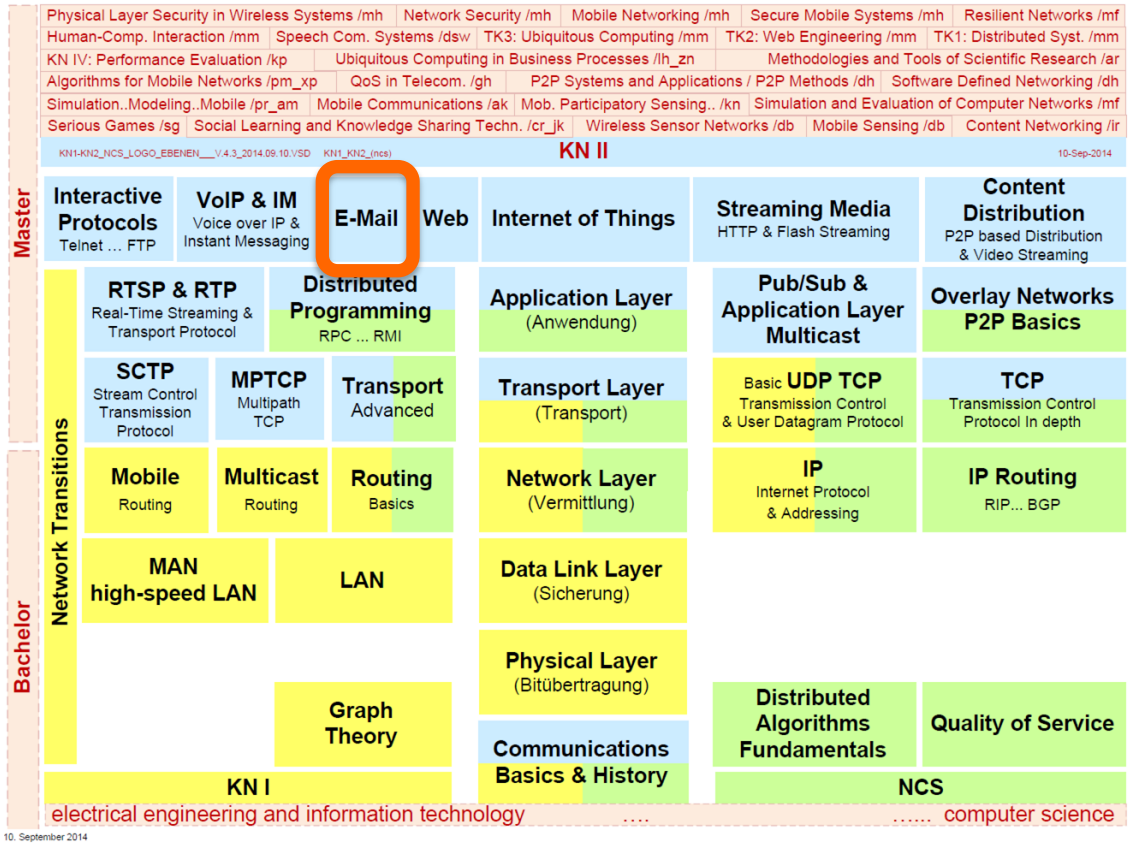
Communication Networks II

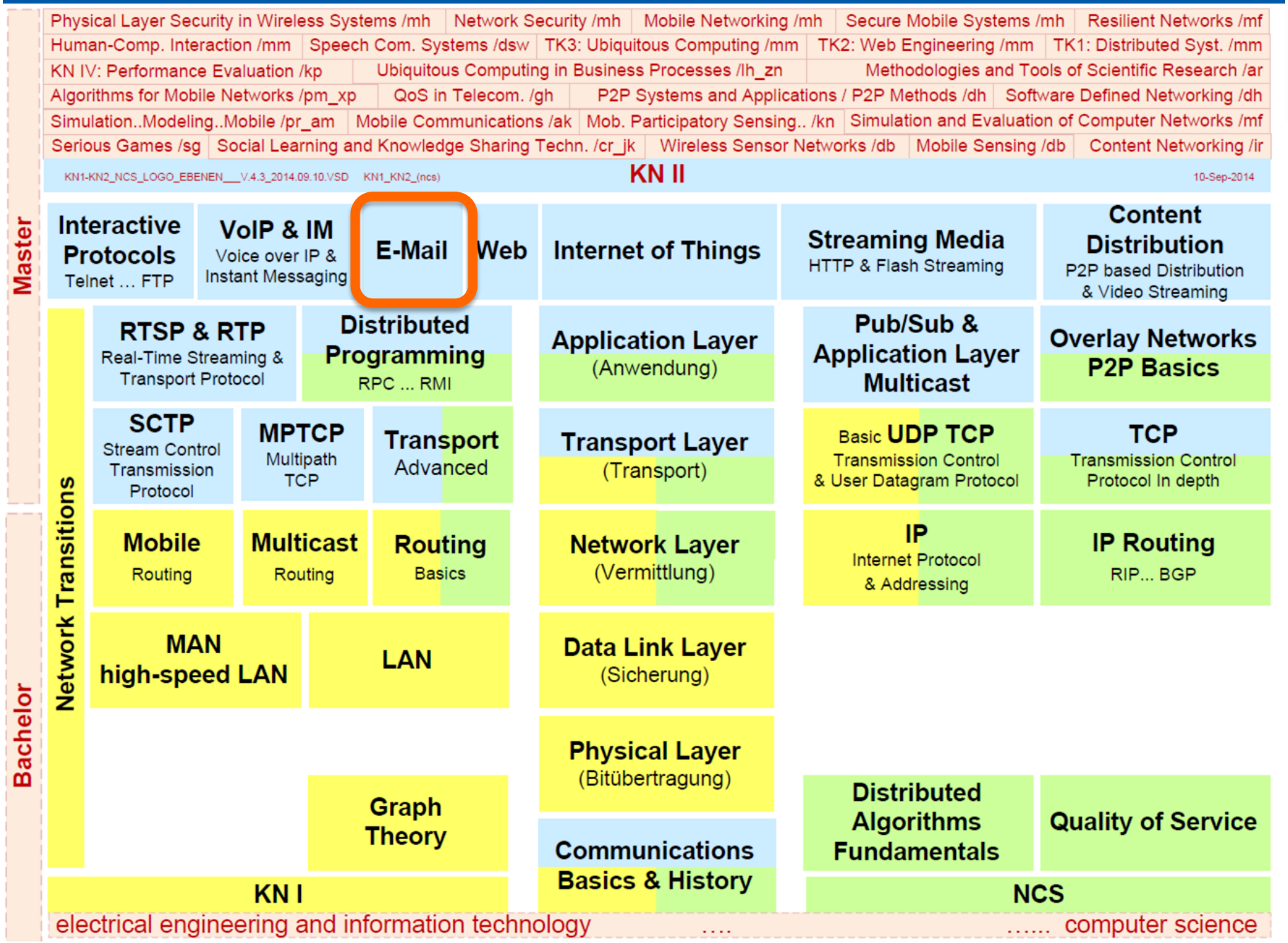
Electronic Mail



Prof. Dr.-Ing. **Ralf Steinmetz**
KOM - Multimedia Communications Lab

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT



10. September 2014

TECHNISCHE UNIVERSITÄT DARMSTADT

Some Numbers:

- 107 trillion – The number of emails sent on the Internet in 2010.
- 294 billion – Average number of email messages per day.
- 1.88 billion – The number of email users worldwide.
- 480 million – New email users since the year before.
- 89.1% – The share of emails that were spam.
- 262 billion – The number of spam emails per day
  - assuming 89% are spam
- 2.9 billion – The number of email accounts worldwide
- 25% – Share of email accounts that are corporate.

Source: http://royal.pingdom.com/2011/01/12/internet-2010-in-numbers/

TECHNISCHE UNIVERSITÄT DARMSTADT

First email systems in 1970ties simply consisted of file transfer protocols

- Convention: First line of message contains recipient address

Some Limitations

- Sending message to group was inconvenient
- Message had no internal structure
  - Separation of messages was difficult
- Sender never knew whether the email was received or not
- Automatic forwarding of messages was not possible
- Poor user interface
  - First edit message
  - Then transfer message with separate program
- Multipart messages were not supported

History

1972

- first e-mail sent between 2 systems
- Invented by Ray Tomlinson
  - Ray Tomlinson sent it to himself
  - RFC 821 (transmission protocol) and RFC 822 (message format) in 1982

e-mail was THE application of the Internet

- until the web was introduced
- and, more recently
  - peer-to-peer communication is in place

Users

- < 1990: universities, research
- > 1990: companies, usually first within the engineering departments
- Since many years: everybody

Composition

- Creating new messages or answers
- Using the Internet email format, including all header fields
- Consisting of text and additional data (documents, excel sheets, pictures…)

Transfer

- Moving message from originator to recipient

Report

- Status of sent messages
- Received? Got lost? Was rejected by recipient?

Display

- Showing received emails to the user

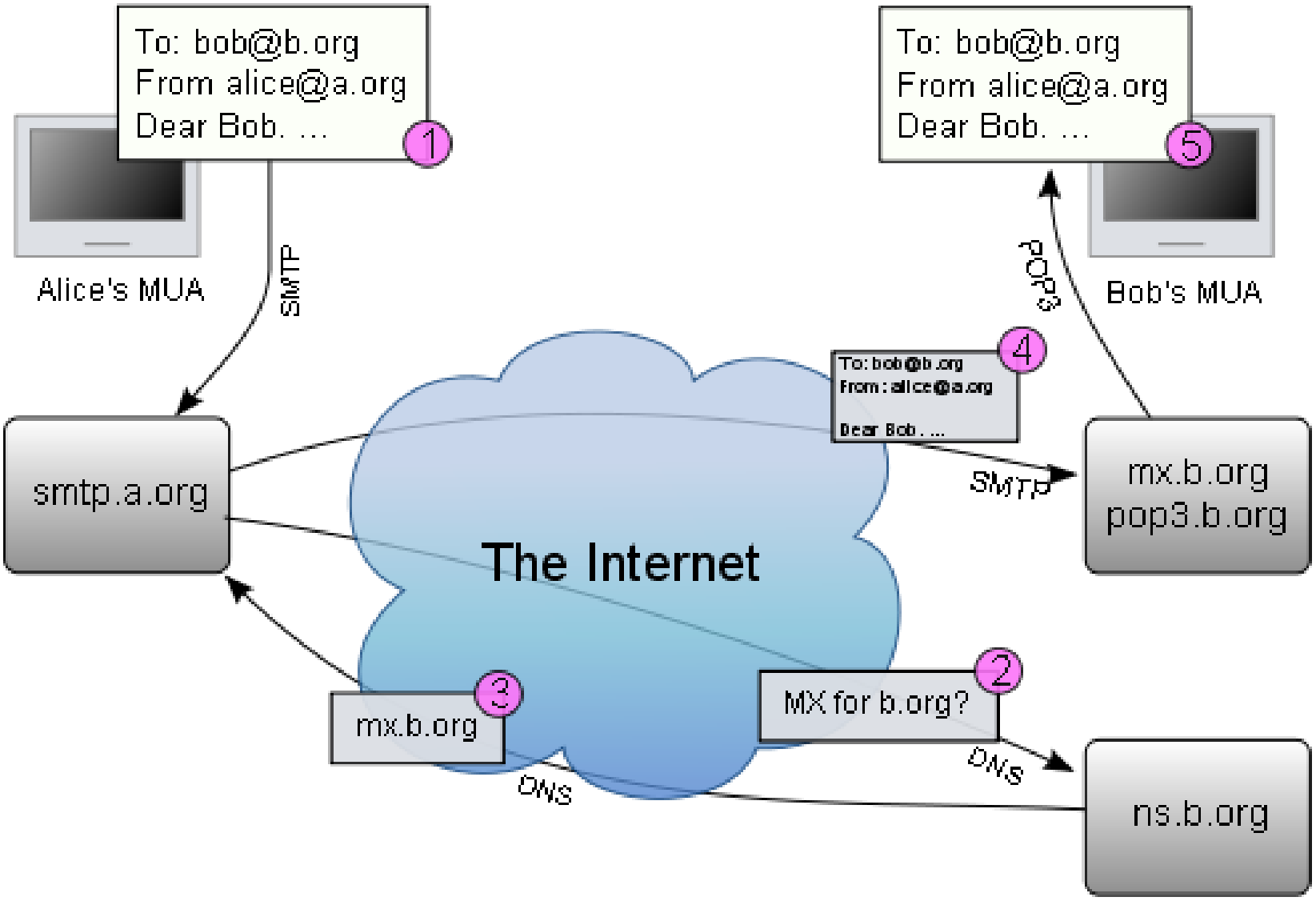
Disposition

- Forwarding, storing, deleting, re-reading of emails

TECHNISCHE UNIVERSITÄT DARMSTADT

Electronic mailbox

- person/addressee is assigned to an electronic mailbox
- address' form is "MAILBOX@COMPUTER"
  - unique
  - split into
    - "MAILBOX":
      - Mailbox name assigned only locally
      - in accordance with the respective local conventions
    - @ at
    - "COMPUTER"
      - for file transfer between systems
- address today in Internet is usually "MAILBOX@DOMAINNAME"
  - "MAILBOX"
  - @ at
  - "DOMAINNAME"
    - name of the destination domain
    - "domainname" is assigned the appropriate "computer" by being entered into the MX-record (MX = Mail eXchange) of the domain's DNS server


To: bob@b.org
From alice@a.org
Dear Bob. ...
1
Alice's MUA
SMTP
smtp.a.org
The Internet
3
mx.b.org
DNS
2
MX for b.org?
DNS
ns.b.org
4
To: bob@b.org
From: alice@a.org
Dear Bob. ...
SMTP
mx.b.org
pop3.b.org
POP3
5
To: bob@b.org
From alice@a.org
Dear Bob. ...
Bob's MUA

Basic Interaction

0. Sender Alice composes message using her mail user agent (MUA)

1. Mail user agent

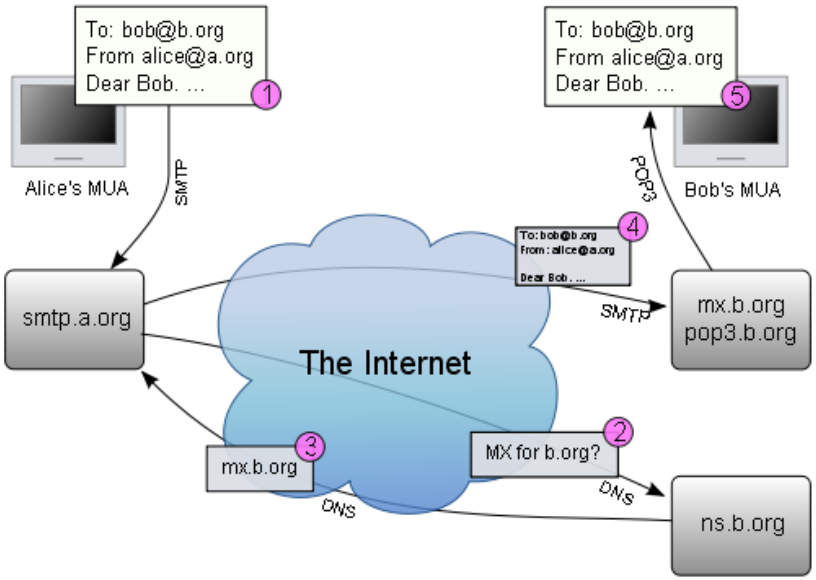
- formats the message in Internet e-mail format
- uses Simple Mail Transfer Protocol (SMTP) to send message to the local mail transfer agent (MTA),
  - in this case smtp.a.org, run by Alice's Internet Service Provider (ISP).

2. Local MTA

- looks at the destination address provided in the SMTP protocol (not from the message header),
  - in this case bob@b.org.
- looks up this domain name in the Domain Name System to find the mail exchange servers accepting messages for that domain.

3. DNS server for b.org domain, ns.b.org,

- responds with MX record listing the mail exchange servers for that domain,
  - in this case mx.b.org, a server run by Bob's ISP.



4. smtp.a.org

- sends the message to mx.b.org using SMTP, which delivers it to the mailbox of the user bob.

5. Receiver Bob retrieves message

- using his MUA, which may use
  - Post Office Protocol (POP3).

Simple Mail Transfer Protocol SMTP

a protocol for sending e-mail messages between servers

- SMTP is also used to send messages from a mail client to a mail server
- SMTP uses a TCP connection to port 25

consists of

- message format (ASCII presentation)
  - in 1982 defined in RFC 822
  - how the messages are structured
- data transfer protocol (ASCII presentation)
  - in 1982 defined in RFC 821
  - how the messages are transferred

Defined in RFC 822

Messages consist of

- an envelope
  - defined in RFC 821
- header fields (see the following table)
- one blank line
- message text
  - originally only 7 bit, i.e. 0-127
  - (extension see also MIME)

SMTP - Message Format & Structure: Header Fields

TECHNISCHE UNIVERSITÄT DARMSTADT

Header Field	Meaning
To:	**Recipient's email address (several addresses may be given).**
Cc:	**Carbon Copy. Email address of second recipient (several addresses may be given).**
Bcc:	**Blind Carbon Copy. Email address of recipients not supposed to be visible to the other recipients (deleted before delivery).**
From:	**Originator of the message.**
Sender:	**Sender of the message.**
Received:	**Displays the route a message has followed until then. A new line is added for each transfer agent.**
Return-Path:	**May be used to list a path back to the sender.**

- difference To: and Cc: solely psychological
- difference Cc: and Bcc: bcc line will be removed from the message and is thus not visible for the recipient
- Sender: and From: if these are one & the same, then sender omitted
- Return-Path: optional

SMTP – Message Format & Structure: Other Optional Header Fields

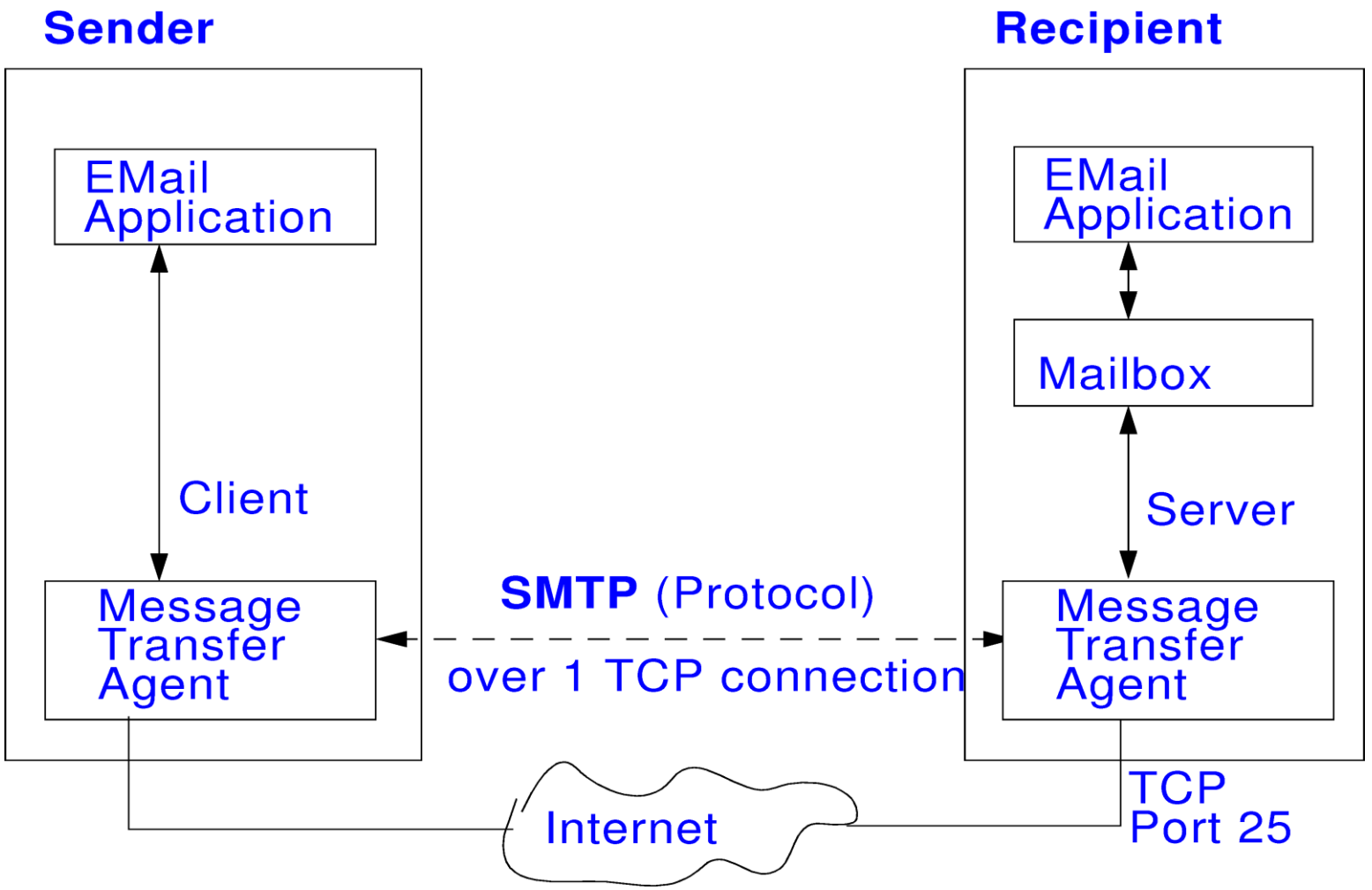
TECHNISCHE UNIVERSITÄT DARMSTADT

Header Field	Meaning
Date:	**Day and time when message was sent.**
Reply-To:	**Email address to which the response is to be sent.**
Message-Id:	**Unique number by which the message may be identified.**
In-Reply-To:	**Id of the message to which this message is a reply.**
References:	**Other relevant message Ids.**
Keywords:	**User defined keywords.**
Subject:	**Short summary of the contents.**

Based on RFC 822, additional (later defined) fields

- may be defined
- these fields have to start with X
  - examples:
    - X-No-Archive:
    - X-Auth:
    - X-SPAM:

e.g. simple example (no hop in between)



SMTP – Data/Mail Transmission

TECHNISCHE UNIVERSITÄT DARMSTADT

Steps

1. sender: application
- generates the message in the correct format
  - (often also the "mail user agent")
- may store a copy of the message that was sent

2. sender: transmission program
- distributes a copy of each message to each recipient
- e.g. "sendmail" in UNIX systems

3. receiver: email server
- receives message and files it in the appropriate mailbox

4. receiver: application
- reads mailbox
  - makes e.g. use of POP, IMAP protocols
- converts the messages into an adequate presentation

Transfer protocol (RFC 821)
- internet email is transferred over a TCP connection to Port 25

Transfer Over Several MTAs

TECHNISCHE UNIVERSITÄT DARMSTADT

i.e. route sender to receiver

- over several Mail Transfer Agents (MTA)

SMTP uses the store-and-forward principle to transfer messages

- identifies the sender
- verifies if receiver's mailbox exists

system name not always known, but domain is

- address usually "mailbox@domainname"
- domain name server
  - resource records:
    - information entered about the systems
  - among others that is Mail eXchange Record (MX-Record) with
    - information about preferred system nodes for accepting mail
    - i.e. possibly different systems with different priorities

3.3 SMTP - Characteristics

TECHNISCHE UNIVERSITÄT DARMSTADT

Characteristics

- all transferred characters are 7 bit ASCII
- commands consist of 4 letters
- forwarding option
- mailing list administration
- receiver confirms command with numerical value

Example:

→ **HELO mysystem.org (establish contact)**
← **250 flute.kom.tu-darmstadt.de Hello ...**

Problems:

- initial issue: message length limited to 64KB (in older versions)
- if sender and receiver have different timeouts
  - it may result in misunderstandings
- "mailstorms" may occur
  - for example because mailing lists refer to each other

Improvements on some of the above mentioned SMTP problems

- ESMTP (extended SMTP), defined initially in RFC 1425
- differentiation by contacting with (**E**xtended **H**E**LO**) EHLO (same syntax as HELO)

EHLO <systemname>

3.4 SMTP - Example Protocol of Direct Interaction

TECHNISCHE UNIVERSITÄT DARMSTADT

```
[saxophon] > TELNET TUBA 25
 Trying 130.83.139.132...
 Connected to tuba.kom.tu-darmstadt.de.
 Escape character is '^]'.
 220 mailserver.KOM.tu-darmstadt.de ESMTP
 Sendmail 8.12.6/8.12.6; Mon, 9 Dec 2008 13:58:09
 +0100 (MET)

HELO SAXOPHON.KOM.TU-DARMSTADT.DE
 250 mailserver.KOM.tu-darmstadt.de Hello
 saxophon.kom.tu-darmstadt.de
 130.83.139.133, pleased to meet you

MAIL FROM: <DIETER.SCHULLER@SAXOPHON>
250 <dieter.schuller@saxophon>... Sender ok

RCPT TO: <GROSS>
250 <gross>... Recipient ok

DATA:
500 Command unrecognized
```

SMTP - Example Protocol of Direct Interaction

TECHNISCHE UNIVERSITÄT DARMSTADT

```
DATA
  354 Enter mail, end with "." on a line by itself

TESTMAIL
THIS MAIL TESTS THE MAIL SYSTEM
.
  250 OAA20896 Message accepted for delivery

QUIT
  221 mailserver.KOM.tu-darmstadt.de closing connection

[Connection closed by foreign host.]
[saxophon]~ >
```

Example of sent message:

```
From rst Fri Jun 17 08:34:50 2008
Subject: Lecture CN II
To: eveking@maigret.rs.tu-darmstadt.de (H. Eveking)
Date: Sat, 18 Jun 2008 17:48:50 +0100 (MET)
Cc: monika.jayme@kom.tu-darmstadt.de (Monika Jayme)
Cc: jan.baum@kom.tu-darmstadt.de (Jan Baum)
X-Mailer: ELM [version 2.4 PL25]
MIME-Version: 1.0
Content-Type: text/plain; charset=ISO-8859-1
Content-Transfer-Encoding: 8bit
Content-Length: 1139

The second exercise re. CN II is ambiguous:
..
best regards Ralf
```

SMTP: Example of Received Message

```
From eveking@maigret.rs.tu-darmstadt.de Fri Jun 17
10:30:32 2008
X-UIDL: ee1a889ea7fece3665d9aaeaa3c558c4
Return-Path: eveking@maigret.rs.tu-darmstadt.de
Received: from KOM.tu-darmstadt.de by
mailserver.KOM.tu-darmstadt.de (8.12.6/8.12.6) with
ESMTP id KAA01703 for <Ralf.Steinmetz@KOM.tu-darmstadt.
de>; Fri, 17 Jun 2008 10:30:30 +0100 (MET)
Received: from mailhost.rs.TU-Darmstadt.DE by
gatekeeper (8.12.6/8.12.6) with ESMTP id KAA26173 for
<Ralf.Steinmetz@KOM.tu-darmstadt.de>; Fri, 17 Jun 2008
10:26:48 +0100 (CET)
Received: from maigret.rs.TU-Darmstadt.DE (maigret
[130.83.34.40]) by mailhost.rs.TU-Darmstadt.DE
(8.12.6/8.12.6) with SMTP id KAA28568
for <Ralf.Steinmetz@KOM.tu-darmstadt.de>; Fri, 17 Jun
2008 10:30:30
+0100 (MET)
```

SMTP: Example of Received Message

```
Received: by maigret.rs.TU-Darmstadt.DE (5.x/SMI-SVR4)
id AA04555; Fri, 17 Jun 2008 10:30:28 +0100
Date: Fri, 17 Jun 2008 10:30:28 +0100
From: eveking@maigret.rs.tu-darmstadt.de (H. Eveking)
Message-Id: <9801160930.AA04555@maigret.rs.TUDarmstadt.
DE>
To: Ralf.Steinmetz@KOM.tu-darmstadt.de
Subject: Re: Lecture CN II
X-Sun-Charset: US-ASCII
Status: OK

May even be an error.
```

3.6 Electronic Mail - Critical Issues of Classical SMTP

TECHNISCHE UNIVERSITÄT DARMSTADT

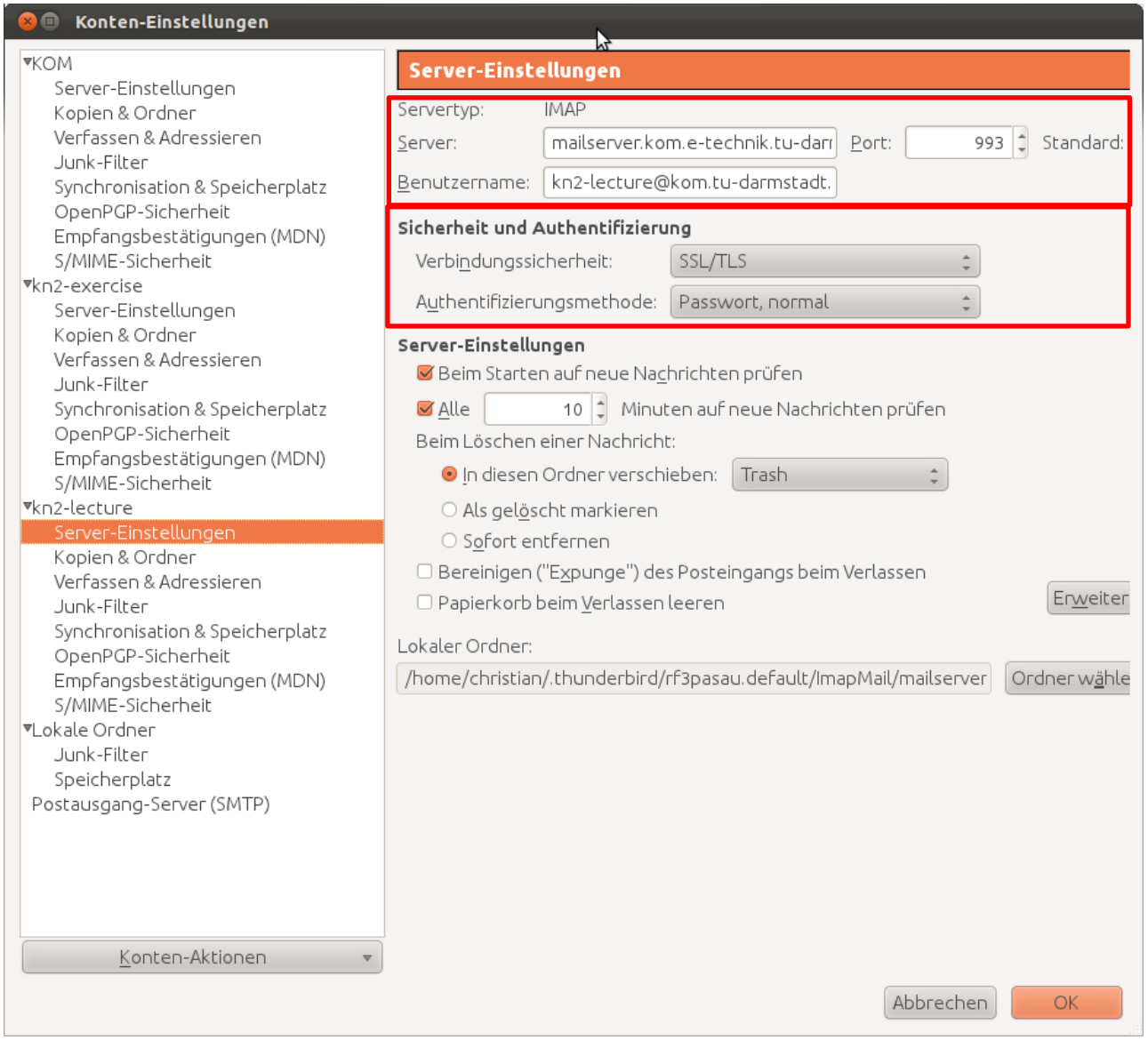
With SMTP and original message format

- sending a message to various recipients
  - done by sending same data to all of them individually
- messages do not have internal structure

Makes automatic processing difficult

- no acknowledgement
  - sender does not know if the message he sent has actually been received by the recipient
- message rerouting arduous ("mühsam")
- user interface not integrated in transfer system
- no way to send message containing a mixture of text, graphics and audio
- messages may contain ASCII characters only
  - no accents or special characters ä,ö.ü, etc.(e.g. French, German)
  - no non-latin alphabets
    - e.g. Hebraic
  - no possibility to present languages that are not bound by an alphabet
    - e.g. Chinese, Japanese

Post Office Protocol POP / Internet Message Access Protocol IMAP



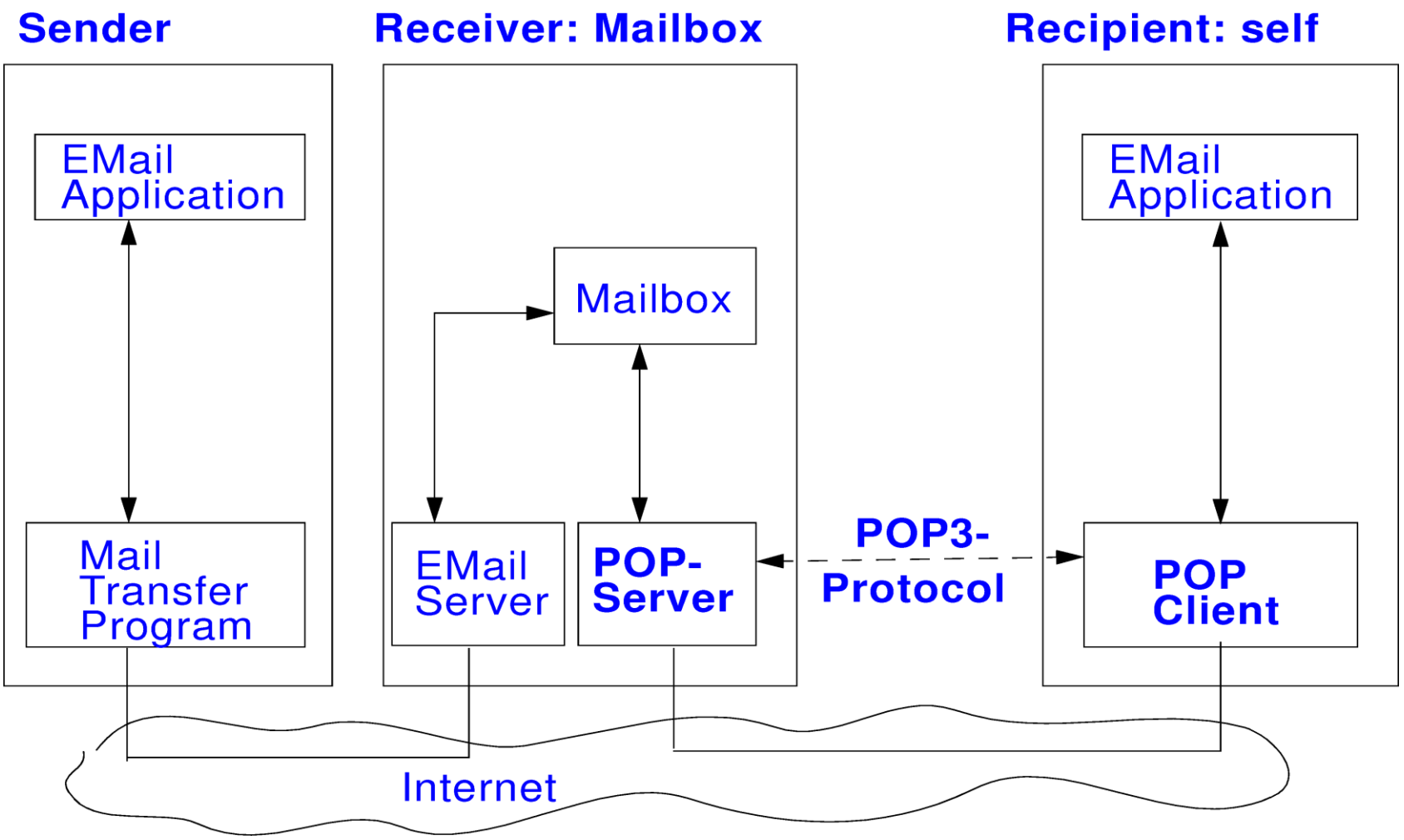
Motivation

- user (mail recipient) uses different systems
  - but his mailbox should always be the same
- server has to run reliably for 24 hours
  - but not necessarily his system
- mailbox and applications
  - often on different systems

Protocol for remote mailbox access

- user (usually) transfers mail for further processing
  - to his local system
- this transfer is defined in a protocol: Post Office Protocol (POP)
- characteristics
  - access permitted only after authentication
  - can provide information about contents without actually transferring them
  - POP uses Port 110, SSL encrypted Port 995
  - POP goes through three states: authorization, transaction and update

POP Interaction



4.2 IMAP - Internet Message Access Protocol

Motivation

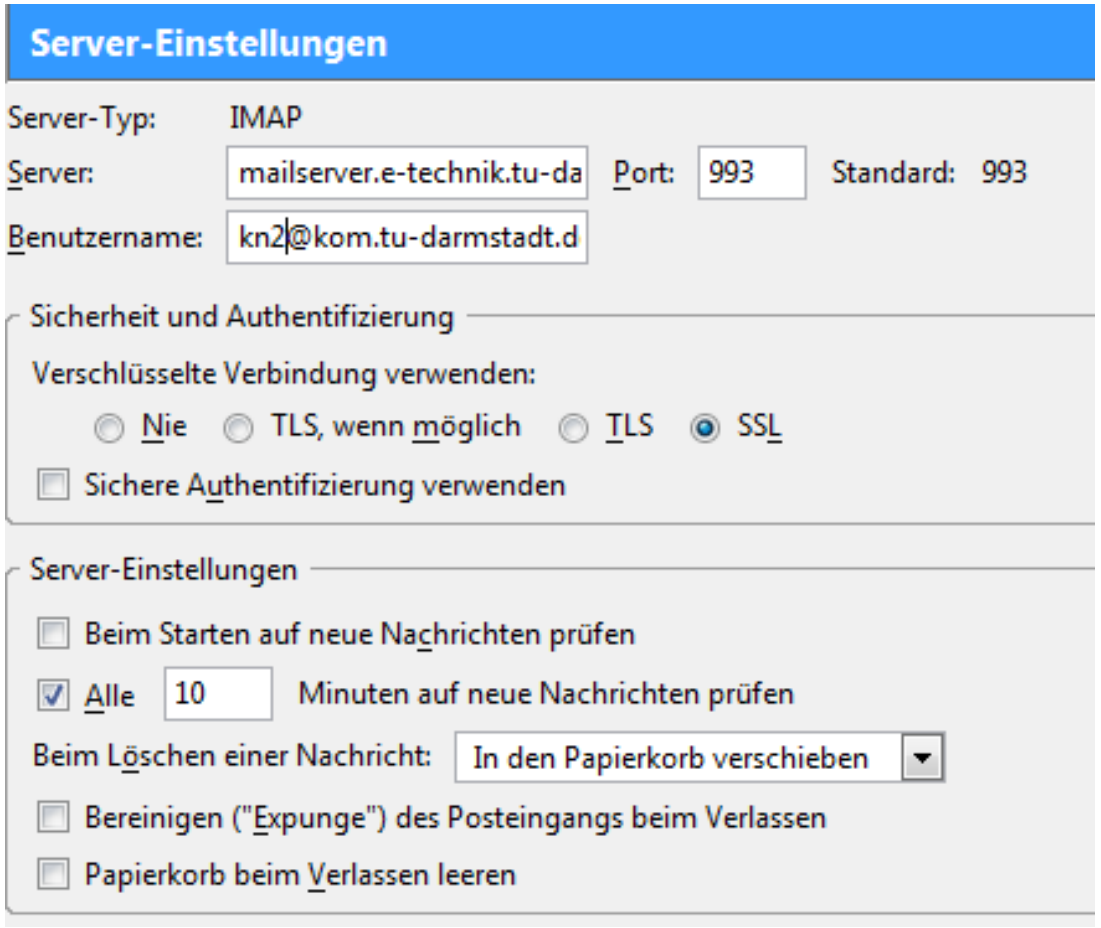
- electronic letters remain on the server
- that means that server management is necessary

IMAP:

- Interactive Mail Access Protocol
- used to retrieve e-mail
  - from a mail server
- alternatively to POP
- RFC 2060

characteristics

- port
  - port 143
  - SSL encrypted Port 993
- security problem
  - access to server data
  - possible actions:
    - copy, delete, move



4.3 Comparison of IMAP and POP3

TECHNISCHE UNIVERSITÄT DARMSTADT

Feature	POP3	IMAP
Where is protocol defined	RFC 1939	RFC 2060
TCP port used	110	143
Email store on	User's PC	Server
E-mail read	Off-line	On-line
Connect time required	Little	Much
Use of server resources	Minimal	Extensive
Multiples mailboxes	No	Yes
…	…	…

Source: [Tanenbaum03]

Defined in RFC 1341 and modified in RFC 2045 to 2049

Possibilities:

- messages may contain non ASCII character
  - accents or special characters ä, ö, ü, etc. (e.g. French, German)
  - non-latin alphabets
    - e.g. Hebraic
  - languages that are not bound by an alphabet
    - e.g. Chinese, Japanese
- messages that may contain audio data, video data or general data

Idea:

- using the format defined in RFC 822 for messages
- add structure to the message body
- define rules for coding non-ASCII messages

→ **(only) programs for generating & displaying messages to be modified**

→ **Programs for sending and receiving remain unmodified**

TECHNISCHE UNIVERSITÄT DARMSTADT

Chosen approach:

- MIME messages consist of multiple parts
- Each part may have a different type: text, audio, image, ...

Content types:

- Text (subtypes: plain, richtext)
- Image (subtypes: gif, jpeg)
- Audio (subtypes: basic)
- Video (subtypes: mpeg, h261)
- Message (subtypes: partial, external-body)
- Multipart (subtypes: mixed, alternative, parallel)
- Application (subtypes: postscript, Octet-Stream)

Subtypes:

- Additional subtypes can be registered
- Designated subtypes for private usage

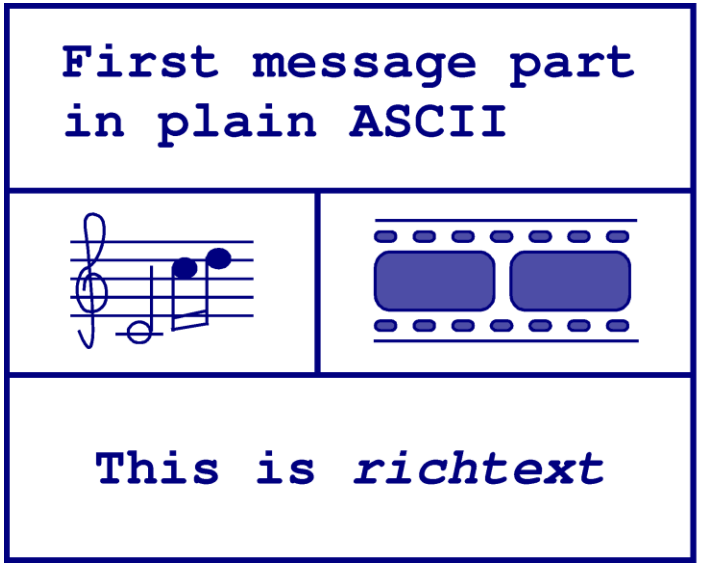
MIME Message: Example

TECHNISCHE UNIVERSITÄT DARMSTADT

MIME message must include

- Data in multiple message parts
- Definition of content types of individual parts
- Boundaries between parts

Structure of an example message:



1.) ASCII text

2.) audio and video in parallel

3.) Richtext text

sequential display

MIME Message: Example

```
Content-type: multipart/mixed;

--unique-boundary-1--
Content-type: text/plain
First message part in plain ASCII.
--unique-boundary-1--
Content-type: multipart/parallel;

--unique-boundary-2--
Content-Type: audio-basic

... base64-encoded audio data goes here ...
--unique-boundary-2--
Content-Type: image/jpeg

... base64-encoded image data goes here ...
--unique-boundary-2--
--unique-boundary-1--
Content-Type: text/richtext
This is <italic>richtext.</italic>
--unique-boundary-1--
```

- Boundaries between message parts
- Definition of content types
- Data

5.2 MIME - Header Fields

TECHNISCHE UNIVERSITÄT DARMSTADT

Header Field	Meaning
MIME-Version:	**Identifies the MIME version**
Content-Description:	**Legible description of the message**
Content-Id:	**Unique number to identify the message**
Content-Transfer-Encoding:	**Encoding type**
Content-Type:	**Message type**

MIME: Header Fields

TECHNISCHE UNIVERSITÄT DARMSTADT

MIME version:

- necessary to identify the message as a MIME message
- example:
  - MIME-Version: 1.0

Header Field	Meaning
MIME-Version:	Identifies the MIME version
Content-Description:	Legible description of the message
Content-Id:	Unique number to identify the message
Content-Transfer-Encoding:	Encoding type
Content-Type:	Message type

Content description:

- example:
  - Content-Description: A picture of my guinea pig

Header Field	Meaning
MIME-Version:	Identifies the MIME version
Content-Description:	Legible description of the message
Content-Id:	Unique number to identify the message
Content-Transfer-Encoding:	Encoding type
Content-Type:	Message type

Content Id:

- Unique message ID

Header Field	Meaning
MIME-Version:	Identifies the MIME version
Content-Description:	Legible description of the message
Content-Id:	Unique number to identify the message
Content-Transfer-Encoding:	Encoding type
Content-Type:	Message type

MIME Header Fields: Content-Transfer-Encoding

TECHNISCHE UNIVERSITÄT DARMSTADT

Content-Transfer-Encoding in 5 different types:

Type	Way
ASCII text	**7-bit ASCII**
ASCII text with 8 bit	**8-bit ASCII violates protocol specification**
binary	**any desired 8-bit violates protocol specification**
quoted-printable	**ASCII presentation for short 8-bit information**
base64 (ASCII armor)	**ASCII presentation for 8-bit information**

e.g. quoted-printable:

- 7-bit ASCII
- all characters > 127:
  - presented as XXh
  - with XXh as a hexadecimal number representing the character
- encoder for quoted-printable:
  - http://www.motobit.com/util/quoted-printable-encoder.asp

Header Field	Meaning
MIME-Version:	Identifies the MIME version
Content-Description:	Legible description of the message
Content-Id:	Unique number to identify the message
Content-Transfer-Encoding:	Encoding type
Content-Type:	Message type

Header Fields: Content-Transfer-Encoding: base64

TECHNISCHE UNIVERSITÄT DARMSTADT

e.g. base64:

- information viewed as a data stream
- 64 characters are used (i.e. 2^6 =64)
  - = (equal) has special function, i.e.
    - = = last group contained only 8 bits
    - = last group contained only 16 bits
- 3 bytes which need to be coded (24 Bit) divided into four 6-bit groups
- carriage return (CR) and line feeds (LF) are ignored

example

- .. next slide

Example for Base 64 Encoding

TECHNISCHE UNIVERSITÄT DARMSTADT

Example:

- Text encoded in ASCII
- Regrouping of bits
  - Every 24 bits = 3*8 bits = 4*6 bytes
- Interpret every 6 bits as new character identified by its number
  - A = 00000 = 0
  - B = 00001 = 1
  - C = 00010 = 2
  - ….
  - + = 111110 = 62
  - / = 111111 = 63

Text	**M**								**a**								**n**							
ASCII	77								97								110							
Bit pattern	0	1	0	0	1	1	0	1	0	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0
Index	19				22				5				46											
Base64-encoded	T				W				F				u											

Example for Increasing Size of Base64 Encoded Content

TECHNISCHE UNIVERSITÄT DARMSTADT

Example:

- Sending 2431 bytes of data
- 2431 is not a multiple of 3 → add 2 additional bytes → 2433 bytes to be transferred
- 2433 byte * 8 bit / byte = 19464 bit
- Determine number of 6 bit blocks: 19464 bit / 6 bits per block = 3244 blocks
- For each 6 bit block add 2 bits to complete full byte = 3244 * 2 bit additional data → 19464 + 3244*2 = 25952 bit = 3244 byte
- Data is sent in lines of 76 bytes:
  - For every 76 bytes add additional CR + LF (2 bytes):
    - 3244 byte + ROUND_INT_UP(3244 byte / 76 byte per line) * 2 bytes per line = 3330 bytes to be transmitted
- Comparison of size: 3330 bytes / 2431 bytes = 1,368

Size of encoded content increases by ~ 37%

TECHNISCHE UNIVERSITÄT DARMSTADT

example base 64

```
Content-type: application/msword; name="A000001.doc"
Content-Disposition: attachment; filename=A000001.doc
Content-transfer-encoding: base64
0M8R4KGxGuEAAAAAAAAAAAAAAAAAAAAAPgADAP7/
CQAGAAAAAAAAAAAAAACAAAAhgAAAA
AAAAAAEAAAiAAAAAEAAAD+////AAAAAIQAAACFAAAA//////////////////////
//////
//////////////////////////////////////////////////////////////////
//////
//////////////////////////////////////////////////
spcEAcQAHBAAACBK/
AAAAAAAAEAAAAAAABAAAm0YAAA4AYmpianQrdCsAAAAAAAAAAAAAAAAAAAAAAAAH
BBYAQo
0AABZBAQAWQQEANUIAAAAAAABlAAAAAAAAAAAAAAAAAAAAAAAAAAAAAD//
w8AAAAAAAAA
AAD//w8AAAAAAAAAAD//
w8AAAAAAAAAAAAAAAAAAAAAF0AAAAAAPADAAAAAAAA8AMAAP
ADAAAAAAAA8AMAAAAAADwAwAAAAAAAPADAAAAAAAA8AMAAJQAAAAAAAAAAAAAAK
4FAAAA
AAAArgUAAAAAACuBQAAAAAAAK4FAAD4AAAApgYAAEQAAADqBgAAhAAAAK4FAAAA
AAAAoT
```

Header Fields: Content-Type

TECHNISCHE UNIVERSITÄT DARMSTADT

Examples

Type	Subtype	Description
Text	Plain	Unformatted text
	Richtext	Text with simple formatting commands in SGML
Image	Gif	Image in GIF format
	Jpg	Image in JPG format
Audio	Basic	Audio
Video	Mpeg	Video in MPEG format
Application	Octet-Stream	Uninterpreted byte stream
	Postscript	Printable document in Postscript format
Message	Rfc822	A MIME RFC 822 message
	Partial	This message has been split for transmission
	External body	This message has to be retrieved from the network
Multipart	Mixed	Independent parts in the specified order
	Alternative	Same message but different formats
	Parallel	Parts have to be presented parallel
	Digest	Each part is a full RFC 822 message

Header Field	Meaning
MIME-Version:	Identifies the MIME version
Content-Description:	Legible description of the message
Content-Id:	Unique number to identify the message
Content-Transfer-Encoding:	Encoding type
Content-Type:	Message type

MIME: Examples

Example:

```
Content-Type: text/richtext
"I am an <bold>owl </bold>", said the
<italic>walrus</italic>.
```

results in

"I am an **owl**", said the *walrus*.

MIME Example: Sent Text Message

```
From: matthias.hollick@saxophon.kom.tu-darmstadt.de
To: ralf.steinmetz@tuba.kom.tu-darmstadt.de
MIME-Version: 1.0
Message-Id: <199707011607.SAA20302@saxophon.kom.tu-
darmstadt.de>
Content-Type: multipart/alternative; boundary=
"------------1DA8FCD5D4D"
This is a preamble, ignored by the user agent.
-----------1DA8FCD5D4D
Content-Type: text/richtext
"I am an <bold>owl</bold>", said the
<italic>walrus</italic>.
The marabu nodded <italic>wisely</italic> and said:
"I am an owl, too!"
```

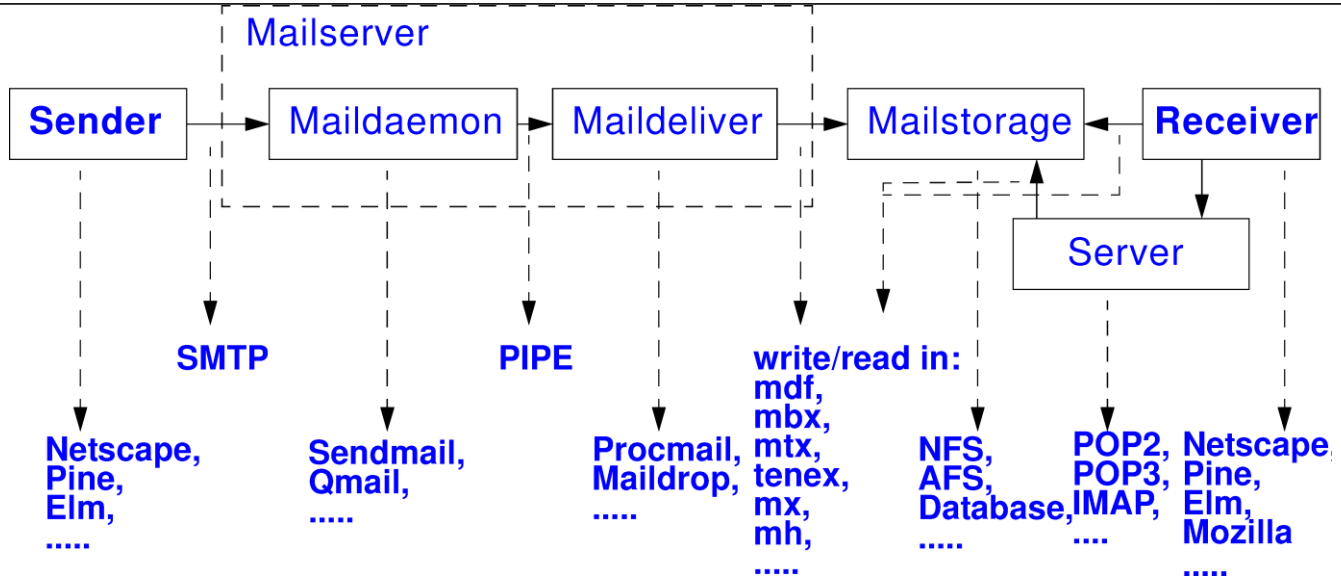
MIME Example: Sent Audio Message

```
From: ralf.steinmetz@tuba.kom.tu-darmstadt.de
To: matthias.hollick@saxophon.kom.tu-darmstadt.de
MIME-Version: 1.0
Message-Id: <199707011607.SAA20302@saxophon.kom.tu-
darmstadt.de>
Content-Type: multipart/alternative; boundary=
"------------1DA8FCD5D4D"
This is the preamble, ignored by the user agent
-----------1DA8FCD5D4D
Content-Type: message/external-body;
access-type="anon-ftp";
site="ftp.kom.tu-darmstadt.de";
directory="/pub/eulen";
name="am_owls_too.snd"
Content-Type: audio/basic
content-transfer-encoding: base64
```

Topics

- Implementation issues
- History
  - X.400
- Other Concepts
  - References
  - Security

6.1 Mail Implementation Overview



Overall mailing process

- of an environment used daily

6.2 X.400 Mail

TECHNISCHE UNIVERSITÄT DARMSTADT

History

- defined 2 years after RFC 821 and RFC 822 (1984)
- idea: to correct the disadvantages of the above RFC's

Supported by:

- CCITT - ITU
- telecommunication corporations, governments, industry

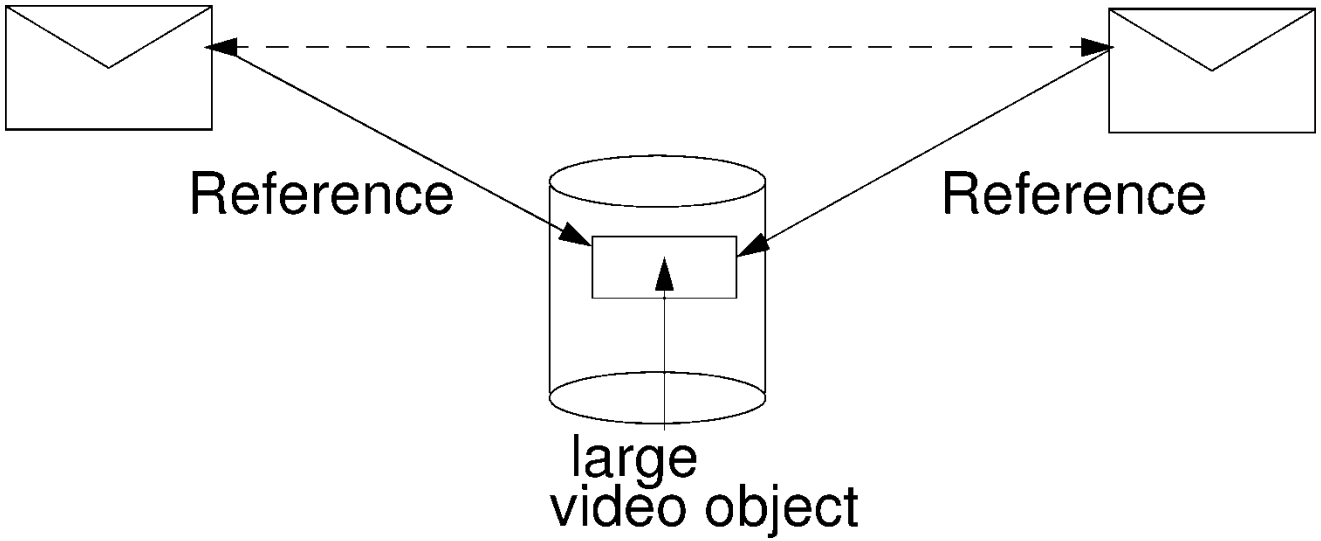
De facto today: X.400 not very widespread anymore

- reasons:
  - poor design
  - extremely complex
  - SMTP had prevailed

Pragmatic decision

- simple but functioning system (YES) or
- beautiful but very complex functioning system

6.3 Referenced Based Mailing



Challenge:

- many objects have a high amount of data (e.g. video)
- receiver has only a limited storage capacity

Solution: global store

- can be realized by url
- but:
  - contents may not necessarily be available
- future
  - combined content management systems & workflow environments

Motivation

- ASCII text is easy to read
  - by e.g. any sniffer
- is the sender really the one it claims it is?

S/MIME

- based on strictly hierarchic certification, X.509 certificates
  - just like SSL

OpenPGP

- Open Pretty Good Privacy
- based on “web of trust”
  - user decides which certification entity he can trust