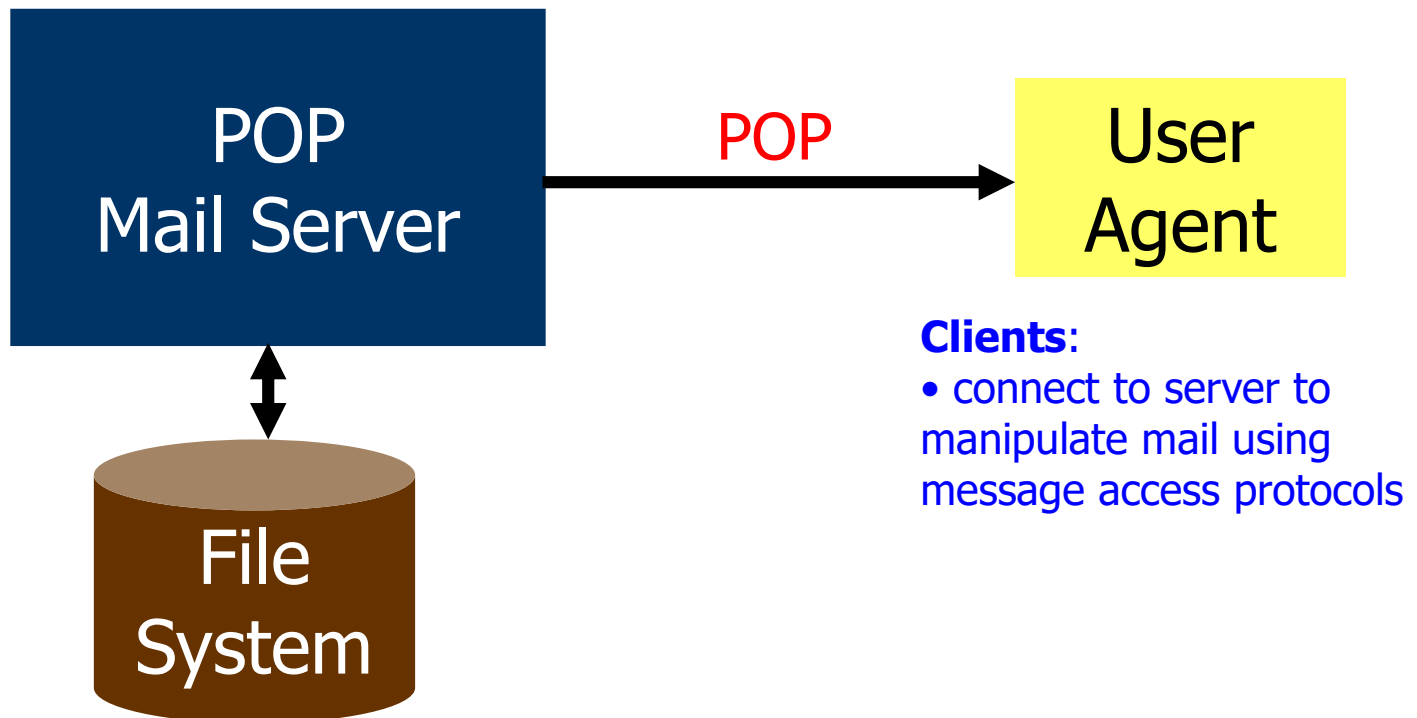
- Only uses ASCII format
 - How to represent other data types?
- No authentication mechanisms
- Messages are sent un-encrypted
- Susceptible to misuse (Spamming, faking sender address)

POP/IMAP – Basic Model

- Used to transfer mail from a mail server to a UA

POP Mail Access Server:

- runs the POP3 service by listening on TCP port 110



Clients:

- connect to server to manipulate mail using message access protocols



POP – Features

- Essentially store and forward. Mail is stored on the server until the client connects and then is **downloaded to the client**. You MAY be able to leave a copy on the server
- **Simple** protocol and widely used.
- Many clients available such as Eudora, foxmail, outlook
- However, very **bad for mobile users** or users that use multiple machines during the day
- Common used version: **POP3** (POP Version 3)



POP3

- Similar to SMTP **command/reply** lockstep protocol
- Used to retrieve mail for a single user
 - requires authentication
- Commands and replies are **ASCII** lines
 - Replies start with “+OK” or “-ERR”
 - Replies may contain multiple lines



POP3 Commands

- **USER** - specify username
- **PASS** - specify password
- **STAT** - get mailbox status
 - number of messages in the mailbox.
- **LIST** - get a list of messages and sizes.
 - One per line, termination line contains '.' only
- **RETR** - retrieve a message
- **DELE** - mark a message for deletion from the mailbox
- **NOOP** - send back positive reply
- **RSET** - reset. All deletion marks are unmarked
- **QUIT** - remove marked messages and close the (TCP) connection



Retrieving Emails Through Telnet (1)

```
C:\Documents and Settings\Administrator>telnet pop3.126.com 110
+OK Welcome to coremail Mail Pop3 Server
(126coms[3adb99eb4207ae5256632eecb8f8b485s])
USER catshiy
+OK core mail
PASS 123456
+OK 1 message(s) [885 byte(s)]
STAT
+OK 1 885
LIST
+OK 1 885
1 885
```

.

Retrieving Emails Through Telnet (2)

RETR 1

+OK 885 octets

Content-Transfer-Encoding: 8bit

MIME-Version: 1.0

Message-ID: <DQ958982777179.06131@mcard.bta.net.cn>

Date: Sun, 17 Oct 2004 22:28:20 +0800 (CST)

From: shp1234@public.bta.net.cn

To: catshiy@126.com

Cc:

Subject:

我十一月中旬以后有空，欢迎你们过来玩。

shp

.

QUIT

+OK core mail

失去了跟主机的连接。

C:\Documents and Settings\Administrator>

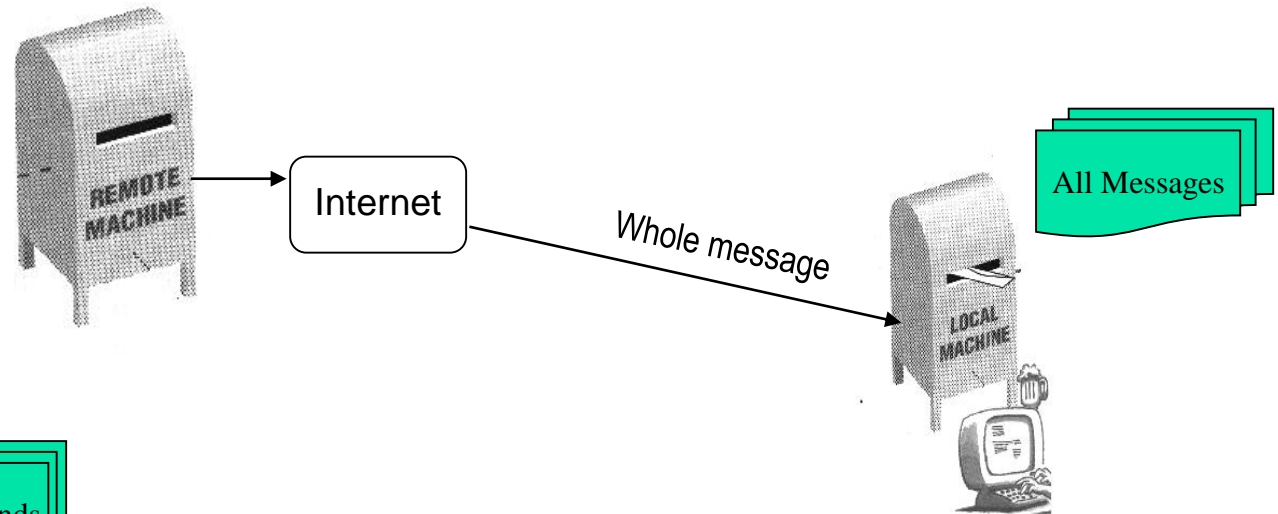


IMAP (Internet Message Access Protocol)

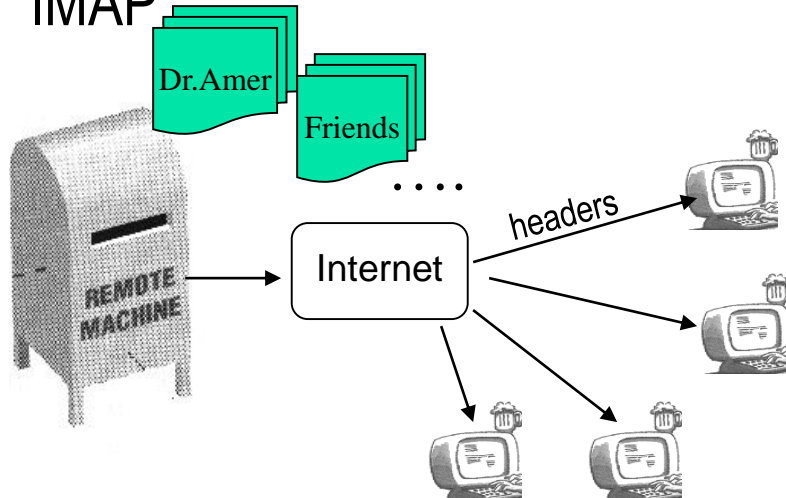
- Features
 - Folders and messages can be stored **either** on the server or on the local computer
 - Since folders can remain on server, it is possible to **access** your same mail store even using a dumb terminal character based client like Pine.
 - Much **better for mobile users than POP** (since mail remains on the server)
 - Can selectively copy (part of) messages from the server to the local client based on many criteria
- QQ/163/gmail supports both IMAP and POP3
- Comparison of POP3 and IMAP
 - https://www.diffen.com/difference/IMAP_vs_POP3

POP vs. IMAP

POP3:




IMAP:





POP vs. IMAP(2)

Feature	POP3	IMAP
Where is protocol defined?	RFC 1939	RFC 2060
Which TCP port is used?	110	143
Where is email stored?	User's PC	Server
Where is email read?	Off-line	On-line
Mail Syncing 	No	Yes
Direction	One-direction	Bi-directional
Good for mobile users?	No	Yes
Partial message downloads?	No	Yes
Speed	Fast	Low

Web-based Mail: HTTP

- Can deliver mail message in web page format



厚德博学 敬业乐群

The screenshot shows the login interface of the BUPT web-based mail system. At the top, there are two tabs: '帐号密码登录' (Account Password Login) and '手机验证码' (Mobile Verification Code). The 'Account Password Login' tab is selected. Below the tabs, there are two input fields: the first for the username (containing 'shiyang') and the second for the password (containing '密码'). To the right of the username field is a placeholder '@bupt.edu.cn'. Below the password field, there is a checkbox labeled '5天内自动登录' (Auto-login within 5 days). At the bottom of the form is a blue button labeled '登录' (Login). Below the button, there are two links: '管理员登录' (Admin Login) on the left and '忘记密码?' (Forgot Password?) on the right.

<https://mail.bupt.edu.cn/>



MIME – Motivation

- Originally, email consisted exclusively of the text messages written in English and expressed in ASCII (**RFC 5322**)
- Nowadays, this approach is no longer appropriate, due to:
 - Messages in languages with accents (e.g. ö, ç, ğ)
 - Messages in non-Latin alphabets
 - Messages in languages without alphabets
 - Messages are not containing text at all -- audio/video
- MIME: Multipurpose Internet Mail Extension



MIME – Features

- Extension for **multipart & multimedia** email
- Additional mail headers define content
 - **type** (text, image, audio, video, application) and subtype within (eg text/html, image/gif)
 - **encoding** (ASCII , quoted printable, base64) to handle arbitrary binary data when email system can only handle normal ASCII chars
- Supports multipart message content type
 - each part has its own type and encoding
- The basic idea of MIME is to use the **ASCII format** (RFC 5322), **but to add structure to the message body** and **define encoding rules for non-ASCII messages**
- By not deviating from RFC 5322, MIME messages can be sent using the existing mail programs and protocols
- Widely used now



MIME – New Headers (1)

Header	Meaning
MIME-Version:	Identifies the MIME version
Content-Description:	Human readable string telling what is in the message
Content-Id:	Unique identifier
Content-Type:	Nature of the message
Content-Transfer-Encoding:	How the body is wrapped for transmission



MIME – New Headers (2)

- *MIME-Version*: simply tells the user agent receiving the messages that it is dealing with MIME messages and which version of MIME it uses - any message not containing a MIME-version is assumed to an English plain-text message
- *Content-Description*: is an ASCII string telling what is in the message
- *Content-Id*: header uniquely identifies the content
- *Content-Transfer-Encoding*: tells how the body is wrapped for transmission - multiple schemes, from the the simplest - ASCII text, through to base64 encoding
- *Content-Type*: tells the type and subtype of the content

MIME – New Headers (3)

- Content types and subtypes

Type	Subtype	Description
Text	Plain	Unformatted text
	Richtext	Text including simple formatting commands
Image	Gif	Still picture in GIF format
	Jpeg	Still picture in JPEG format
Audio	Basic	Audible sound
Video	Mpeg	Movie in MPEG format
Application	octet-stream	An uninterpreted byte sequence
	Postscript	A printable document in PostScript
Message	RFC822	A MIME RFC 822(current RFC5322) message
	Partial	Message has been split for transmission
	External-body	Message itself must be fetched over the net
Multipart	Mixed	Independent parts in the specified order
	Alternative	Same message in different formats
	Parallel	Parts must be viewed simultaneously
	Digest	Each part is a complete RFC 5322 message

MIME – Message Example (1)

Date: Sat, 07 Dec 2002 16:37:32 +0800
From: Adun Gaos
X-Accept-Language: zh-cn
MIME-Version: 1.0
To: adungaos@celldoft.com
Subject: MIME message!
Content-Type: multipart/mixed;
boundary="-----080202030206040206090704"

This is a multi-part message in MIME format.

-----080202030206040206090704

Content-Type: text/html; charset=us-ascii
Content-Transfer-Encoding: 7bit

This is a MIME message. Here is body.

-----080202030206040206090704

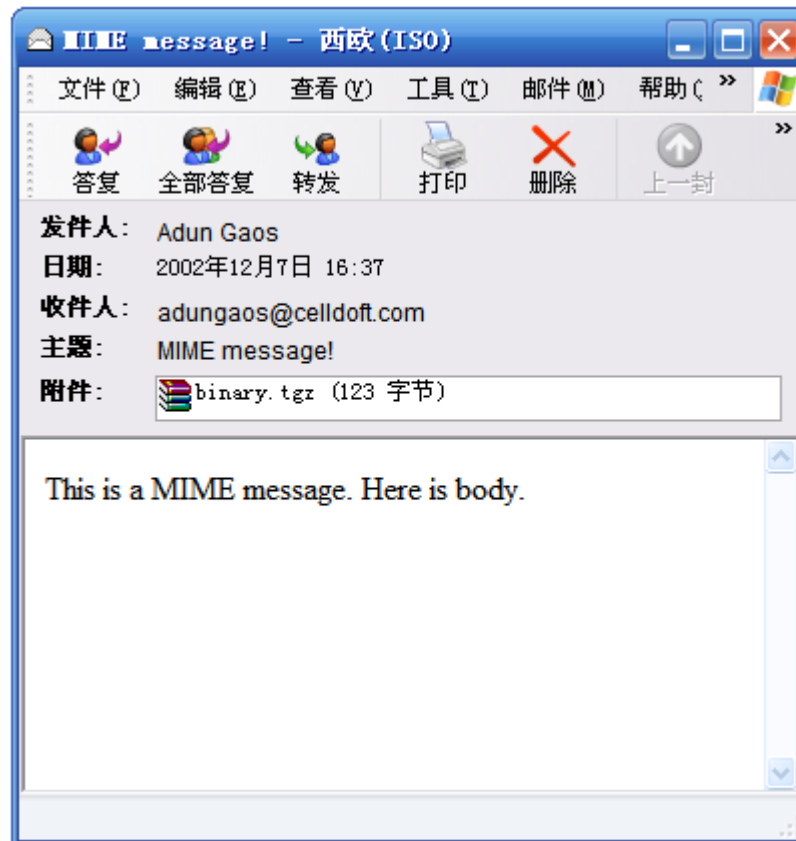
Content-Type: application/x-gtar;
name="binary.tgz"
Content-Transfer-Encoding: base64
Content-Disposition: inline;
filename="binary.tgz"

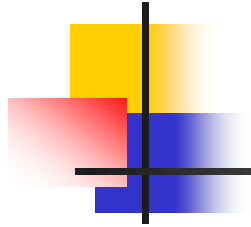
H4sIABmy8T0AA+3OsQ3CMBQEUI/iEb6dBM9jhIRogpSQgu1BQhQUIcPU7zVX3BV3vMx9uadd
RYk4RKSIKG36yLcUbWrjMJQo9bmv41BTjn1vvWzrrS85p37a5nO/rt92v3oAAAAAAAAAAAAAD4
oweF/KCgACgAAA==

-----080202030206040206090704--

MIME – Message Example (2)

- When you save the above as .eml file and open it with outlook, you can see:





Summary



Summary

- **Email**
 - Components of email system
 - Basic functions of email system
 - Email address
- **SMTP**
 - Communication procedure
 - Model
 - Commands and replies
- **POP**
 - Model
 - Commands and replies
 - Communication procedure
- **IMAP**
 - Comparison of POP and IMAP
- **Message formats**
 - RFC 5322
 - MIME
- What are the limitations of SMTP? How are MIME and SSL used to offset the limitations of SMTP?



Useful URLs

- RFCs
 - www.ietf.org
- SMTP, POP & IMAP
 - <http://whatismyipaddress.com/email-basics>
- Base64 encoding and decoding online
 - http://tools.jb51.net/tools/base64_decode-gb2312.php

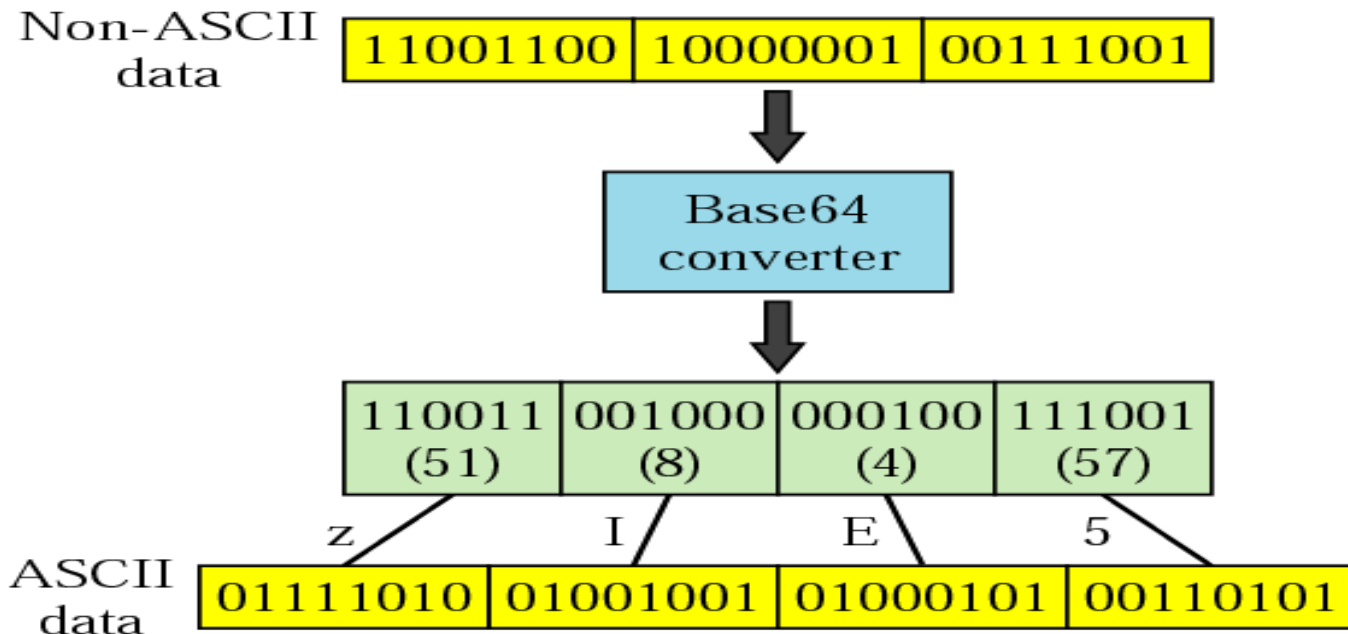


Abbreviations

IMAP	Internet Message Access Protocol
IMF	Internet Message Format
MIME	Multipurpose Internet Mail Extension
MTA	Mail Transfer Agent
POP	Post Office Protocol
SMTP	Simple Mail Transfer Protocol
UA	User Agent

Base64 Encoding

- Divides binary data into 24 bit blocks
- Each block is then divided into 6 bit chunks
- Each 6-bit section is interpreted as one character (8 bits), 25% overhead



Base64 Example

hello

01101000 01100101 01101100 01101100 01101111

011010 000110 010101 101100 011011 000110 111100

(26) (6) (21) (44) (27) (6) (60)
(a) (G) (V) (s) (b) (G) (8) (=)



aGVsbG8=

01100001	01000111	01010110	01110011	01100010	01000111	00111000	00111101
----------	----------	----------	----------	----------	----------	----------	----------

Online encoding and decoding:

http://tools.jb51.net/tools/base64_decode-gb2312.php

Base64 Encoding Table



6-bit value	character encoding	6-bit value	character encoding	6-bit value	character encoding	6-bit value	character 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	/
						(pad)	=