Communication Networks

Prof. Laurent Vanbever

Communication Networks

Spring 2021

Laurent Vanbever
nsg.ee.ethz.ch

ETH Zürich (D-ITET)
May 31 2021

Materials inspired from Scott Shenker and Jennifer Rexford

Last Monday on
Communication Networks

DNS

Web

google.ch ⟷ 172.217.16.131

http://www.google.ch
(the beginning)

Internet has one **global system** for

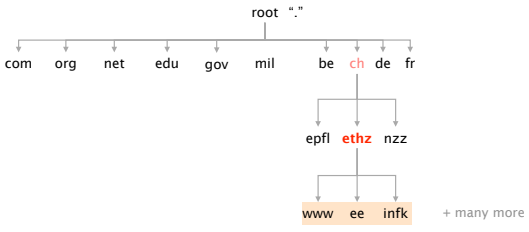
- **addressing** hosts — IP
  by design
- **naming** hosts — DNS
  by "accident", an afterthought

To scale,
DNS adopt three intertwined hierarchies

naming structure	hierarchy of addresses https://www.ee.ethz.ch/de/departement/
management	hierarchy of authority over names
infrastructure	hierarchy of DNS servers

naming structure — hierarchy of addresses
https://www.ee.ethz.ch/de/departement/

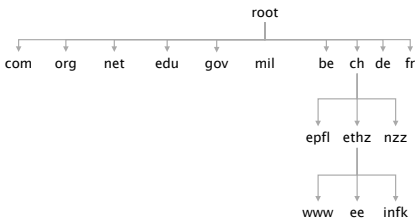
A name, *e.g.* ee.ethz.ch, represents
a leaf-to-root path in the hierarchy

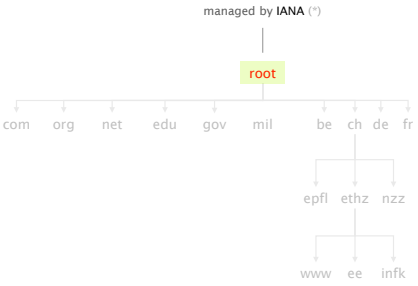


management

hierarchy of authority
over names

The DNS system is
hierarchically administered





(*) see http://www.iana.org/domains/root/db

infrastructure

hierarchy of DNS servers

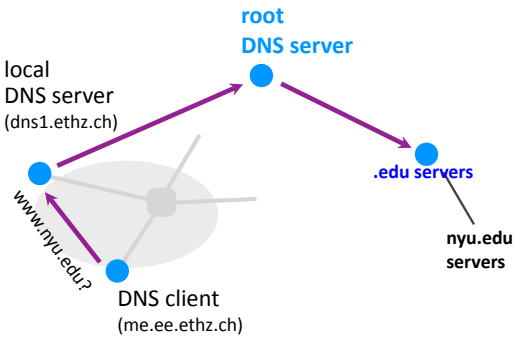
13 root servers (managed professionally)
serve as root

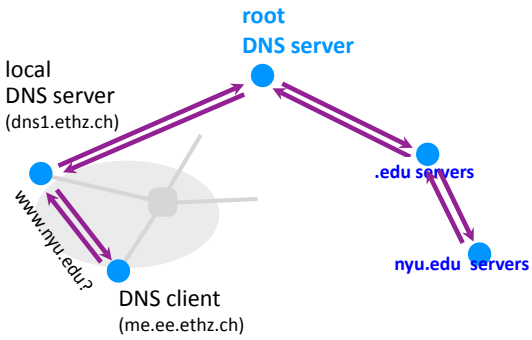
a.	root-servers.net	VeriSign, Inc.
b.	root-servers.net	University of Southern California
c.	root-servers.net	Cogent Communications
d.	root-servers.net	University of Maryland
e.	root-servers.net	NASA
f.	root-servers.net	Internet Systems Consortium
g.	root-servers.net	US Department of Defense
h.	root-servers.net	US Army
i.	root-servers.net	Netnod
j.	root-servers.net	VeriSign, Inc.
k.	root-servers.net	RIPE NCC
l.	root-servers.net	ICANN
m.	root-servers.net	WIDE Project

A DNS server stores Resource Records
composed of a (name, value, type, TTL)

Records	Name	Value
A	hostname	IP address
NS	domain	DNS server name
MX	domain	Mail server name
CNAME	alias	canonical name
PTR	IP address	corresponding hostname

DNS resolution can either be
recursive or iterative





Today on
Communication Networks

Web

http://www.google.ch
(the end)

Email

MX, SMTP, POP, IMAP

Web

http://www.google.ch
(the end)

Email

Web

Email

MX, SMTP, POP, IMAP

We'll study e-mail from three different perspectives

Content

Format: Header/Content
Encoding: MIME

Infrastructure/
Transmission

SMTP: Simple Mail
Transfer Protocol
Infrastructure
mail servers

Retrieval

POP: Post Office Protocol
IMAP: Internet Message
Access Protocol

Content

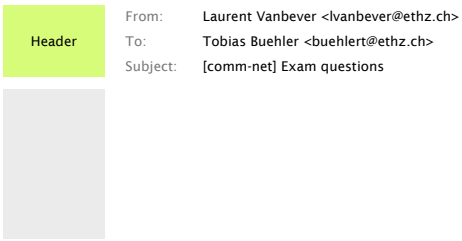
Format: Header/Content
Encoding: MIME

Infrastructure/
Transmission

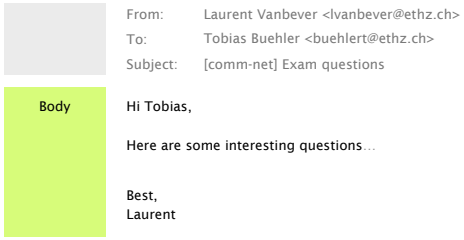
Retrieval

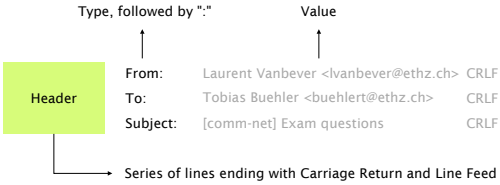
An e-mail is composed of two parts

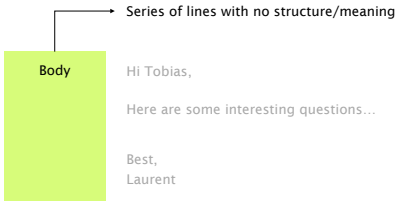
A header, in 7-bit U.S. ASCII text

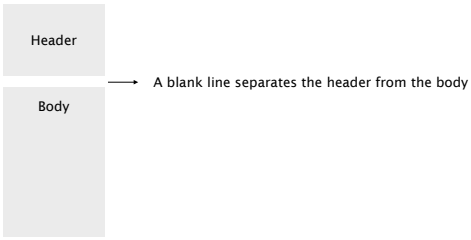


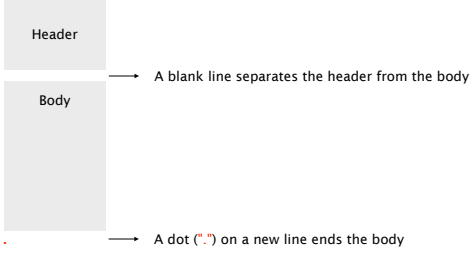
A body, also in 7-bit U.S. ASCII text











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

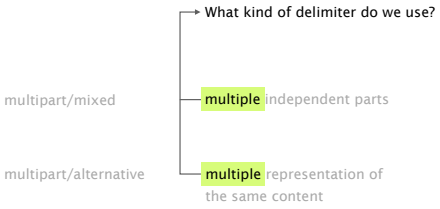
MIME defines

- 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



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: Laurent Vanbever <lvanbever@ethz.ch>
To: Tobias Buehler <buehlert@ethz.ch>
Subject: [comm-net] 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 defines

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

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

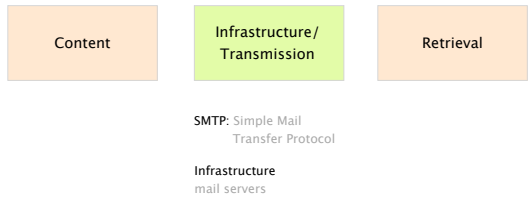
Binary input	0x14fb9c03d97e
8-bits	00010100 11111011 10011100 00000011 11011001 01111110
6-bits	000101 001111 101110 011100 000000 111101 100101 111110
Decimal	5 15 46 28 0 61 37 62
base64	F P u c A 9 l +

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	/

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 = =

```
From: Laurent Vanbever <lvanbever@ethz.ch>
To: Tobias Buehler <buehlert@ethz.ch>
Subject: [comm-net] Final exam
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: multipart/mixed;
        boundary="123boundary"

This is a multipart message in MIME format.

--123boundary
Content-Type: text/plain

Hi Tobias, Please find the exam enclosed. Laurent

--123boundary
Content-Type: application/pdf;
Content-Disposition: attachment;
        filename="exam_2020.pdf"

base64 encoded data .....
.........................
......base64 encoded data
```



SMTP: Simple Mail Transfer Protocol

Infrastructure
mail servers

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

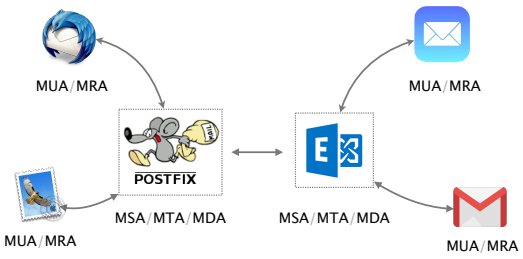
lvanbever @ ethz.ch
↓ local mailbox
↓ domain name

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

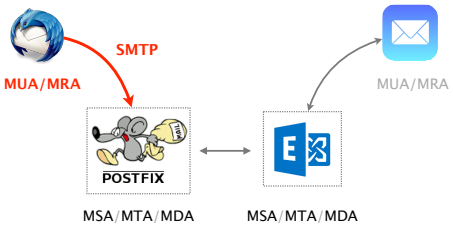
SMTP uses reliable data transfer
built on top of TCP (port 25 and 465 for SSL/TLS)

SMTP is a push-based protocol
sender pushes the file to the receiving server

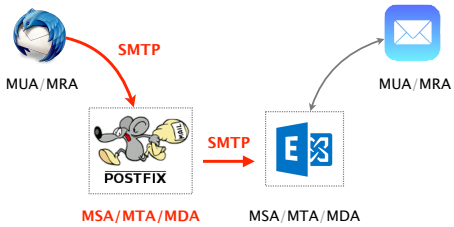
	SMTP 3 digit response code			comment
Status	2XX	success	220	Service ready
			250	Requested mail action completed
	3XX	input needed	354	Start mail input
	4XX	transient error	421	Service not available
			450	Mailbox unavailable
			452	Insufficient space
	5XX	permanent error	500	Syntax error
			502	Unknown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

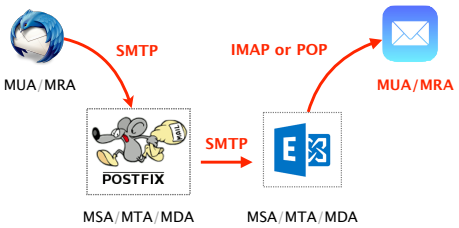
The sender MUA uses SMTP to transmit the e-mail to a local MTA (e.g. mail.ethz.ch, gmail.com, hotmail.com)



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 lvanbever@tik.ee.ethz.ch to lvanbever@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

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

authentication required

```
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

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 ⟵ +OK POP3 server ready
user bob
+OK
client ⟵ 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

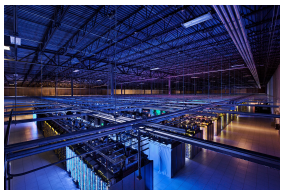
Access to individual MIME parts and partial fetch
client can download only the text content of an e-mail

Support multiple clients connected to one mailbox
server keep state about each message (e.g. read, replied to)

Knowledge

Understand how the Internet works and why

from your network plug…



…to the largest data-centers out there

List any

technologies, principles, applications…

used after typing in:

> www.google.ch

and pressing enter in your browser

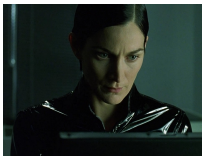
Insight

Key concepts and problems in Networking

Naming Layering Routing Reliability Sharing

Skill

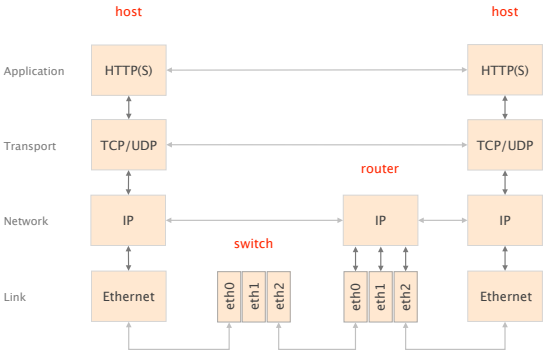
Build, operate and configure networks



Trinity using a port scanner (nmap) in Matrix Reloaded™

The Internet is organized as layers, providing a set of services

	layer	service provided
L5	Application	network access
L4	Transport	end-to-end delivery (reliable or not)
L3	Network	global best-effort delivery
L2	Link	local best-effort delivery
L1	Physical	physical transfer of bits



We started with the fundamentals of routing and reliable transport

	Application	network access
L4	Transport	end-to-end delivery (reliable or not)
L3	Network	global best-effort delivery
	Link	local best-effort delivery
	Physical	physical transfer of bits

We saw three ways to compute valid routing state

	Intuition	Example
#1	Use tree-like topologies	Spanning-tree
#2	Rely on a global network view	Link-State SDN
#3	Rely on distributed computation	Distance-Vector BGP

We saw **how to design a reliable transport protocol**
and you implemented one yourself

goals

correctness	ensure data is delivered, in order, and untouched
timeliness	minimize time until data is transferred
efficiency	optimal use of bandwidth
fairness	play well with other concurrent communications

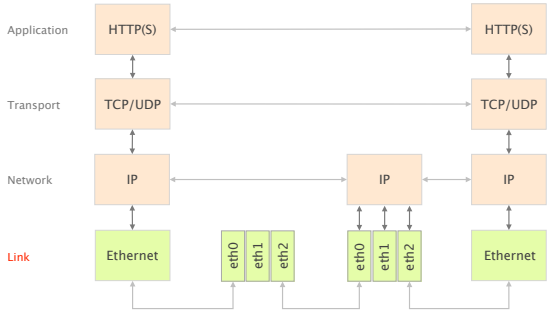
In each case, we explored the rationale behind each protocol and why they came to be

Why did the protocols end up looking like this?
minimum set of features required

What tradeoffs do they achieve?
efficiency, cost,…

When is one design more adapted than another?
packet switching *vs* circuit switching, DV *vs* LS,…

We then climbed up the layers,
starting from layer 2



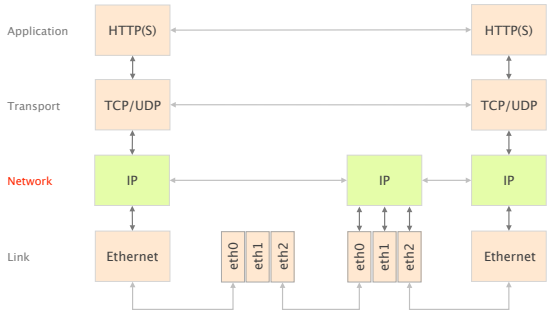
Communication Networks

Part 2: The Link Layer

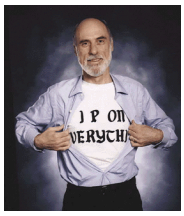
ETH

#1	What is a link?
#2	How do we identify link adapters?
#3	How do we share a network medium?
#4	What is Ethernet?
#5	How do we interconnect segments at the link layer?

We then spent **multiple weeks on layer 3**



Internet Protocol and Forwarding



source: Boardwatch Magazine

1. **IP addresses**
   use, structure, allocation
2. **IP forwarding**
   longest prefix match rule
3. **IP header**
   IPv4 and IPv6, wire format

We also talked about **IPv6**

Internet routing

from here to there, and back

1. **Intra-domain routing**
   Link-state protocols
   Distance-vector protocols
2. **Inter-domain routing**
   Path-vector protocols

Internet routing comes into two flavors:
intra- and *inter-domain* routing

inter-domain routing

Find paths between networks

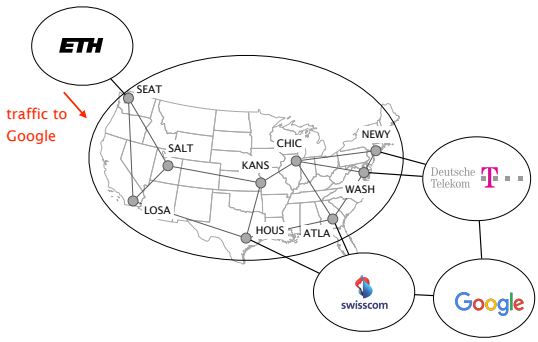
intra-domain routing

Find paths within a network

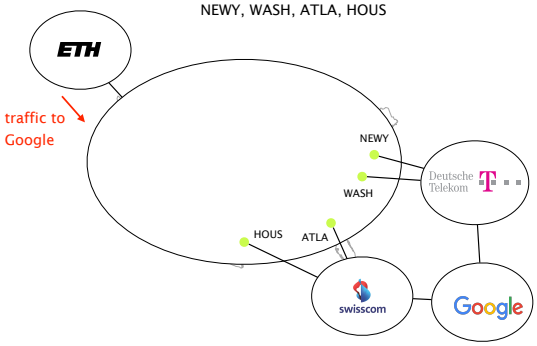
inter-domain routing

Find paths between networks

intra-domain routing



Google can be reached via
NEWY, WASH, ATLA, HOUS

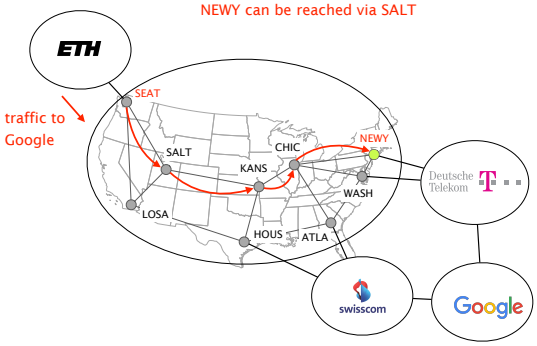


inter-domain routing

intra-domain routing

Find paths within a network

NEWY can be reached via SALT



Border Gateway Protocol
policies and more

1. BGP Policies
   Follow the money
2. Protocol
   How does it work?
3. Problems
   security, performance, …

Business relationships conditions
route selection

For a destination *p*, prefer routes coming from

- customers over
- peers over
- providers

route type

Business relationships conditions
route exportation

		send to		
		customer	peer	provider
from	customer			
	peer			
	provider			

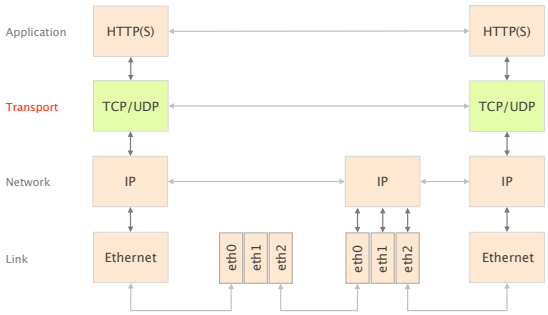
Routes coming from customers
are propagated to everyone else

		send to		
		customer	peer	provider
from	customer	✓	✓	✓
	peer			
	provider			

Routes coming from peers and providers
are only propagated to customers

		send to		
		customer	peer	provider
from	customer	✓	✓	✓
	peer	✓	-	-
	provider	✓	-	-

4 = 3+1



We looked at the requirements and implementation
of transport protocols (UDP/TCP)

Data delivering, to the *correct* application
- IP just points towards next protocol
- *Transport needs to demultiplex incoming data (ports)*

Files or bytestreams abstractions for the applications
- Network deals with packets
- *Transport layer needs to translate between them*

Reliable transfer (if needed)

Not overloading the receiver

Not overloading the network

We then looked at Congestion Control
and how it solves three fundamental problems

#1	bandwidth estimation	How to adjust the bandwidth of a single flow to the bottleneck bandwidth? could be 1 Mbps or 1 Gbps…
#2	bandwidth adaptation	How to adjust the bandwidth of a single flow to variation of the bottleneck bandwidth?
#3	fairness	How to share bandwidth "fairly" among flows, without overloading the network

… by combining two key mechanisms

detecting congestion

reacting to congestion

We then looked at
what's running on top of all this …

DNS

Web

E-mail

Your final grade

Exam

Projects

70%
written, open book

30%
20% routing
10% transport

Your final grade

Exam

Projects

70%
written, open book

30%
20% routing
10% transport

The exam will be open book, most of the questions will be open-ended, with some multiple choices

verify your understanding of the material

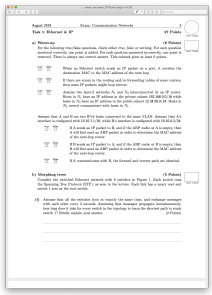
Make sure you can do *all* the exercises, especially the ones in previous exams

Millesime 2017

Millesime 2016

Millesime 2018

Millesime 2019



https://comm-net.ethz.ch/#tab-exam

Don't forget the assignments, they matter

No programming question — no Python at the exam

but we could ask you to describe a procedure in English

What would you change in your solution to achieve *X*?

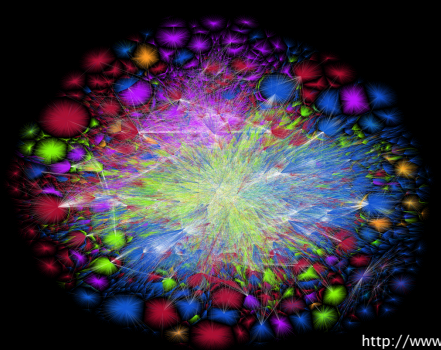
No configuration question — no Quagga at the exam

but we could ask you to describe a configuration in English

How would you enforce policy *X*?

We'll organize another remote Q&A session closer to the exam (details to follow)

Now you (better) understand this!



http://www.opte.org

Master-level lecture, every Fall semester

Advanced Topics in Communication Networks

Topics
(not exhaustive)

- Tunneling
- Hierarchical routing
- Traffic Engineering
- Virtual Private Networks
- Quality of Service/Scheduling
- IP Multicast
- Fast Convergence
- Network virtualization
- Network programmability
- Network measurements

+ labs & a project

if you liked the routing project, you will like this lecture as well

https://adv-net.ethz.ch/

Master-level lecture, every Spring semester

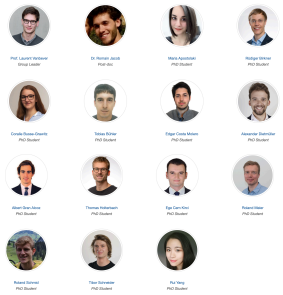
Seminar in Communication Networks

- Understand recent research result
- Read, present, and critique research papers
- Identify new research opportunities

https://seminar-net.ethz.ch/

Consider doing one of your theses with our group!

bachelor, semester or master



https://nsg.ee.ethz.ch/theses/

Communication Networks

Spring 2021

Laurent Vanbever
nsg.ee.ethz.ch

ETH Zürich (D-ITET)
May 31 2021