E-MAIL: A RICH INTRODUCTION

Fabrizio d'Amore – DIAG, CIS

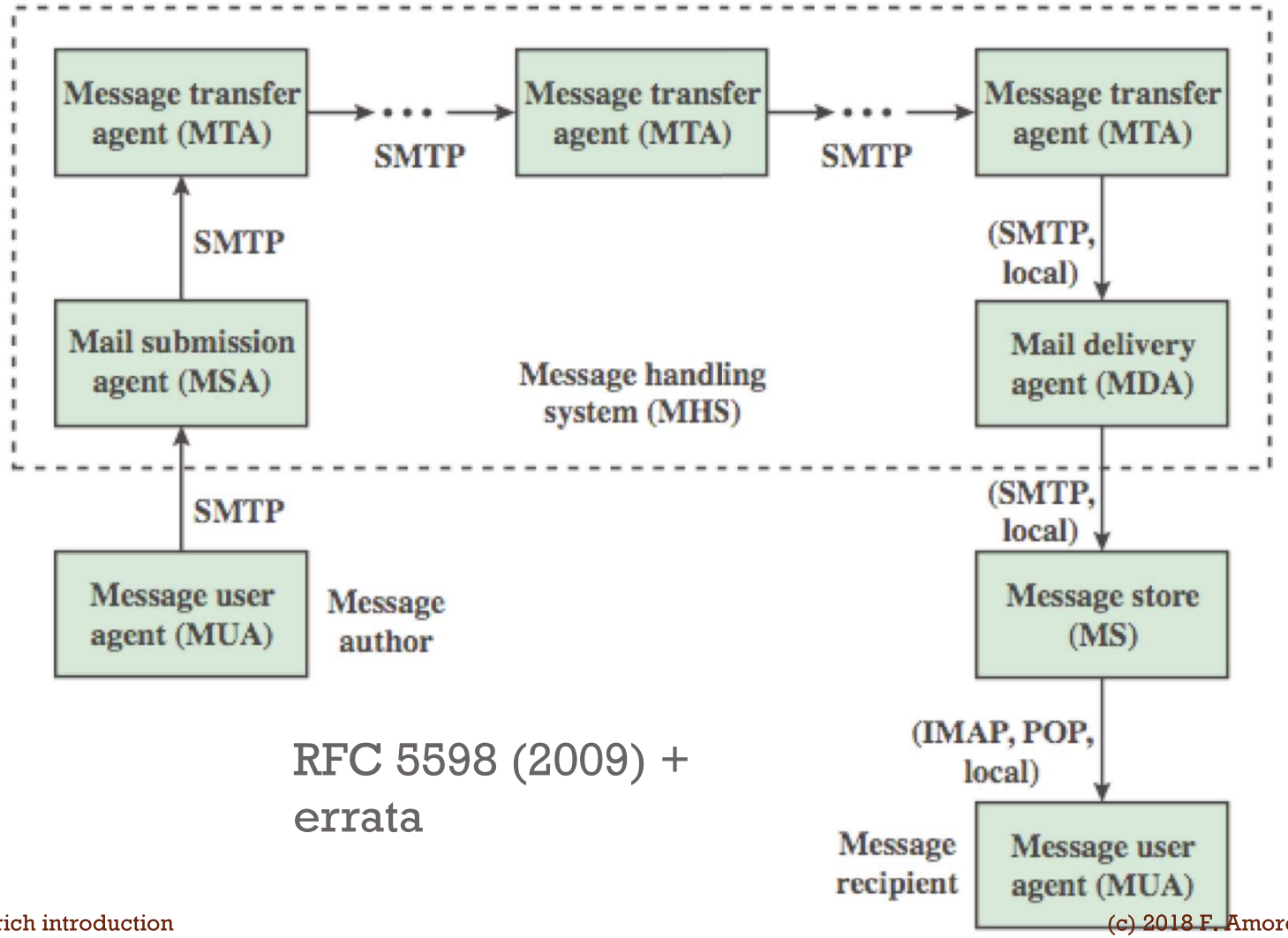
damore@diag.uniroma1.it

THE INTERNET E-MAIL

The Internet e-mail system

- architecture and functioning
- extensions (MIME)
- phishing, spamming, SPF, DKIM
- intro to e-mail forensics (tracking)
- secure e-mailing: PGP

INTERNET E-MAIL ARCHITECTURE



RFC 5598 (2009) + errata

THE E-MAIL SYSTEM

- e-mail is a method of exchanging digital messages from an author to one or more recipients, operating across the Internet/intranet
- modern e-mail systems are based on a **store-and-forward model**: e-mail servers accept, forward, store and deliver messages
- neither the users nor their computers are required to be online simultaneously

THE E-MAIL SYSTEM

- an e-mail message consists of three components
  - the **message envelope**
  - the **message header**, containing control information (originator's email address, one or more recipient addresses, a subject, a message submission date/time stamp etc.)
  - the **message body**
- originally a text-only (7-bit ASCII, or US-ASCII) communications medium
- extended to carry multi-media content attachments, a process standardized in RFC 2045 through 2049 (Multipurpose Internet Mail Extensions - MIME)

E-MAIL EXCHANGE

- e-mail transmission across IP networks is carried by the Simple Mail Transfer Protocol (SMTP, RFC 821, 1982)
  - last update: RFC 5321 (2008). Includes the extended SMTP (ESMTP) additions
- SMTP communicates delivery parameters using a message envelope separate from the message (header and body) itself
- an Internet e-mail address is a string of the form localpart@exampledomain
  - the part before the @ sign is the local part of the address
  - the part after the @ sign is a domain name (PS+1) or a fully qualified domain name
- user-level client mail applications only use SMTP for sending messages to a mail server for relaying
- to access their mail box accounts, client applications usually use either the Post Office Protocol (POP) or the Internet Message Access Protocol (IMAP) or a proprietary system (such as Microsoft Exchange or Lotus Notes/Domino)

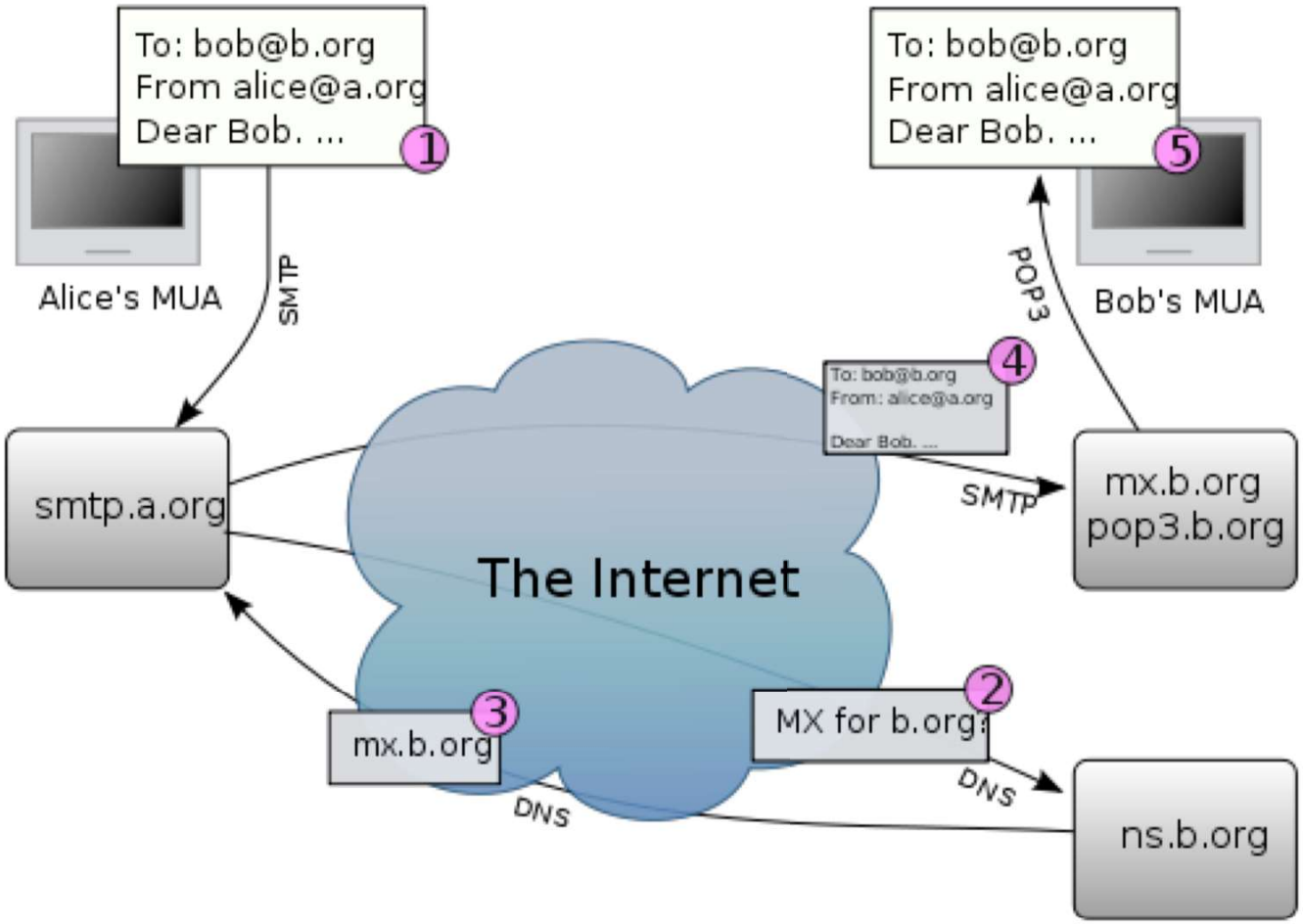
MUA, MSA, MTA

- MUA: computer program used to access and manage a user's e-mail
- MSA: computer program or software agent that receives e-mail messages from a MUA and cooperates with a mail transfer agent (MTA) for delivery of the mail
  - it uses a variant of the Simple Mail Transfer Protocol (SMTP), as specified in RFC 6409
  - its functions are a subset of those of MTA, since it deals with senders (as for input) and MTA (as for output)
  - it makes it easer for a MTA to deny relaying
- MTA: software that transfers e-mail messages from one computer to another using a client–server application architecture
  - MTAs implements both the client and server portions of the Simple Mail Transfer Protocol

MDA, MRA

- MDA: computer software component that is responsible for the delivery of e-mail messages to a local recipient's mailbox
  - within the Internet mail architecture, local message delivery is achieved through a process of handling messages from the message transfer agent, and storing mail into the recipient's environment (typically a mailbox)
- MRA: computer application that retrieves or fetches e-mail from a remote mail server and works with an mail delivery agent to deliver mail to a local or remote email mailbox
  - MRAs may be external applications or be built into a bigger application like an MUA
  - The concept of MRA is not a standardized in e-mail architecture. Although they operate like mail transfer agents, MRAs are technically clients when they retrieve and submit messages

OPERATION OVERVIEW



OPERATION OVERVIEW

1. Alice composes a message using her mail user agent (MUA); she enters the e-mail address of her correspondent, and hits the "send" button
2. the MUA formats the message in email format and uses the Submission Protocol (variant of SMTP, see RFC 6409) to send the message to the local message submission agent (MSA), in this case smtp.a.org, run by Alice's ISP
3. the MSA looks at the destination address provided in the SMTP protocol, in this case bob@b.org and resolves a domain name to determine the fully qualified domain name of the mail exchange server in the Domain Name System (DNS)
4. the DNS server for the b.org domain, ns.b.org, responds with any MX records listing the mail exchange servers for that domain, in this case mx.b.org, a message transfer agent (MTA) server run by Bob's ISP

OPERATION OVERVIEW

5. smtp.a.org sends the message to mx.b.org using SMTP
   - this server may need to forward the message to other MTAs before the message reaches the final message delivery agent (MDA), which delivers it to the mailbox of Bob
6. Bob presses the "get mail" button in his MUA, which picks up the message using either the Post Office Protocol (POP3) or the Internet Message Access Protocol (IMAP4)

VARIANTS AND OTHER NOTES

A few alternative possibilities and complications can occurr

- Alice or Bob may use a client connected to a corporate email system, such as IBM Lotus Notes or Microsoft Exchange and the entire transaction may be completely local
- Alice may not have a MUA on her computer but instead may connect to a webmail service
- Alice's computer may run its own MTA, so avoiding the transfer at step 1
- Bob may pick up his email in many ways, for example logging into mx.b.org and reading it directly, or by using a webmail service
- Domains usually have several mail exchange servers so that they can continue to accept mail when the main mail exchange server is not available
- Email messages are not secure if email encryption is not used correctly
- Number of open e-mail relays is decreasing (to prevent spam)

MESSAGE FORMAT

- Message format is defined by RFC 5322
  - support to MIME (RFC 2045 through RFC 2049), collectively called Multipurpose Internet Mail Extensions
- Internet e-mail messages consist of two major sections:
  - **Header** — Structured into fields such as From, To, CC, Subject, Date, and other information about the email.
  - **Body** — The basic content, as unstructured text; sometimes containing a signature block at the end. This is exactly the same as the body of a regular letter.
- The header is separated from the body by a blank line

HEADER

- Each message has exactly one header, which is structured into fields. Each field has a name and a value. RFC 5322 specifies the precise syntax
- Informally, each line of text in the header that begins with a printable character begins a separate field and its name starts in the first character of the line and ends before the separator character ":"
- The separator is then followed by the field value (the "body" of the field). The value is continued onto subsequent lines if those lines have a space or tab as their first character. Field names and values are restricted to 7-bit ASCII characters. Non-ASCII values may be represented using MIME encoded words
- Email header fields can be multi-line, and each line must be at most 76 characters long. Header fields can only contain US-ASCII characters; for encoding characters in other sets, a syntax specified in RFC 2047 can be used

MANDATORY HEADER FIELDS

- **From:**
  The email address, and optionally the name of the author(s)

- **Date:**
  The local time and date when the message was written

OTHER (SUGGESTED) HEADER FIELDS

- **Message-ID**
  Also an automatically generated field; used to prevent multiple delivery and for reference in In-Reply-To:
- **In-Reply-To**
  Message-ID of the message that this is a reply to. Used to link related messages together. This field only applies for reply messages
- RFC 3864 describes registration procedures for message header fields at the IANA; it provides for permanent and provisional message header field names

COMMON HEADER FIELDS

To:
The email address(es), and optionally name(s) of the message's recipient(s). Indicates primary recipients (multiple allowed)

Subject:
A brief summary of the topic of the message. Certain abbreviations are commonly used in the subject, including "RE:" and "FW:"

Bcc:
Blind Carbon Copy; addresses added to the SMTP delivery list but not (usually) listed in the message data, remaining invisible to other recipients.

Cc:
Carbon copy; Many email clients will mark email in your inbox differently depending on whether you are in the To: or Cc: list.

Content-Type:
Information about how the message is to be displayed, usually a MIME type.

COMMON HEADER FIELDS

- **Precedence:**
  commonly with values "bulk", "junk", or "list"; used to indicate that automated "vacation" or "out of office" responses should not be returned for this mail
  - With modern high-bandwidth networks delivery priority is less of an issue than it once was.
- **References:**
  Message-ID of the message that this is a reply to, and the message-id of the message the previous reply was a reply to, etc.
- **Reply-To:**
  Address that should be used to reply to the message
- **Sender:**
  Address of the actual sender acting on behalf of the author listed in the From: field (secretary, list manager, etc.)
- **Archived-At:**
  A direct link to the archived form of an individual email message

- Note that the To: field is not necessarily related to the addresses to which the message is delivered. The actual delivery list is supplied separately to SMTP, which may or may not originally have been extracted from the header content. In the same way, the "From:" field does not have to be the real sender of the email message.

TRACE INFORMATION OF A MESSAGE

SMTP defines the trace information of a message, which is also saved in the header using the following two fields:

- **Received:**
  when an SMTP server accepts a message it inserts this trace record at the top of the header (last to first)
- **Return-Path:**
  when the delivery SMTP server makes the final delivery of a message, it inserts this field at the top of the header

Other header fields that are added on top of the header by the receiving server may be called trace fields, in a broader sense

- **Authentication-Results:**
  when a server carries out authentication checks, it can save the results in this field for consumption by downstream agents
- **Received-SPF:**
  stores the results of SPF checks
- **Auto-Submitted:**
  is used to mark automatically generated messages
- **VBR-Info:**
  claims VBR whitelisting

SAMPLE SMTP INTERACTION

as of RCF 821

```
S: 220 hamburger.edu
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
```

OTHER SMTP COMMANDS

as of RCF 821

- **RSET** current mail transaction is to be aborted
- **SEND**, **SAML**, **SOML** announce message to terminal and/or mailbox
- **VRFY** asks to confirm that the argument identifies a user
- **EXPN** asks to confirm that the argument identifies a mailing list (to be expanded)
- **HELP** for general or specific help
- **NOOP** no operation
- **TURN** reverses the SMTP communication, by exchanging the role of the parties

some of the above commands may be not implemented

ESMTP

- RFC 821 (1982) was obsoleted by RFC 5321 (2008), where ESMTP, an extended version of SMPT, was introduced
- software agents should stick to ESMTP but for backward compatibility reason a client connecting by SMTP will be also served
- the greeting command for ESMTP is **EHLO**, which gets a (possibly multiline) response listing the supported extended commands

```
S: 220 foo.com Simple Mail Transfer Service Ready
C: EHLO bar.com
S: 250-foo.com greets bar.com
S: 250-8BITMIME
S: 250-SIZE
S: 250-DSN
S: 250-VRFY
S: 250 HELP
```

OTHER ESMTP COMMANDS

from Wikipedia

- 8BITMIME — 8 bit data transmission, RFC 6152
- ATRN — Authenticated TURN for On-Demand Mail Relay, RFC 2645
- AUTH — Authenticated SMTP, RFC 4954
- CHUNKING — Chunking, RFC 3030
- DSN — Delivery status notification, RFC 3461 (See Variable envelope return path)
- ETRN — Extended version of remote message queue starting command TURN, RFC 1985
- HELP — Supply helpful information, RFC 821
- PIPELINING — Command pipelining, RFC 2920
- SIZE — Message size declaration, RFC 1870
- STARTTLS — Transport layer security, RFC 3207 (2002)
- SMTPUTF8 — Allow UTF-8 encoding in mailbox names and header fields, RFC 6531
- UTF8SMTP — Allow UTF-8 encoding in mailbox names and header fields, RFC 5336 (deprecated)

SOFTWARE FOR ESMTP

- according to a survey (from Security Space), sendmail, Microsoft Exchange Server, Postfix, and Exim together control over 90% of market share for SMTP service in 2014
  - http://www.securityspace.com/s_survey/data/man.201403/mxsurvey.html

MIME (MULTIPURPOSE INTERNET MAIL EXTENSIONS)

- Internet standard that extends the format of email to support:
  - Text in character sets other than ASCII
  - Non-text attachments
  - Message bodies with multiple parts
  - Header information in non-ASCII character sets
- MIME's use has grown beyond describing the content of email to describe content type in general (web, storage)
- Virtually all human-written Internet email and a fairly large proportion of automated email is transmitted via SMTP in MIME format
- MIME is specified in six linked RFC memoranda

MIME

Important RFCs

- RFC-822 Standard for the format for ARPA Internet text messages
- RFC-2045 MIME Part 1: Format of Internet Message Bodies
- RFC-2046 MIME Part 2: Media Types
- RFC-2047 MIME Part 3: Message Header Extensions
- RFC-2048 MIME Part 4: Registration Procedure
- RFC-2049 MIME Part 5: Conformance Criteria

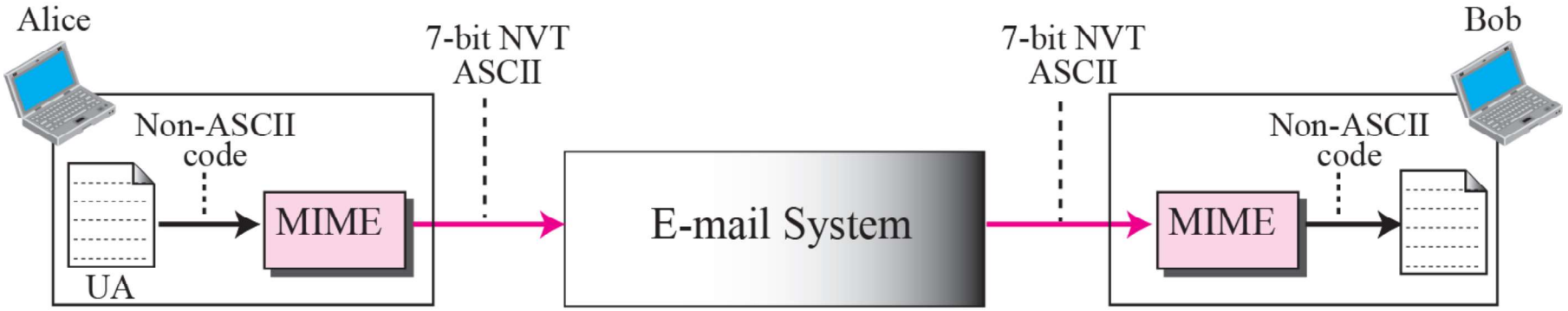
MIME – WHAT IS IT?

- MIME refers to an official Internet standard that specifies how messages must be formatted so that they can be exchanged between different email systems.
- MIME permits the inclusion of virtually any type of file or document in an email message.
- Specifically, MIME messages can contain
  - text
  - images
  - audio
  - video
  - application-specific data
    - spreadsheets
    - word processing documents

MIME FEATURES

- Support of character sets other than ASCII
- Content type labeling system
- Support of non-text content in e-mail messages
- Support for compound documents

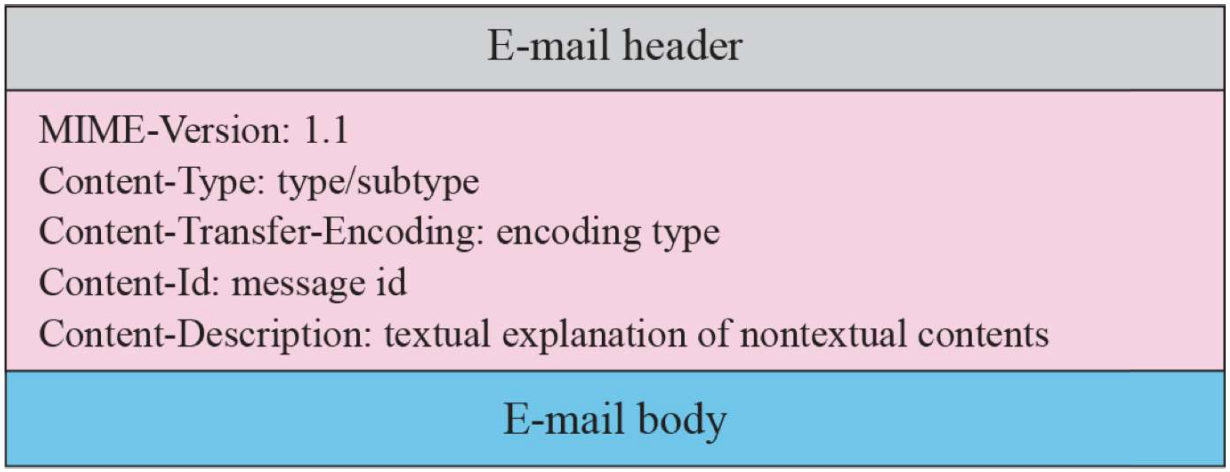
MIME SCHEME



NVT = *network virtual terminal*

MIME HEADERS

MIME headers



NON-ASCII CHARACTER SET SUPPORT

- Message header
  - content-type field
    - put in the header by the client program creating the e-mail for use by the client program used to display the received message
    - charset= optional parameter
      - if absent ASCII is assumed
- Content-Type: text/plain; charset="ISO-8859-1"
  - ISO-8859-1 extends the basic character set of ASCII to include many of the accented characters used in languages such as Spanish, French, German and Italian.
  - US-ASCII is the standard character set used in the US

CONTENT LABELING

- a set of registered MIME Types that map to specific file types
  - MIME Types consist of :
    - a primary type
    - a sub type separated by a / (as text/html)
- Common Mime Types:

FileExtension	MIME Type	Description
.txt	text/plain	Plain text
.htm	text/html	Styled text in HTML format
.jpg	image/jpeg	Picture in JPEG format
.gif	image/gif	Picture in GIF format
.wav	audio/x-wave	Sound in WAVE format
.mp3	audio/mpeg	Music in MP3 format
.mpg	video/mpeg	Video in MPEG format
.zip format	application/zip	Compressed file in PK-ZIP

MIME TYPES/SUBTYPES

Type	*Subtype*	*Description*
Text	Plain	Unformatted
	HTML	HTML format (see Appendix E)
Multipart	Mixed	Body contains ordered parts of different data types
	Parallel	Same as above, but no order
	Digest	Similar to Mixed, but the default is message/RFC822
	Alternative	Parts are different versions of the same message
Message	RFC822	Body is an encapsulated message
	Partial	Body is a fragment of a bigger message
	External-Body	Body is a reference to another message
Image	JPEG	Image is in JPEG format
	GIF	Image is in GIF format
Video	MPEG	Video is in MPEG format
Audio	Basic	Single channel encoding of voice at 8 KHz
Application	PostScript	Adobe PostScript
	Octet-stream	General binary data (eight-bit bytes)

CONTENT-TRANSFER-ENCODING

Type	*Description*
7bit	NVT ASCII characters and short lines
8bit	Non-ASCII characters and short lines
Binary	Non-ASCII characters with unlimited-length lines
Base64	6-bit blocks of data are encoded into 8-bit ASCII characters
Quoted-printable	Non-ASCII characters are encoded as an equal sign plus an ASCII code

NON-TEXT CONTENT

To be sent through the e-mail system non-textual content is converted (encoded) to ASCII for transmission and decoded back to its original format for display upon receipt

- originally done via uuencode
- MIME uses base 64 encoding (RFC 2045)
  - binary to text encoding scheme
  - targets A-Z, a-z, 0-9, +,/
- scheme:
  - take three bytes of data, put into a 24 bit buffer
  - extract 4 six-bits values
  - use each value as an index into: ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/
  - this yields 4 ASCII characters
  - use zero, one or two = symbols for padding (at the end)

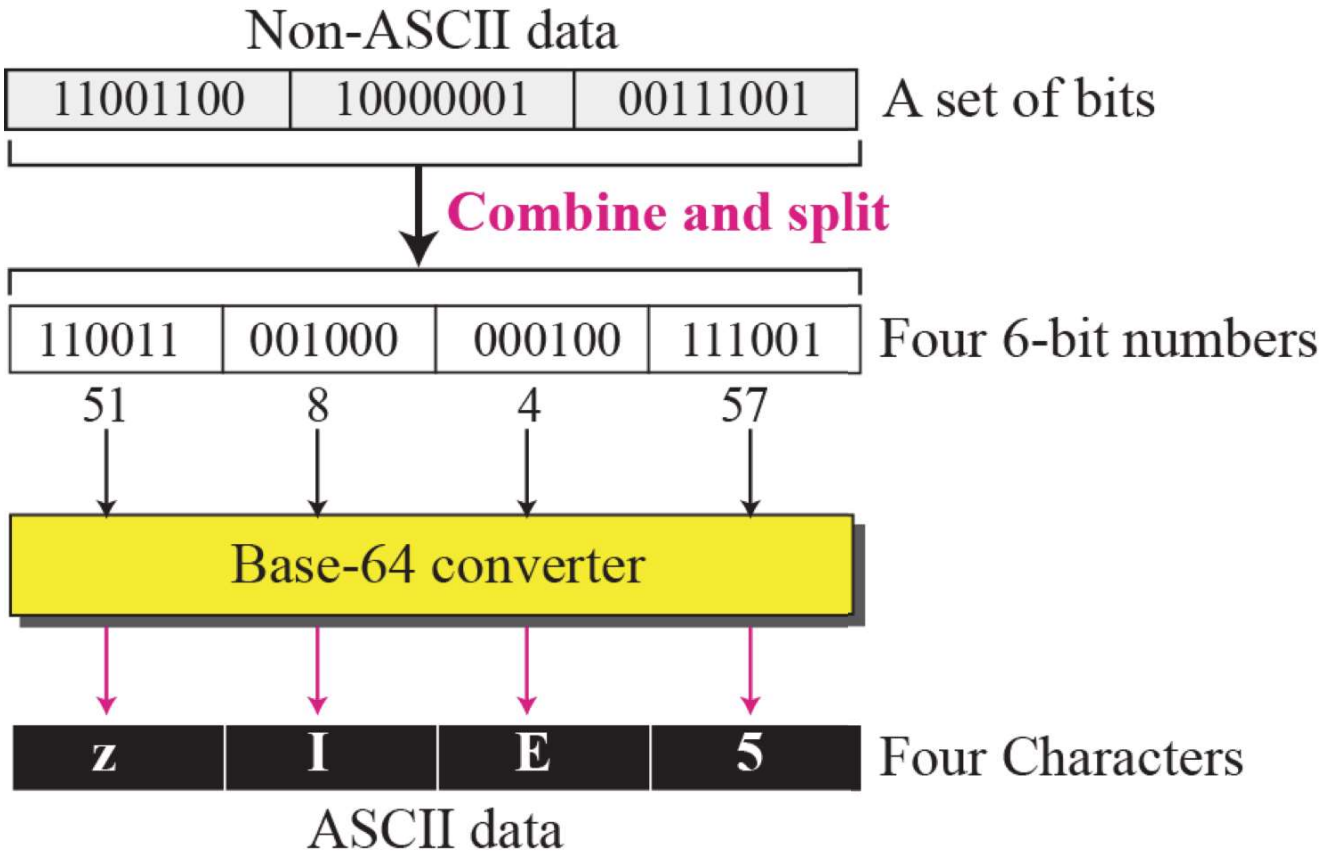
BASE-64 ENCODING EXAMPLE

Man is distinguished, not only by his reason, but by this singular passion from other animals, which is a lust of the mind, that by a perseverance of delight in the continued and indefatigable generation of knowledge, exceeds the short vehemence of any carnal pleasure.

base64 encoded:

TWFuIGlzIGRpc3Rpbmd1aXNoZWQsIG5vdCBvbmx5IGJ5IGhpcyByZWFzb24sIGJ1dCBieSB0
aGlzIHNpbmd1bGFyIHBhc3Npb24gZnJvbSBvdGhlciBhbmltYWxzLCB3aGljaCBpcyBhIGx1
c3Qgb2YgdGhlIG1pbmQsIHRoYXQgYnkgYSBwZXJzZXZlcmFuY2Ugb2YgZGVsaWdodCBpbiB0
aGUgY29udGludWVkIGFuZCBpbmRlZmF0aWdhYmxlIGdlbmVyYXRpb24gb2Yga25vd2xlZGdl
LCBleGNlZWRzIHRoZSBzaG9ydCB2ZWhlbWVuY2Ugb2YgYW55IGNhcm5hbCBwbGVhc3VyZS
4=

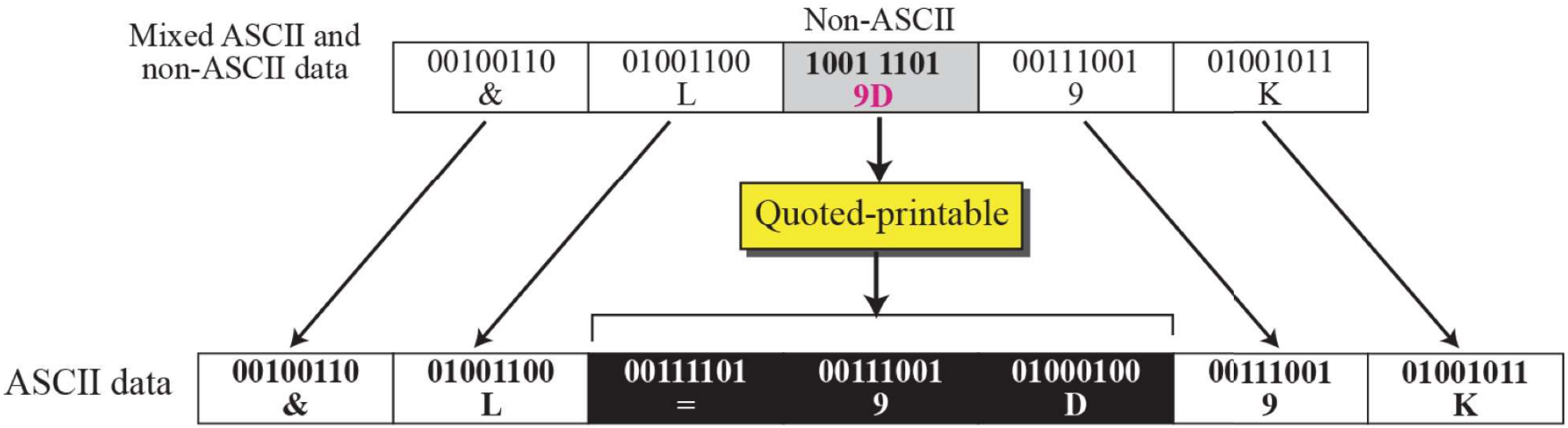
BASE-64 ENCODING SCHEMA



BASE-64 CONVERTING TABLE

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

QUOTED-PRINTABLE ENCODING



- any 8-bit byte value may be encoded with 3 characters: an '=' followed by two hexadecimal digits (0–9 or A–F) representing the byte's numeric value
- non 8-bit byte values are ASCII chars from 33 to 126 (excluded 61, the '=' sign)
- special cases for SPACE and TAB

MULTIPART SUBTYPES

- **Mixed**. For sending files with different "Content-Type" headers.
- **Digest**. To send multiple text messages.
- **Message**. Contains any MIME email message, including any headers
- **Alternative**. Each part is an "alternative" version of the same (or similar) content (e.g., text + HTML)

- more subtypes…

MULTIPART MESSAGES

from Wikipedia

- A MIME multipart message contains a boundary in the "Content-Type: " header; this boundary, which must not occur in any of the parts, is placed between the parts

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=frontier

This is a message with multiple parts in MIME format.
--frontier
Content-Type: text/plain

This is the body of the message.
--frontier
Content-Type: application/octet-stream
Content-Transfer-Encoding: base64

PGh0bWw+CiAgPGhlYWQ+CiAgPC9oZWFkPgogIDxib2R5PgogICAgPHA+VGhpcyBpcyB0aGUg
Ym9keSBvZiB0aGUgbWVzc2FnZS48L3A+CiAgPC9ib2R5Pgo8L2h0bWw+Cg==
--frontier--
```

- Each part consists of its own content header (zero or more Content-* header fields) and a body. Multipart content can be nested.

PLAIN TEXT AND HTML

- modern graphic email clients allow use of HTML for the message body
  - HTML email messages often include an automatically generated plain text copy as well
- HTML messages should have an additional header: "Content-type: text/html". Most email programs insert this header automatically
- advantages of HTML include the ability to include in-line links and images, set apart previous messages in block quotes, wrap naturally on any display, use emphasis such as underlines and italics, and change font styles
- disadvantages include the increased size of the email, privacy concerns about web bugs, abuse of HTML email as a vector for phishing attacks and the spread of malicious software
- some mailing lists recommend that all posts be made in plain-text, with 72 or 80 characters per line for all the above reasons, but also because they have a significant number of readers using text-based email clients
- some Microsoft email clients allow rich formatting using RTF (portability issues)

UNWANTED E-MAIL MESSAGES

- SPAM = unwanted ads (?)
  - both normal and low quality merchandize (drugs, pharmacy, dating, online sex, pirated software/multimedia etc.)
- frauds/malware
  - "write here your username/password"
  - "write here your credit card number"
  - "help me to retrieve $ 20 000 000 …"
  - "you haven't claimed your € 500 prize"
  - loans and funds at lowest rates
  - "I'm so lonely and looking for love…"
  - "you won the lottery"
  - "the message you have sent is undeliverable"
  - "invoice to be paid: click here"
- e-mail chain letters
  - exponential growth
- all of above, joint to low-quality automatic language translation

we'll use the generic terms spam or junk for denoting unwanted or undesirable e-mail messages

WHAT DO SPAMMERS WANT?

- sell products/services (aggressive marketing)
- sell low-qualities/fake/expired goods/medicines (low prices)
- distribute/spread malware (viruses, worms, Trojan horses, backdoors, rootkits etc.) and grayware (adware, spyware, dialers etc.)
  - computer can be enrolled/controlled for participation in (future) attacks
  - Internet activity (browsing, instant messaging and other social activity) can be monitored, users can be profiled
  - audio/video sessions can be recorded
  - collect (any) data on you and on your contacts (databases are built to the purpose of digital identity thefts)
- phishing
  - username/password stealing, credit card data capture, frauds etc.
  - often based on malicious links
- validate e-mail addresses
  - can be re-sold at a higher price
  - based on HTML images and links
- *this list is non-exhaustive*

BASIC E-MAIL NONALOGUE

- disable HTML messages or, at least, disable download of remote images
  - prevent the sender to validate our e-mail address
- don't click links (specially if tiny or IP-based URLs)
  - could redirect to bad web sites containing malware/spyware
- don't open unknown/unexpected attachments
  - they may contain malware/spyware
  - executables (.exe, .app, .bat etc.), documents(.doc, .pdf etc.) and others (.src, …)
- activate local anti-spam filter
- don't participate with chain letters
  - google their contents!
- protect and respect privacy of other recipients
  - be careful in e-mail forwarding (don't uselessly disclose e-mail addresses)
- even if non-Windows user, activate anti-virus for protecting your (Windows) recipients
- don't provide your personal/sensitive data
  - identity thefts!
- don't click "delete me"
  - may validate your email address
  - OK with known senders

HTML IMAGES

- could be used for e-mail address validation
- prepare in remote http server a file image to be referred in HTML formatted message (e.g., "small.gif")
- image may be a small dot having same color as background
- for each e-mail address make symbolic link (alias) to image file and define a corresponding table

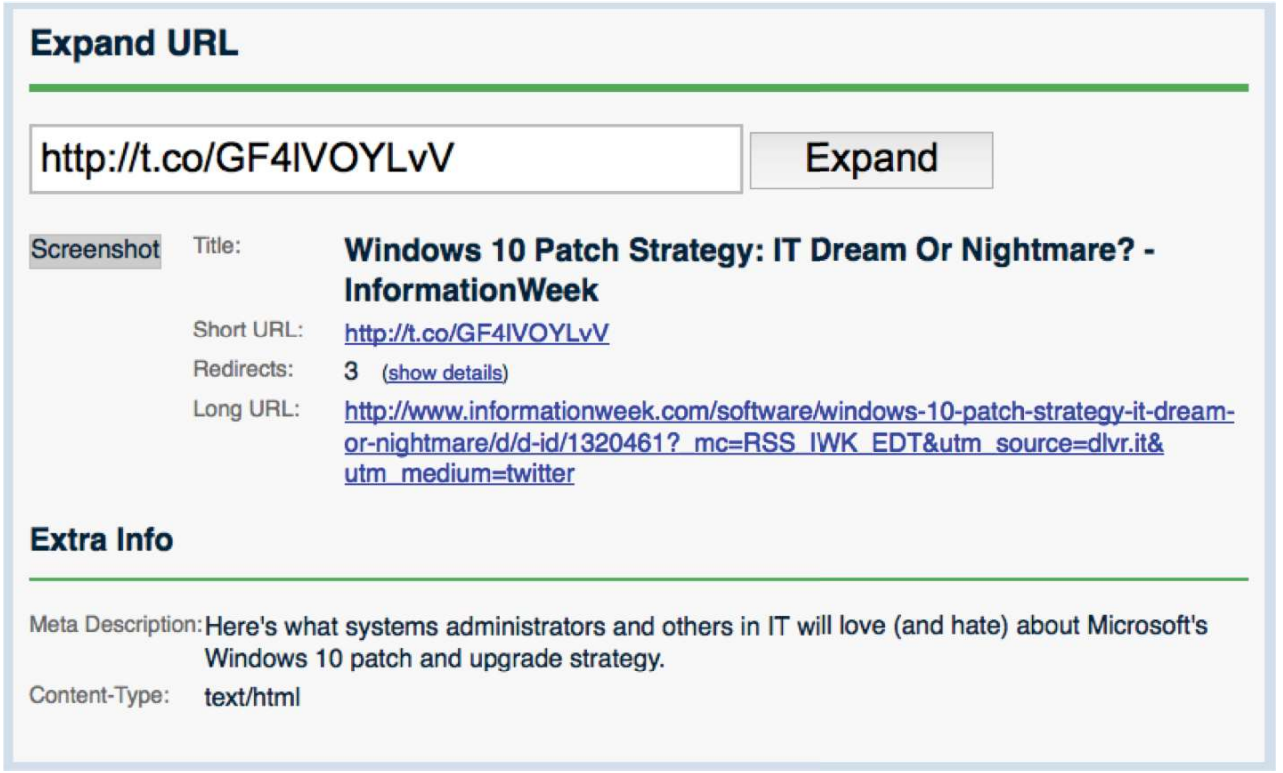
address	*alias*
a@b.c	s1.gif
d@e.f	s2.gif
g@h.i	s3.gif

- print/merge HTML messages using the above table (with each recipient use appropriate filename)
- send messages, then grep logs of remote http server

EXPANSION OF TINY URLS

how to?

- click & see
  - risky!
- use analyzing tools
  - where can we find them?
- ad hoc services on the Web
  - e.g.: http://longurl.org/
  - good results?

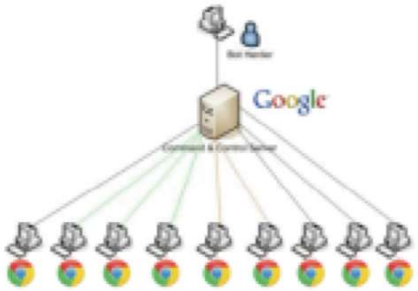


URL CHECKING

twitted (twitpic) on March 18, 2013

Google Chrome

- **Sends the name of the file you're downloading** to Google for whitelist checking; **stores your IP address** associated with the file for a few weeks
- **Every URL** you even begin to type in the address bar is **sent to Google**, in whole or in fragments, for auto-completion purposes
- **Connects to Google every 30 minutes** to download a list of malicious URLs, so the fact that you even have Chrome open is transmitted to Google
- Asks you to login to your Google account, so your **browsing tabs, history, etc. is stored on Google servers**
- **Connects to websites** in the background before you are even finished typing them in, **without your explicit instruction**



Summary: there is nothing, *nothing*, you can do in Chrome that isn't transmitted to Google through some channel.

Welcome to the Botnet.