Computer Networks and Internet Technology

2021W703033 VO Rechnernetze und Internettechnik
Winter Semester 2021/22

Jan Beutel

universität innsbruck

With materials and inspiration from Scott Shenker, Jennifer Rexford, Phil Levis, Adrian Perrig and Laurent Vanbever

Communication Networks and Internet Technology

Recap of last weeks lecture

Web

Video Streaming

http://www.google.at

HTTP performance goals vary depending on who you ask

	User	Network operators	Content provider
wish	fast downloads high availability	no overload	happy users cost-effective infrastructure
solution	Improve HTTP to compensate for TCP weakspots	Caching and Replication	

Considering the time to retrieve *n* small objects, pipelining wins

	# RTTS
one-at-a-time	~2*n*
M concurrent	~2*n*/M
persistent	~*n*+1
pipelined	2

Considering the time to retrieve *n* big objects, there is no clear winners as bandwidth matters more

# RTTS

$$\frac{\sim n * \text{avg. file size}}{\text{bandwidth}}$$

To limit staleness of cached objects, HTTP enables a client to validate cached objects

Server hints when an object expires (kind of TTL)

as well as the last modified date of an object

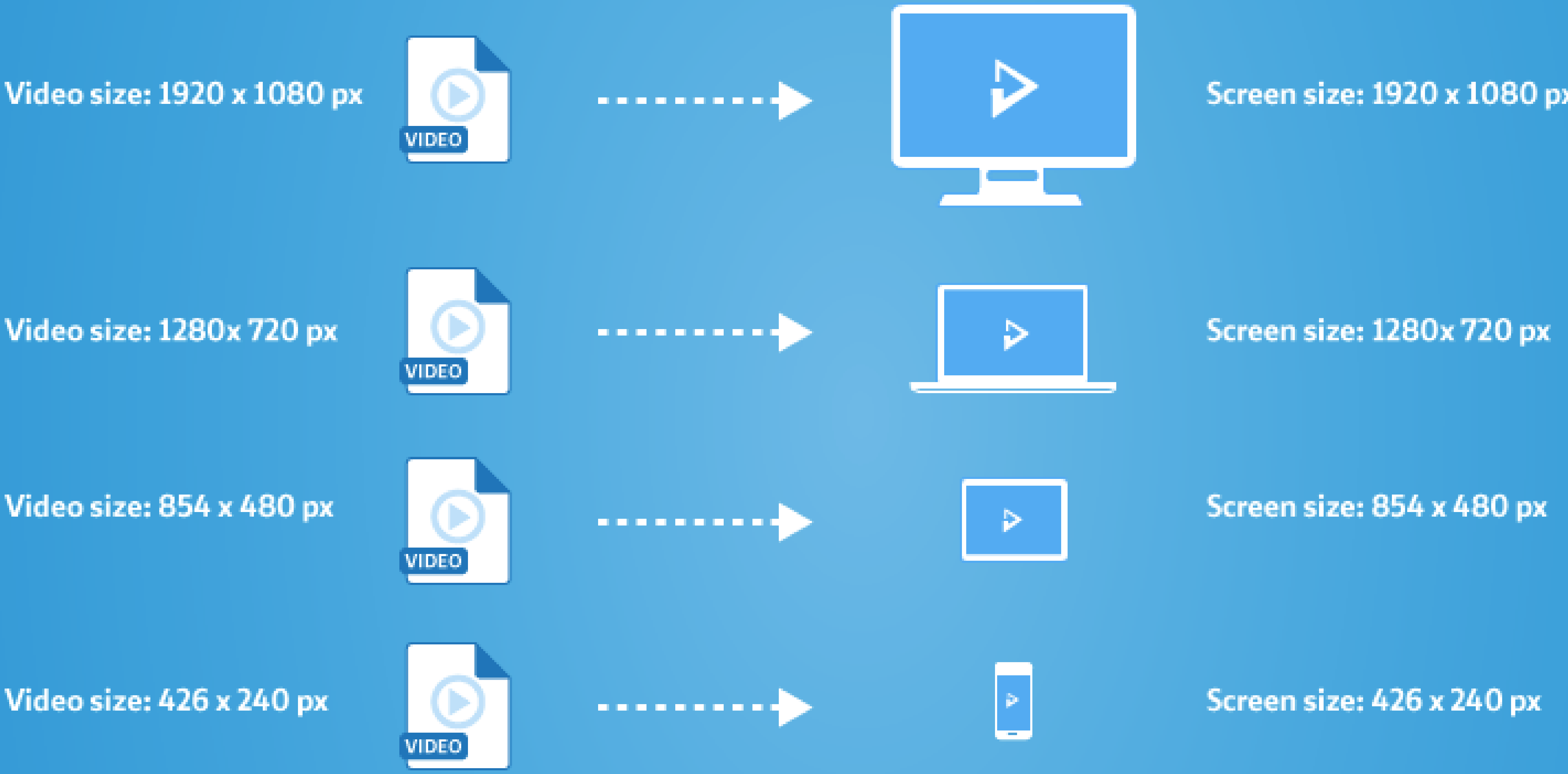
Client conditionally requests a ressources

using the “if-modified-since” header in the HTTP request

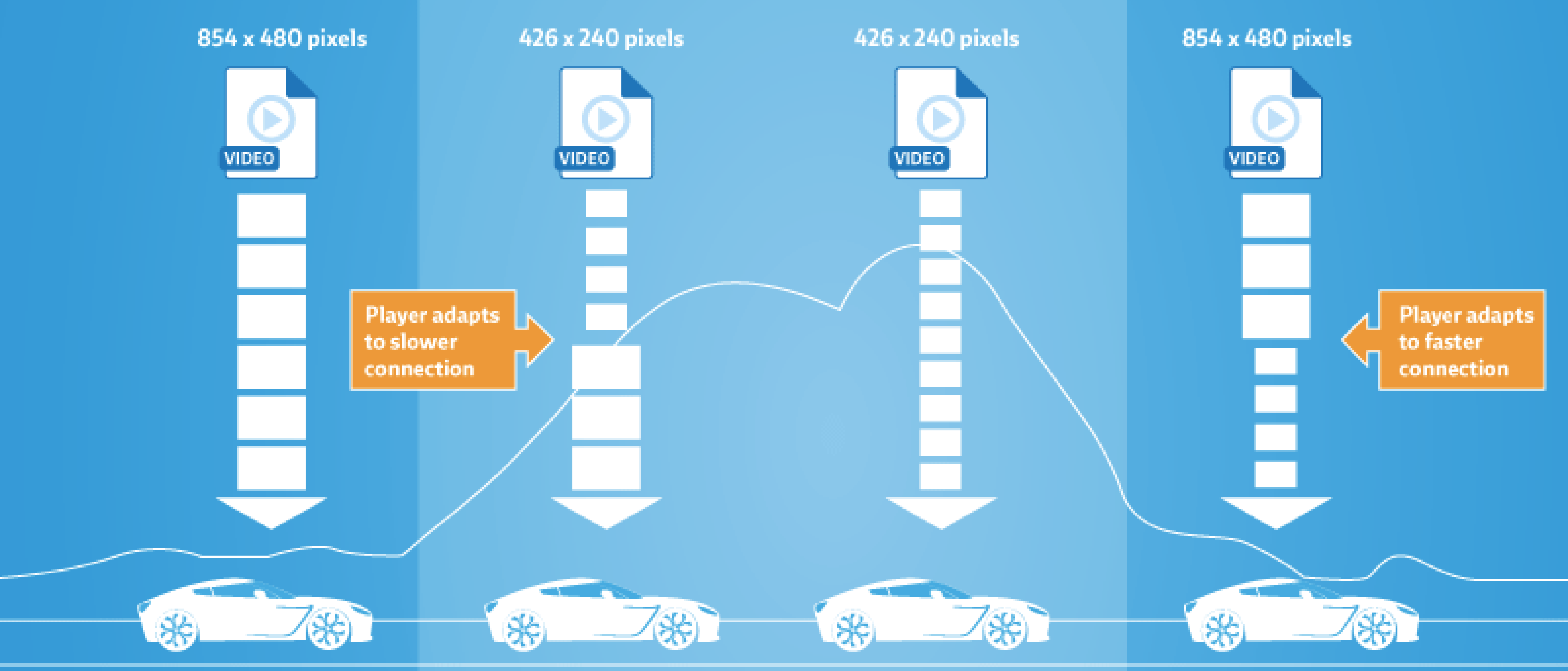
Server compares this against “last modified” time

of the resource and returns:

- Not Modified if the resource has not changed
- OK with the latest version


Video size: 1920 x 1080 px
VIDEO
Screen size: 1920 x 1080 p
Video size: 1280x 720 px
VIDEO
Screen size: 1280x 720 px
Video size: 854 x 480 px
VIDEO
Screen size: 854 x 480 px
Video size: 426 x 240 px
VIDEO
Screen size: 426 x 240 px

[bitmovin.com]



Normal connection:
The Player downloads the best quality video

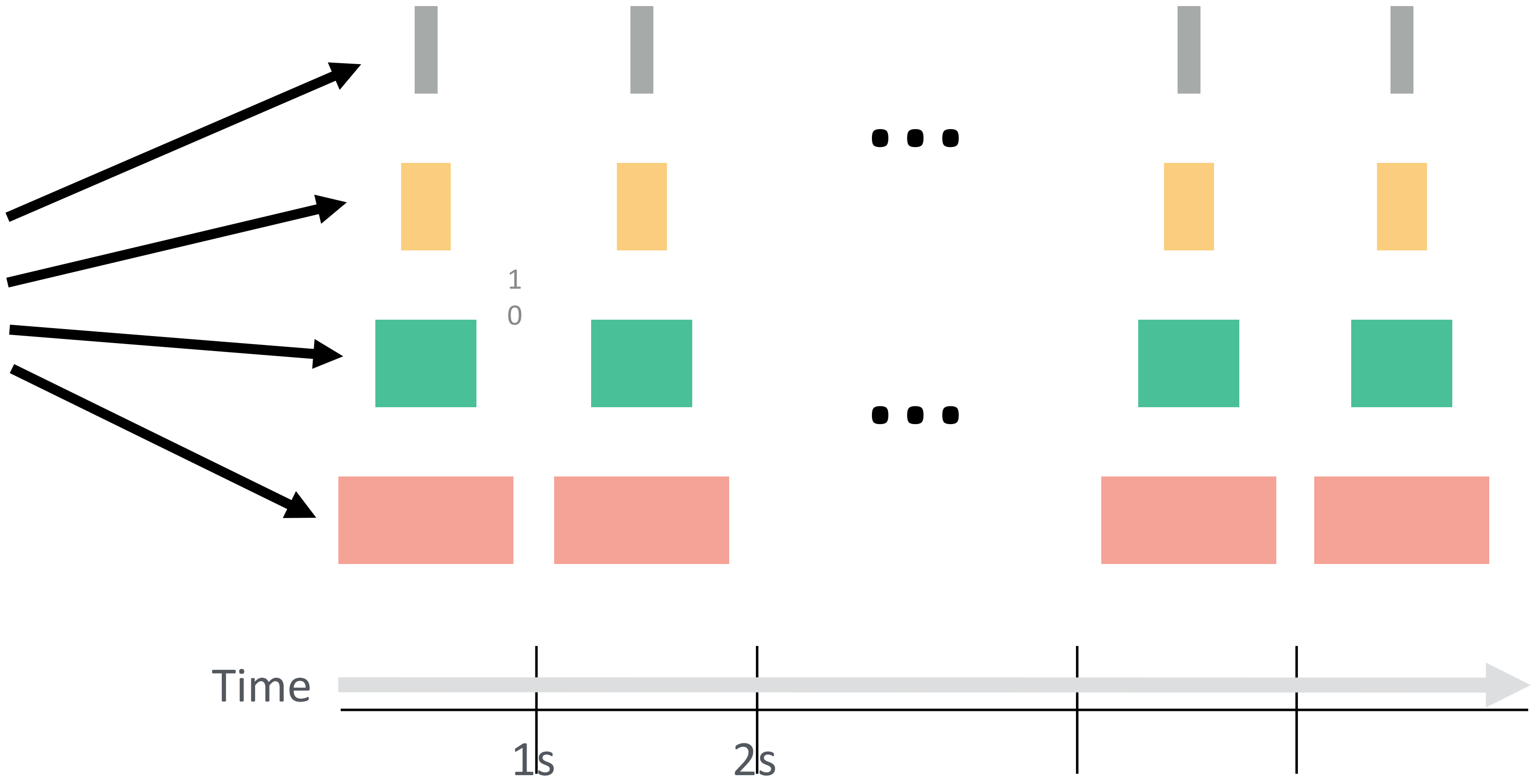
Poor connection:
The Player changes to downloading a smaller, faster video file

Normal connection:
The Player returns to the maximum quality video file

[bitmovin.com]

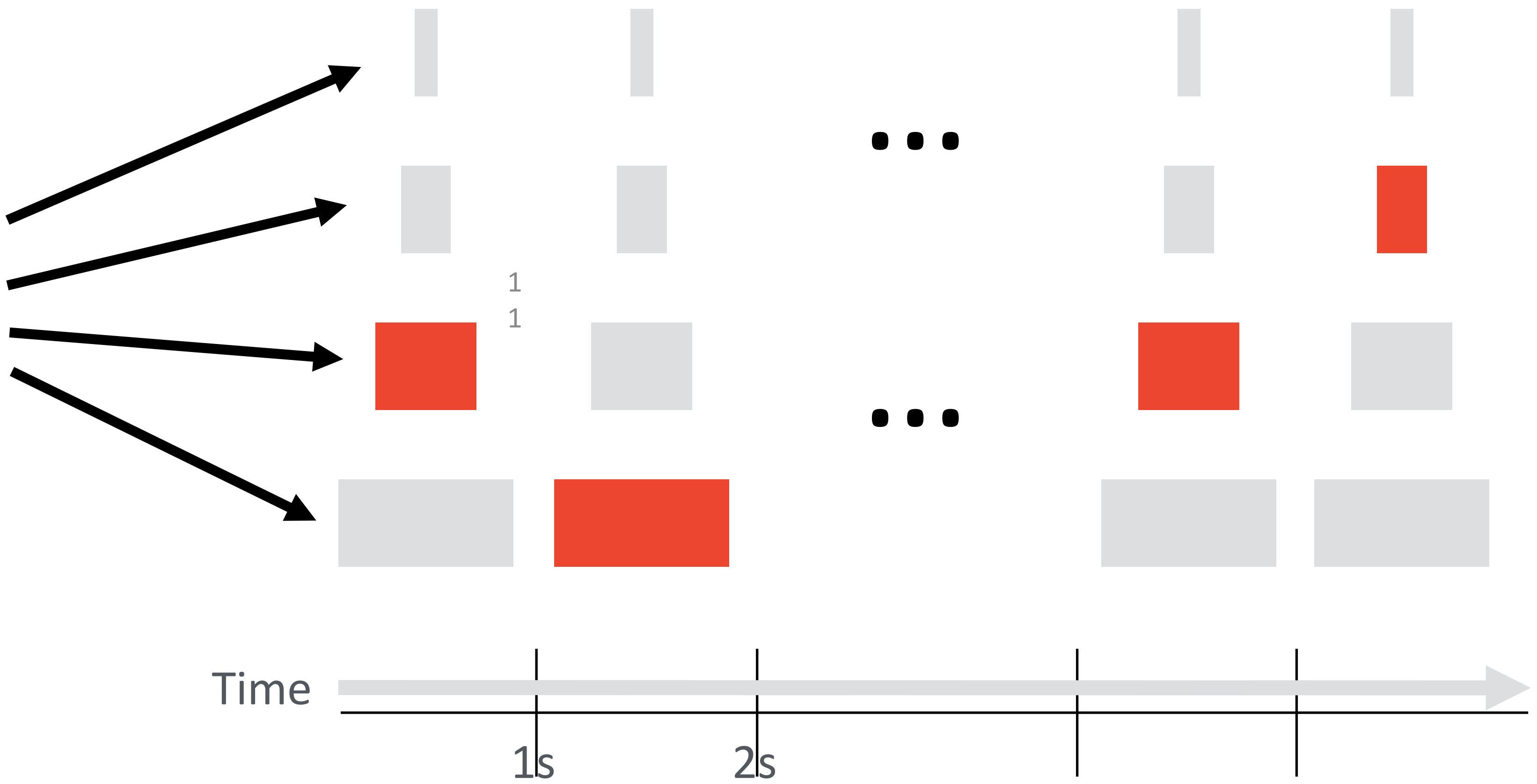
Your player download “chunks” of video at different bitrates

[netflix.com]



Depending on your network connectivity, your player fetches chunks of different qualities

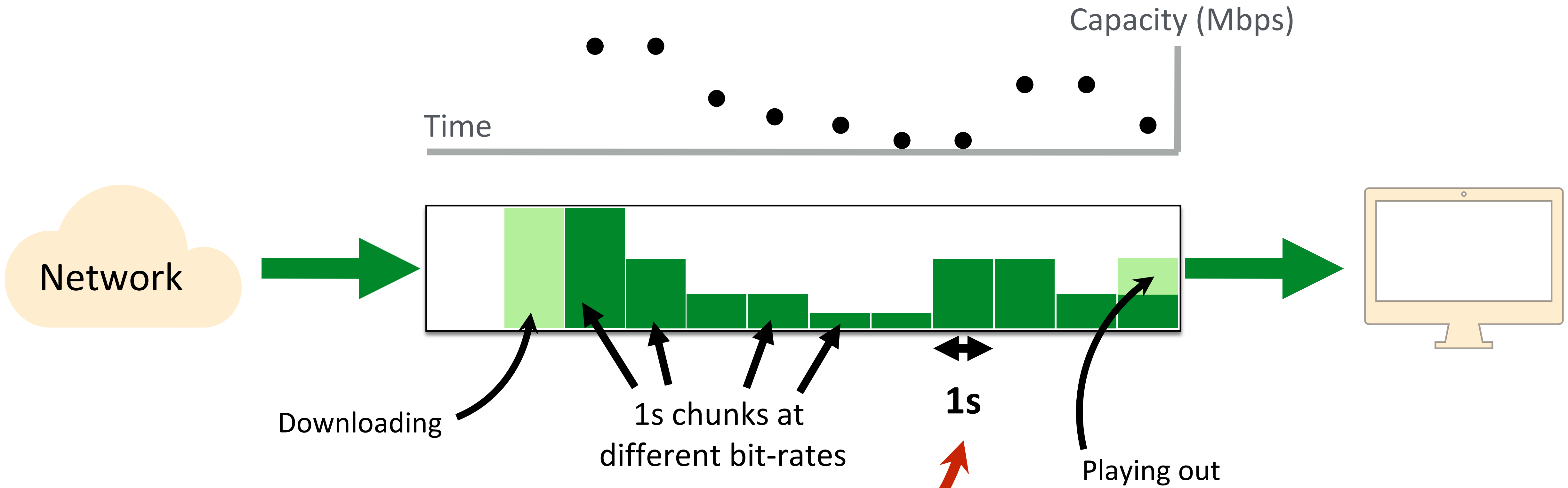
[netflix.com]



Encoding

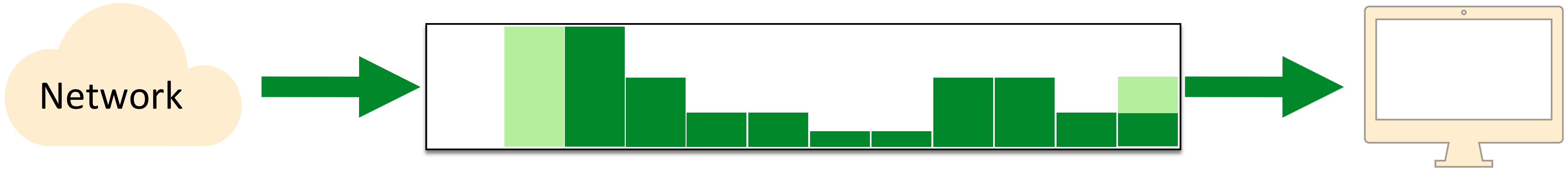
Replication

Adaptation



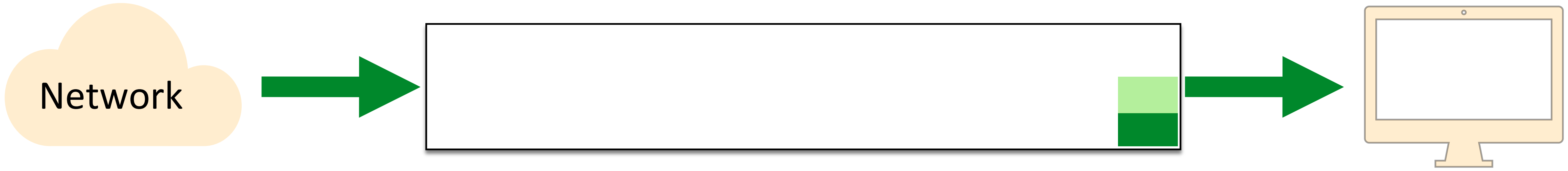
Capacity < current rate ⇒ decrease rate

Buffer-based adaptation



Nearly full buffer ⇒ large rate

Buffer-based adaptation



Nearly empty buffer ⇒ small rate

Communication Networks and Internet Technology

This weeks lecture

E-mail

MX, SMTP, POP, IMAP

E-mail

MX, SMTP, POP, IMAP

We'll study e-mail from three different perspectives

Content	Infrastructure/ Transmission	Retrieval
Format: Header/Content	SMTP: Simple Mail Transfer Protocol	POP: Post Office Protocol
Encoding: MIME	Infrastructure mail servers	IMAP: Internet Message Access Protocol

Content

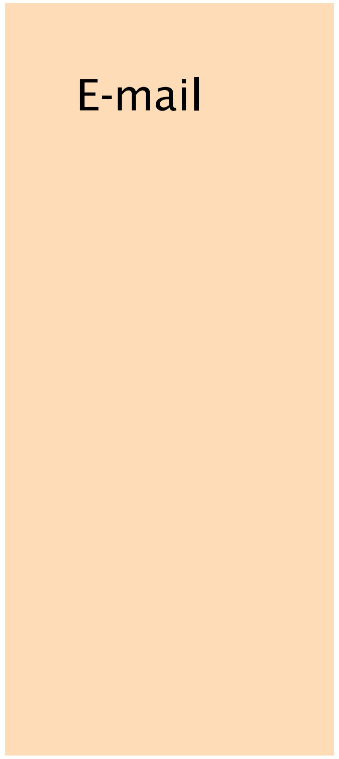
Infrastructure/
Transmission

Retrieval

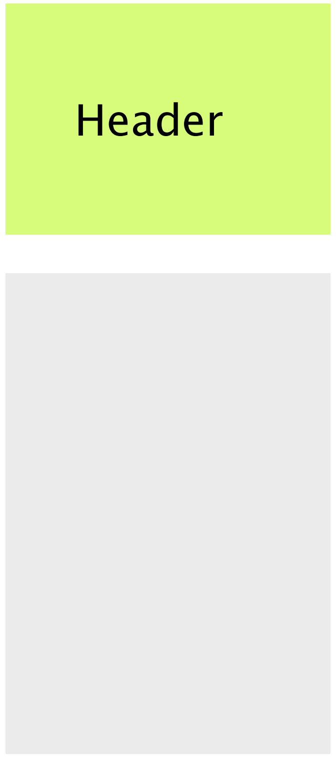
Format: Header/Content

Encoding: MIME

An e-mail is composed of two parts



A header, in 7-bit U.S. ASCII text



From:	Jan Beutel <jan.beutel@uibk.ac.at>
To:	Tobias Buehler <buehlert@ethz.ch>
Subject:	[RNIT] Exam questions

A body, also in 7-bit U.S. ASCII text

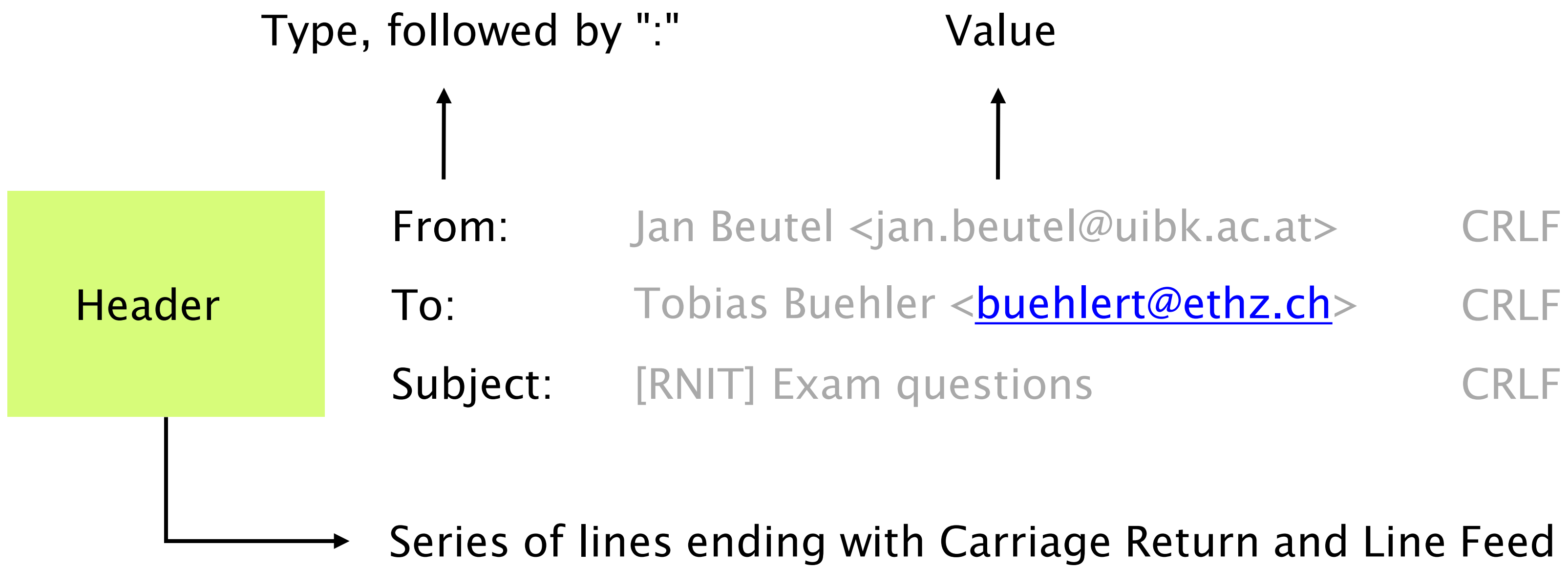
From:	Jan Beutel <jan.beutel@uibk.ac.at>
To:	Tobias Buehler <buehlert@ethz.ch>
Subject:	[RNIT] Exam questions

Body

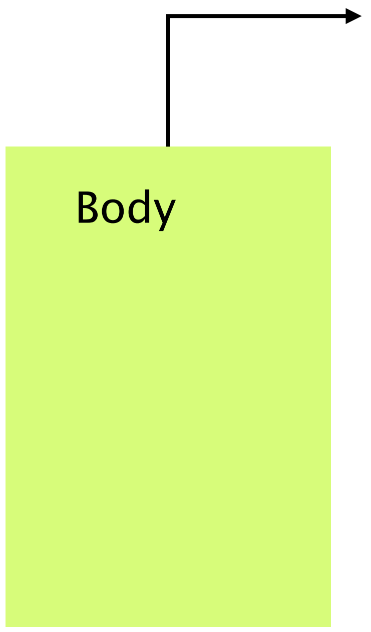
Hi Tobias,

Here are some interesting questions…

Best,
Jan



```
From:      Jan Beutel <jan.beutel@uibk.ac.at>       CRLF
To:        Tobias Buehler <buehlert@ethz.ch>        CRLF
Subject:   [RNIT] Exam questions                    CRLF
```

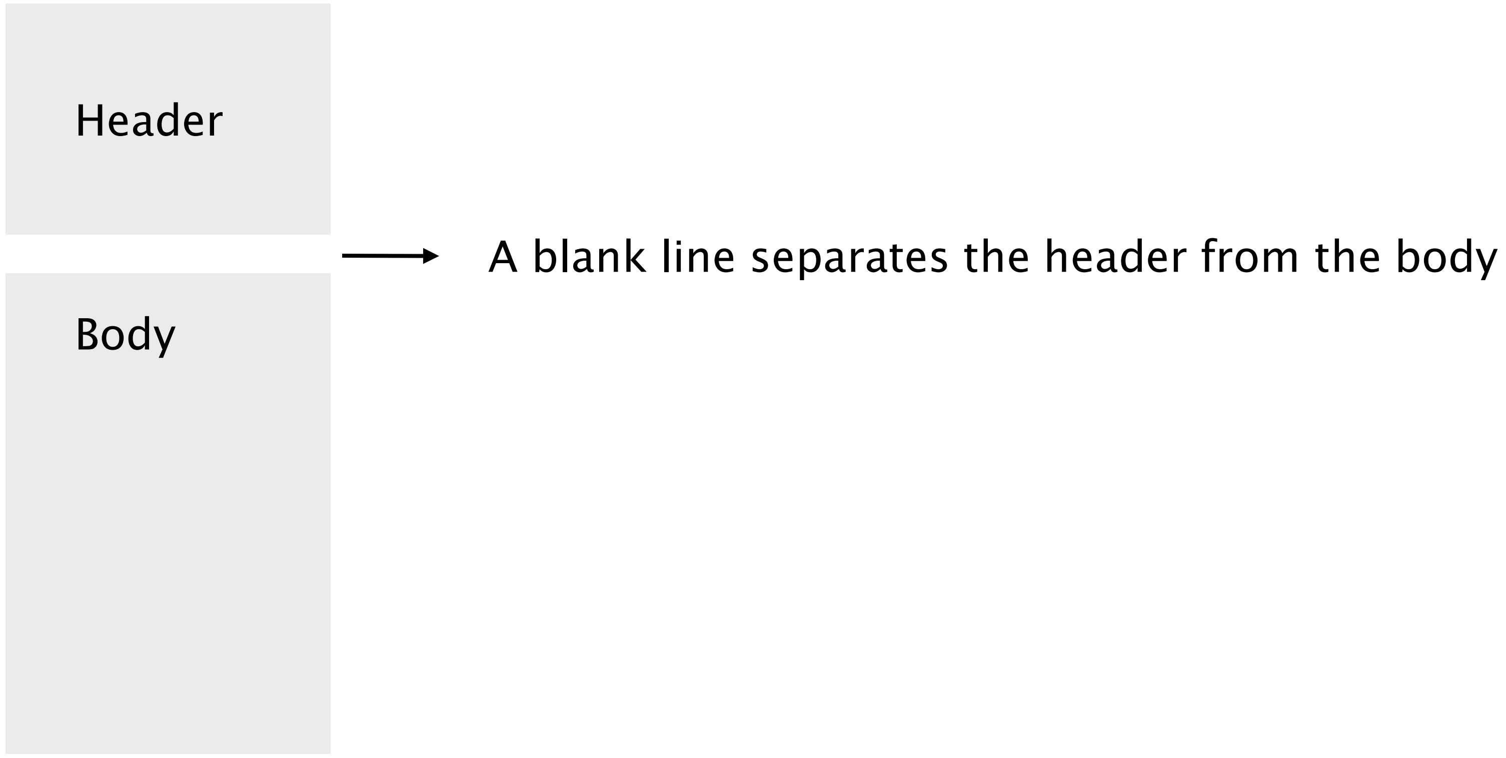


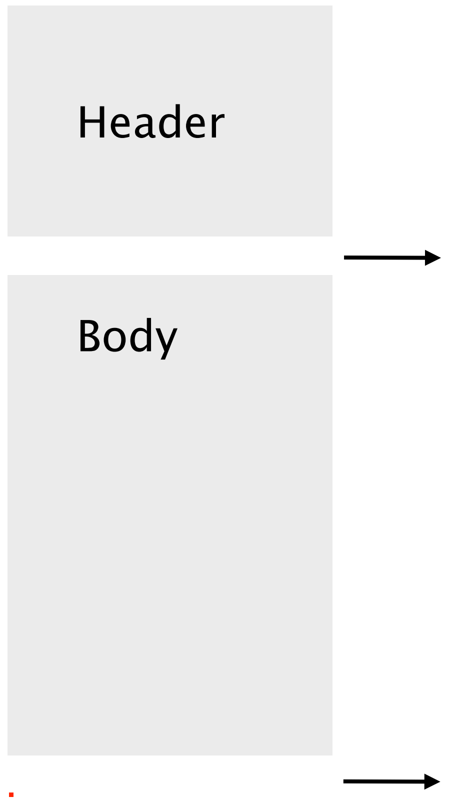
Series of lines with no structure/meaning

Hi Tobias,

Here are some interesting questions…

Best,
Jan


Header
Body
A blank line separates the header from the body



A blank line separates the header from the body

A dot (".") on a new line ends the body

Email relies on 7-bit U.S. ASCII…

How do you send non-English text? Binary files?

Solution

Multipurpose Internet Mail Extensions

commonly known as MIME, standardized in RFC 822

MIME defines

- additional headers for the email body
- a set of content types and subtypes
- base64 to encode binary data in ASCII

MIME defines

- additional headers for the email body

  MIME-Version: the version of MIME being used

  Content-Type: the type of data contained in the message

  Content-Transfer-Encoding: how the data is encoded

MIME defines

- additional headers for the email body
- a set of content types and subtypes

  e.g. image with subtypes gif or jpeg

  text with subtypes plain, html, and rich text

  application with subtypes postscript or msword

  multipart with subtypes mixed or alternative

The two most common types/subtypes for MIME are: *multipart/mixed* and *multipart/alternative*

Content-Type	indicates that the message contains
multipart/mixed	multiple independent parts e.g. plain text *and* a binary file
multipart/alternative	multiple representation of the same content e.g. plain text *and* HTML

MIME defines

- additional headers for the email body
- a set of content types and subtypes
- base64 to encode binary data in ASCII

What kind of delimiter do we use?

multipart/mixed

multiple independent parts

multipart/alternative

multiple representation of
the same content

Content-Type contains a parameter that specifies a string delimiter (chosen randomly by the client)

ensuring that the delimiter
does *not* appear in the email itself

```
From: Jan Beutel<jan.beutel@uibk.ac.at>
To: Tobias Buehler <buehlert@ethz.ch>
Subject: [RNIT] Final exam
MIME-Version: 1.0
Content-Type: multipart/related;
boundary="_004_cc163051808f425a9b67b778666b785eeeethzch_";
   type="multipart/alternative"

--_004_cc163051808f425a9b67b778666b785eeeethzch_
Content-Type: multipart/alternative;
   boundary="_000_cc163051808f425a9b67b778666b785eeeethzch_"

--_000_cc163051808f425a9b67b778666b785eeeethzch_
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit

Let's start the exam with ...

--_000_cc163051808f425a9b67b778666b785eeeethzch_
Content-Type: text/html; charset="utf-8"
Content-Transfer-Encoding: base64

PGh0bWwgeG1sbnM6dj0idX ...
```

MIME relies on Base64 as binary-to-text encoding scheme

Relies on 64 characters out of the 128 ASCII characters

the most common *and* printable ones, i.e. A-Z, a-z, 0-9, +, /

Divides the bytes to be encoded into sequences of 3 bytes

each group of 3 bytes is then encoded using 4 characters

Uses padding if the last sequence is partially filled

i.e. if the |sequence| to be encoded is not a multiple of 3

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	/

Binary input	0x14fb9c03d97e
8-bits	00010100 11111011 10011100
	00000011 11011001 01111110
6-bits	000101 001111 101110 011100
	000000 111101 100101 111110
Decimal	5 15 46 28 0 61 37 62
base64	F P u c A 9 l +

If the length of the input is not a multiple of three, Base64 uses "=" as padding character

Binary input	`0x14`
8-bits	`00010100`
6-bits	`000101 000000`
Decimal	`5 0`
base64	`F A = =`

```
Date: Thu, 14 Jan 2021 07:33:26 +0100
Message-ID: <202101140633.10E6XQ14046508@lora1.intra.uibk.ac.at>
From: =?ISO-8859-1?Q?RektorInnenteam?= <rektorenteam@uibk.ac.at>
Organization: =?ISO-8859-1?Q?Universitaet=20Innsbruck?=
To: <Jan.Beutel@uibk.ac.at>
Subject: =?ISO-8859-1?Q?Einladung=20zum=20virtuellen=20Neujahrsempfang=20?=
MIME-Version: 1.0
Content-Type: text/plain; charset="ISO-8859-1"
Content-Transfer-Encoding: 8bit
X-Priority: 3
Precedence: bulk
Errors-To: <>
X-Scanned-By: MIMEDefang 2.84 at uibk.ac.at
Return-Path: rektorenteam@uibk.ac.at
X-MS-Exchange-Organization-Network-Message-Id: ea5dd553-bcfb-491c-e577-08d8b856880f
X-MS-Exchange-Organization-AVStamp-Enterprise: 1.0
X-MS-Exchange-Organization-SCL: -1
X-MS-Exchange-Organization-AuthSource: xmbx6.uibk.ac.at
X-MS-Exchange-Organization-AuthAs: Anonymous
X-MS-Exchange-Transport-EndToEndLatency: 00:00:00.1920577
X-MS-Exchange-Processed-By-BccFoldering: 15.01.2044.012
```

```
To: "'institut-intern@informatik.uibk.ac.at'" <institut-intern@informatik.uibk.ac.at>
From: Philipp Gschwandtner <philipp.gschwandtner@uibk.ac.at>
Message-ID: <b550dc6a-afd8-b944-fe0a-63f3e369ed39@uibk.ac.at>
Subject: [Institut-intern] FZ HPC Lecture Series: Towards Exascale Computing in Europe
Date: Tue, 12 Jan 2021 15:48:58 +0100
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:78.0) Gecko/20100101
 Thunderbird/78.6.1

MIME-Version: 1.0
Content-Language: en-US
Content-Type: multipart/mixed;
   boundary="===============4407401938683644177=="

Errors-To: institut-intern-bounces@informatik.uibk.ac.at
Sender: Institut-intern <institut-intern-bounces@informatik.uibk.ac.at>
Return-Path: institut-intern-bounces@informatik.uibk.ac.at
X-MS-Exchange-Organization-Network-Message-Id: d6d420a1-5218-4e8e-7e46-08d8b70933a8
X-MS-Exchange-Organization-AVStamp-Enterprise: 1.0
X-MS-Exchange-Organization-SCL: -1
X-MS-Exchange-Organization-AuthSource: xmbx14.uibk.ac.at
X-MS-Exchange-Organization-AuthAs: Anonymous
X-MS-Exchange-Transport-EndToEndLatency: 00:00:00.3724855
X-MS-Exchange-Processed-By-BccFoldering: 15.01.2044.012
```

Content

Infrastructure/
Transmission

Retrieval

SMTP: Simple Mail
Transfer Protocol

Infrastructure

mail servers

An e-mail address is composed of two parts identifying the local mailbox and the domain

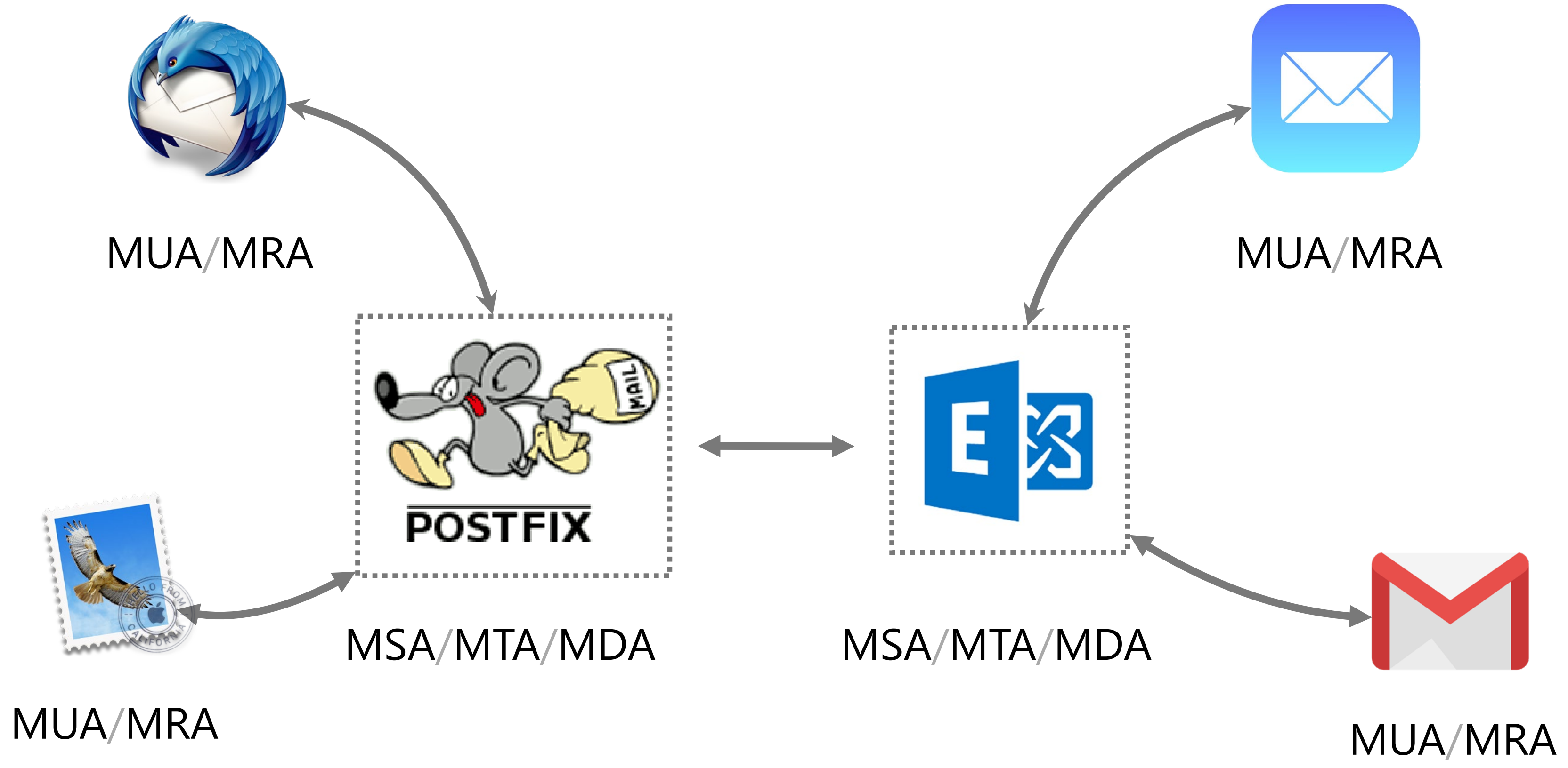
$$\begin{tikzcd} \text{jan.beutel} \arrow[d] & @ & \text{uibk.ac.at} \arrow[d] \\ \text{local mailbox} & & \text{domain name} \end{tikzcd}$$

actual mail server is identified using
a DNS query asking for MX records

We can divide the e-mail infrastructure into five functions

Mail	User	Agent	Use to read/write emails (mail client)
Mail	Submission	Agent	Process email and forward to local MTA
Mail	Transmission	Agent	Queues, receives, sends mail to other MTAs
Mail	Delivery	Agent	Deliver email to user mailbox
Mail	Retrieval	Agent	Fetches email from user mailbox

MSA/MTA/MDA and MRA/MUA are often packaged together leading to simpler workflows



Simple Mail Transfer Protocol (SMTP) is the current standard for transmitting e-mails

SMTP is a text-based, client-server protocol

client sends the e-mail, server receives it

SMTP uses reliable data transfer

built on top of TCP (port 25 and 465 for SSL/TLS)

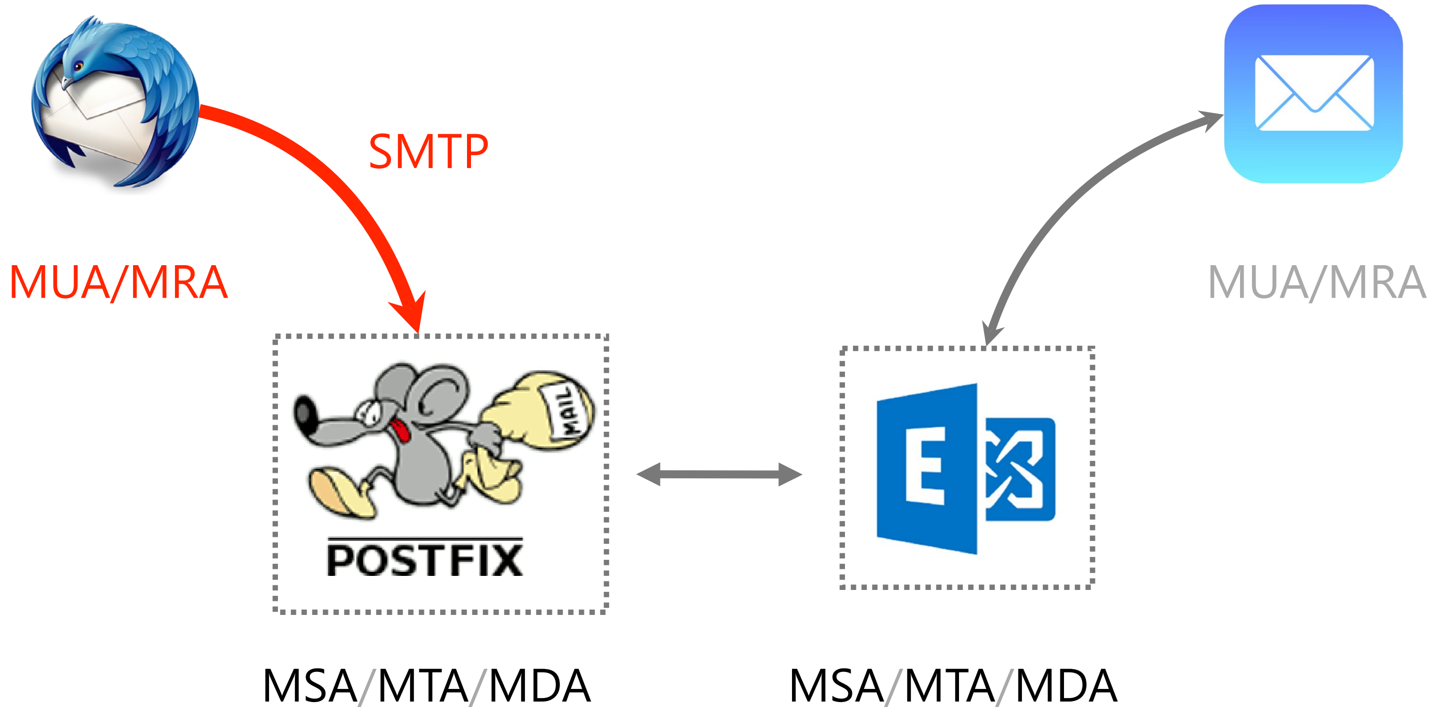
SMTP is a push-based protocol

sender pushes the file to the receiving server

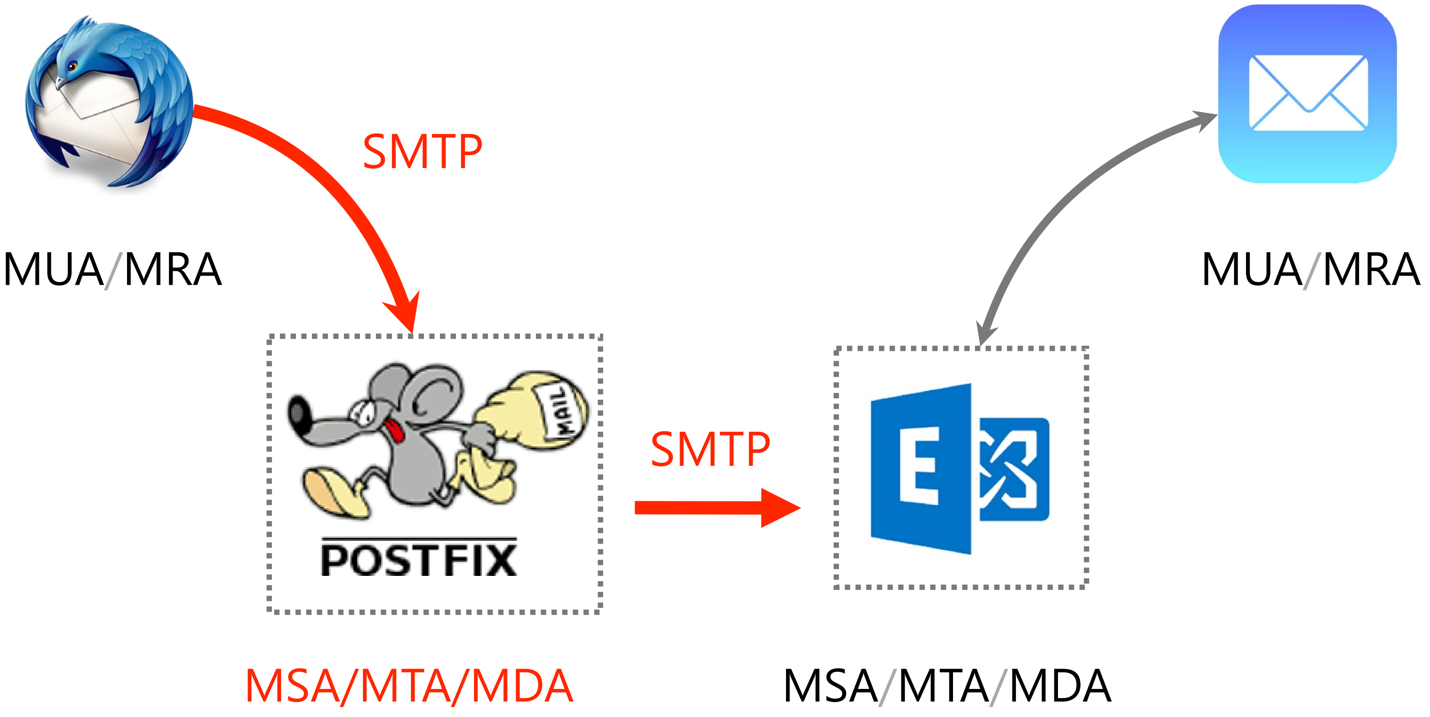
	SMTP 3 digit response code			comment
Status	2XX	success	220	Service ready
			250	Requested mail action completed
	3XX	input needed	354	Start mail input
	4XX	transient error	421	Service not available
			450	Mailbox unavailable
			452	Insufficient space
	5XX	permanent error	500	Syntax error
			502	Unknown command
			503	Bad sequence

```
server —— 220 hamburger.edu
          EHLO crepes.fr
          250  Hello crepes.fr, pleased to meet you
client —— MAIL FROM: <alice@crepes.fr>
          250 alice@crepes.fr... Sender ok
          RCPT TO: <bob@hamburger.edu>
          250 bob@hamburger.edu ... Recipient ok
          DATA
          354 Enter mail, end with "." on a line by
          itself
          Do you like ketchup?
          How about pickles?
          .
          250 Message accepted for delivery
          QUIT
          221 hamburger.edu closing connection
```

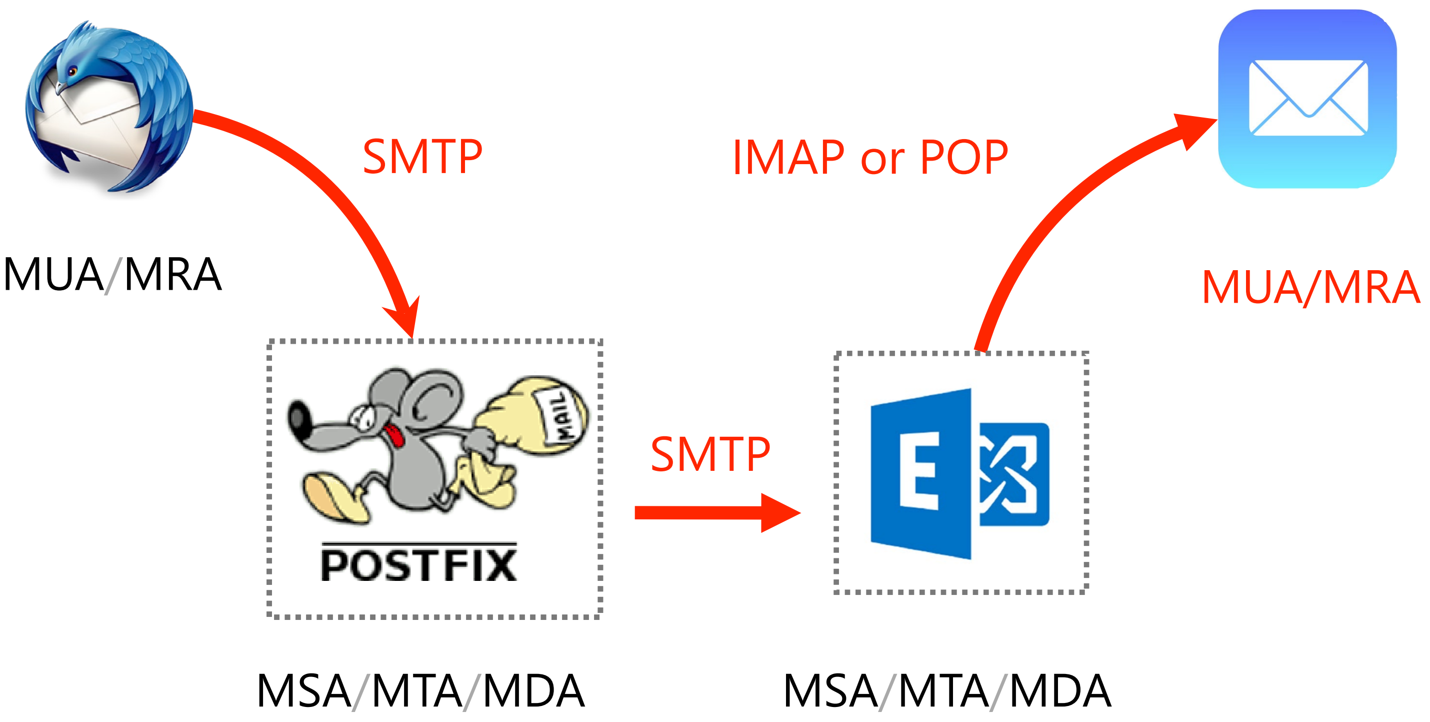
The sender MUA uses SMTP to transmit the e-mail to a local MTA (e.g., mail.uibk.ac.at, gmail.com, hotmail.com)



The local MTA then looks up the MTA of the recipient domain (DNS MX) and transmits the e-mail further



Once the e-mail is stored at the recipient domain, IMAP or POP is used to retrieve it by the recipient MUA



E-mails typically go through at least 2 SMTP servers, but often way more

sending and receiving sides

Each SMTP server/MTA hop adds its identity to the e-mail header by prepending a "Received" entry

```
8  Received: from edge20.ethz.ch (82.130.99.26) by CAS10.d.ethz.ch
    (172.31.38.210) with Microsoft SMTP Server (TLS) id 14.3.361.1; Fri, 23 Feb
    2018 01:48:56 +0100
7  Received: from phil4.ethz.ch (129.132.183.133) by edge20.ethz.ch
    (82.130.99.26) with Microsoft SMTP Server id 14.3.361.1; Fri, 23 Feb 2018
    01:48:57 +0100
6  Received: from outprodmail02.cc.columbia.edu ([128.59.72.51])          by phil4.ethz.ch
    with esmtps (TLSv1:AES256-SHA:256)          (Exim 4.69) (envelope-from
    <ethan@ee.columbia.edu>)        id 1ep1Xg-0002s3-FH  for lvanbever@ethz.ch; Fri, 23
    Feb 2018 01:48:55 +0100
5  Received: from hazelnut (hazelnut.cc.columbia.edu [128.59.213.250])   by
    outprodmail02.cc.columbia.edu (8.14.4/8.14.4) with ESMTP id w1N0iAu4026008
                 for <lvanbever@ethz.ch>; Thu, 22 Feb 2018 19:48:51 -0500
4  Received: from hazelnut (localhost.localdomain [127.0.0.1])             by hazelnut
    (Postfix) with ESMTP id 421126D   for <lvanbever@ethz.ch>; Thu, 22 Feb 2018
    19:48:52 -0500 (EST)
3  Received: from sendprodmail01.cc.columbia.edu (sendprodmail01.cc.columbia.edu
    [128.59.72.13])          by hazelnut (Postfix) with ESMTP id 211526D   for
    <lvanbever@ethz.ch>; Thu, 22 Feb 2018 19:48:52 -0500 (EST)
2  Received: from mail-pl0-f43.google.com (mail-pl0-f43.google.com
    [209.85.160.43])          (user=ebk2141 mech=PLAIN bits=0)             by
    sendprodmail01.cc.columbia.edu (8.14.4/8.14.4) with ESMTP id w1N0mnlx052337
                 (version=TLSv1/SSLv3 cipher=AES128-GCM-SHA256 bits=128 verify=NOT)         for
    <lvanbever@ethz.ch>; Thu, 22 Feb 2018 19:48:50 -0500
1  Received: by mail-pl0-f43.google.com with SMTP id u13so3927207plq.1        for
    <lvanbever@ethz.ch>; Thu, 22 Feb 2018 16:48:50 -0800 (PST)
```

E-mails typically go through at least 2 SMTP servers, but often way more

Separate SMTP servers for separate functions

SPAM filtering, virus scanning, data leak prevention, etc.

Separate SMTP servers that redirect messages

e.g. from beutel@tik.ee.ethz.ch to janbeutel@ethz.ch

Separate SMTP servers to handle mailing-list

mail is delivered to the list server and then expanded

Try it out yourself!

SMTP-MTA

plaintext (!),
hard to find

```
telnet server_name 25
```

SMTP-MSA

rely on TLS
encryption

authentication
required

```
openssl s_client -starttls smtp
   -connect mail.ethz.ch:587
   -crlf -ign_eof (*)
```

```
perl -MMIME::Base64 -e 'print encode_base64("username");'
perl -MMIME::Base64 -e 'print encode_base64("password");'
```

(*) https://www.ndchost.com/wiki/mail/test-smtp-auth-telnet

As with most of the key Internet protocols, security is an afterthought

SMTP Headers

MAIL FROM:	no checks are done to verify that the sending MTA is authorized to send e-mails on behalf of that address

Email content (DATA)

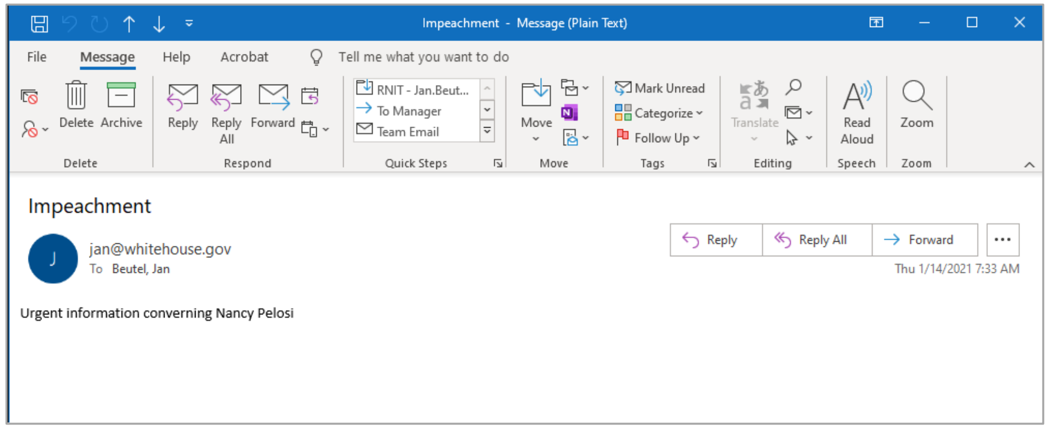
From:	no checks are done to verify that the sending system is authorized to send e-mail on behalf of that address
Reply-to:	ditto

In short, *none* of the addresses in an email are typically reliable

Let's spoof some e-mails!

(don't try this at home)

Emails that exist but don't really exist….

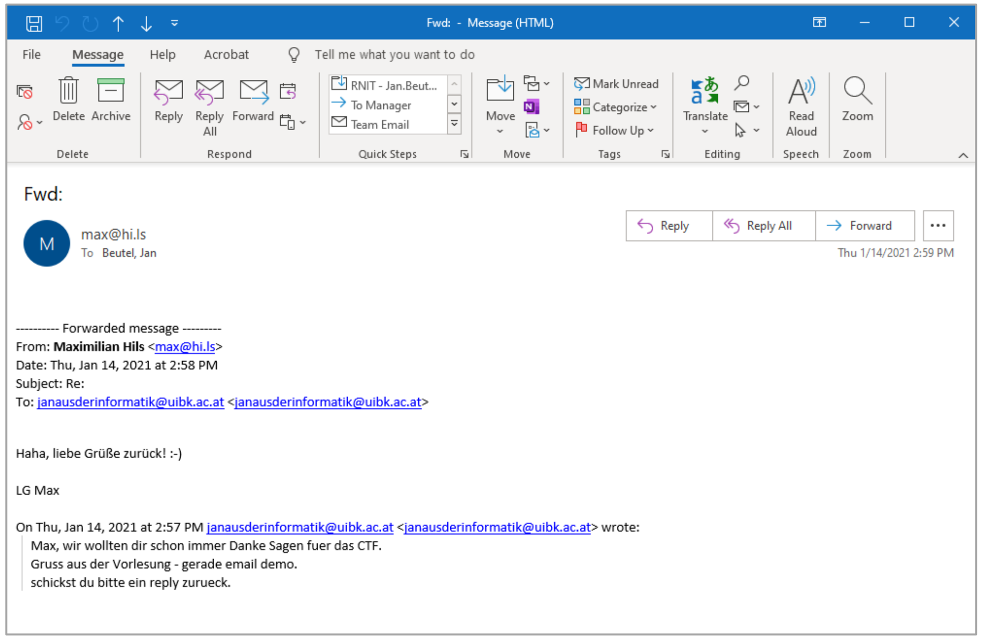


Impeachment

jan@whitehouse.gov
To Beutel, Jan

Thu 1/14/2021 7:33 AM

Urgent information converning Nancy Pelosi



Fwd:

max@hi.ls
To Beutel, Jan

Thu 1/14/2021 2:59 PM

---------- Forwarded message ---------
From: **Maximilian Hils** <max@hi.ls>
Date: Thu, Jan 14, 2021 at 2:58 PM
Subject: Re:
To: janausderinformatik@uibk.ac.at <janausderinformatik@uibk.ac.at>

Haha, liebe Grüße zurück! :-)

LG Max

On Thu, Jan 14, 2021 at 2:57 PM janausderinformatik@uibk.ac.at <janausderinformatik@uibk.ac.at> wrote:

> Max, wir wollten dir schon immer Danke Sagen fuer das CTF.
> Gruss aus der Vorlesung - gerade email demo.
> schickst du bitte ein reply zurueck.

And, as usual, multiple countermeasures have been proposed with various level of deployment success

Example*	Sender Policy Framework (SPF)

Enables a domain to explicitly authorize
a set of hosts that are allowed to send emails
using their domain names in "MAIL FROM".

How? using a DNS TXT resource record

look for "v=spf1" in the results of "dig TXT google.com"

* if you are interested, also check out Sender ID, DKIM, and DMARC

Content

Infrastructure/
Transmission

Retrieval

POP: Post Office Protocol

IMAP: Internet Message
Access Protocol

Content

Infrastructure/
Transmission

Retrieval

POP: Post Office Protocol

IMAP: Internet Message
Access Protocol

POP is a simple protocol which was designed to support users with intermittent network connectivity

POP enables e-mail users to

▪ retrieve e-mails locally	when connected
▪ view/manipulate e-mails	when disconnected

and that's pretty much it...

Example

POP server

client

```
+OK POP3 server ready
user bob
+OK
pass hungry
+OK user successfully logged on

list
1 498
2 912
.
retr 1
<message 1 contents>
.
dele 1
retr 2
<message 1 contents>
.
dele 2
quit
+OK POP3 server signing off
```

Authorization phase

Clients declares username

password

Server answers +OK/-ERR

```
+OK POP3 server ready
user bob
+OK
pass hungry
+OK user successfully logged on

list
1 498
2 912
.
retr 1
<message 1 contents>
.
dele 1
retr 2
<message 1 contents>
.
dele 2
quit
+OK POP3 server signing off
```

Transaction phase

`list`	get message numbers
`retr`	retrieve message X
`dele`	delete message X
`quit`	exit session

```
+OK POP3 server ready
user bob
+OK
pass hungry
+OK user successfully logged on
```

```
list
1 498
2 912
.
retr 1
<message 1 contents>
.
dele 1
retr 2
<message 1 contents>
.
dele 2
quit
+OK POP3 server signing off
```

POP is heavily limited. Among others, it does not go well with multiple clients or always-on connectivity

Cannot deal with multiple mailboxes

designed to put incoming emails in one folder

Not designed to keep messages on the server

designed to download messages to the client

Poor handling of multiple-client access

while many (most?) users have now multiple devices

Content

Infrastructure/
Transmission

Retrieval

POP: Post Office Protocol

IMAP: Internet Message
Access Protocol

Unlike POP, Internet Message Access Protocol (IMAP) was designed with multiple clients in mind

Support multiple mailboxes and searches on the server

client can create, rename, move mailboxes & search on server

Access to individual MIME parts and partial fetch

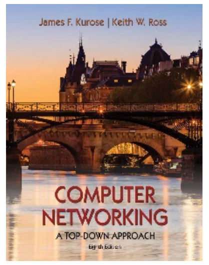
client can download only the text content of an e-mail

Support multiple clients connected to one mailbox

server keep state about each message (e.g. read, replied to)

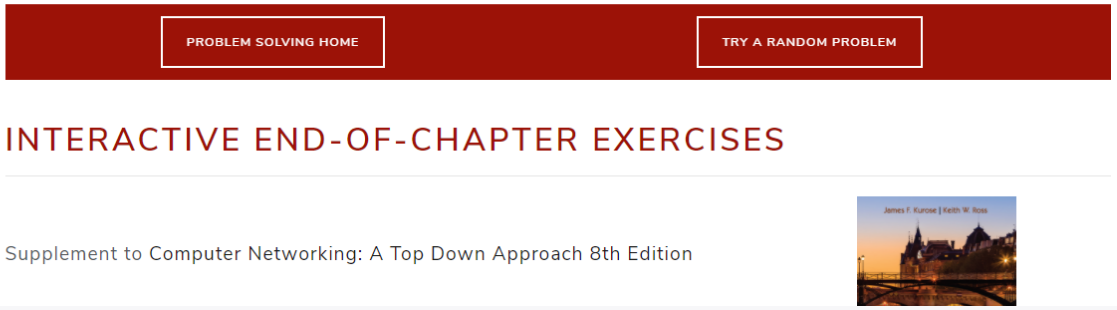
Reading: Book Kurose & Ross

Class textbook:
Computer Networking: A Top-Down Approach (8th *ed.)*
J.F. Kurose, K.W. Ross
Pearson, 2020
http://gaia.cs.umass.edu/kurose_ross



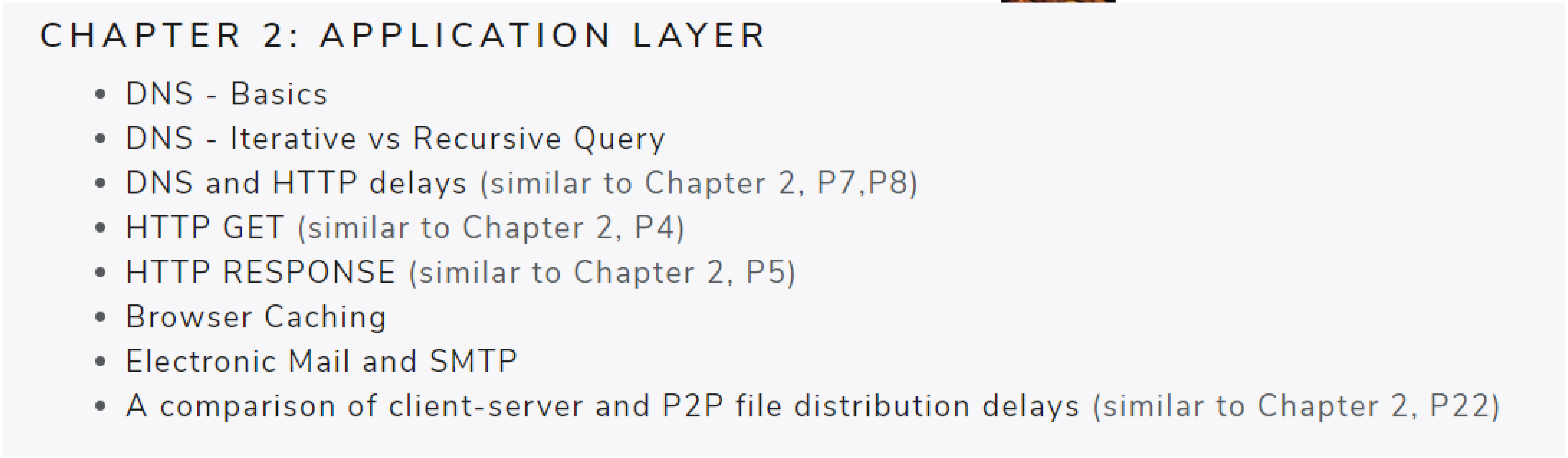
- Week 11
  - 2.5 (DNS. The Internet's Directory Service)
  - 2.2 (The Web and HTTP)

Check Your Knowledge



INTERACTIVE END-OF-CHAPTER EXERCISES

Supplement to Computer Networking: A Top Down Approach 8th Edition



CHAPTER 2: APPLICATION LAYER

- DNS - Basics
- DNS - Iterative vs Recursive Query
- DNS and HTTP delays (similar to Chapter 2, P7,P8)
- HTTP GET (similar to Chapter 2, P4)
- HTTP RESPONSE (similar to Chapter 2, P5)
- Browser Caching
- Electronic Mail and SMTP
- A comparison of client-server and P2P file distribution delays (similar to Chapter 2, P22)

http://gaia.cs.umass.edu/kurose_ross/interactive/