Communication Networks

Prof. Laurent Vanbever

Communication Networks

Spring 2021

Laurent Vanbever

nsg.ee.ethz.ch

ETH Zürich (D-ITET)

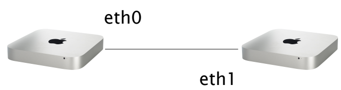
March 29 2021

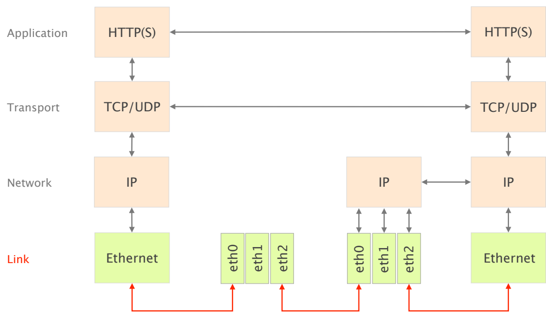
Materials inspired from Scott Shenker & Jennifer Rexford

Last week on

Communication Networks

How do local computers communicate?





Communication Networks

Part 2: The Link Layer

#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?

Communication Networks

Part 2: The Link Layer

#1 What is a link?

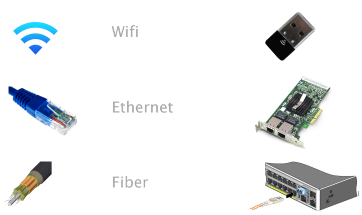
How do we identify link adapters?

How do we share a network medium?

What is Ethernet?

How do we interconnect segments at the link layer?

Link Communication medium and Network adapter



Communication Networks
Part 2: The Link Layer

What is a link?

#2 How do we identify link adapters?

How do we share a network medium?

What is Ethernet?

How do we interconnect segments at the link layer?

MAC addresses…

identify the sender & receiver adapters
used within a link

are uniquely assigned
hard-coded into the adapter when built

use a flat space of 48 bits
allocated hierarchically

You need to solve two problems
when you bootstrap an adapter

Who am I?
MAC-to-IP binding

How do I acquire an IP address?

Who are you?
IP-to-MAC binding

Given an IP address reachable on a link,
How do I find out what MAC to use?

Who am I?
MAC-to-IP binding

How do I acquire an IP address?
Dynamic Host Configuration Protocol

Who are you?
IP-to-MAC binding

Given an IP address reachable on a link,
How do I find out what MAC to use?
Address Resolution Protocol

This week on
Communication Networks

Link Layer
The end
see last weeks' slides

Network Layer
The beginning

Link Layer
The end
see last weeks' slides

Network Layer
The beginning

The Local Area Networks we have considered so far
define single broadcast domains

If one user broadcast a frame,
every other user receives it

As the network scales,
network operators like to segment their LANs

Why?

Improves security
smaller attack surface (visibility & injection)

Improves performance
limit the overhead of broadcast traffic (e.g. ARP)

Improves logistics
separates traffic by role (e.g. staff, students, visitors)

Organizational changes are too frequent to segment networks purely physically—rewiring is a major pain

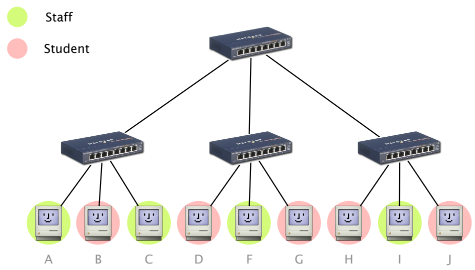
What about doing this in software though?

Enters "Virtual Local Area Networks" (VLANs)

Definition

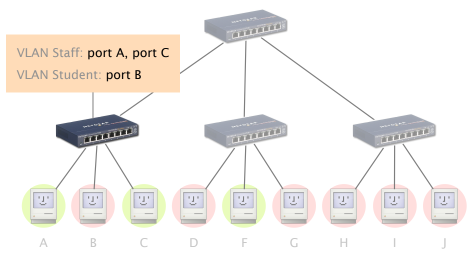
A VLAN logically identifies
a set of ports attached to
one (or more) Ethernet switches,
forming one broadcast domain

A VLAN identifies a set of ports attached to
one or more Ethernet switches

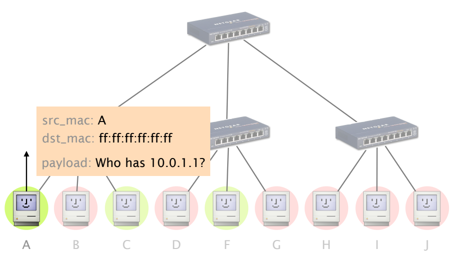


Switches need configuration tables telling them
which VLANs are accessible via which interfaces

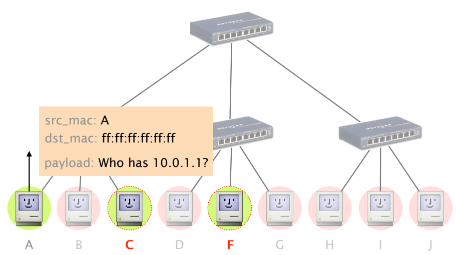
Switches need configuration tables telling them
which VLANs are accessible via which interfaces

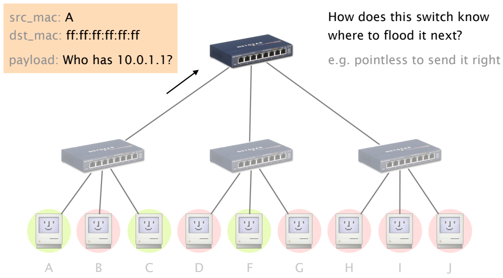


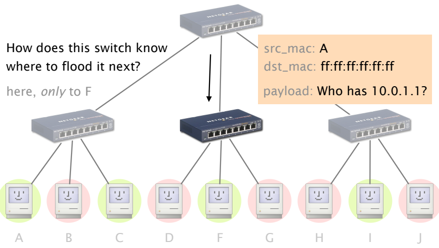
Consider that A sends a broadcast frame
say, an ARP request



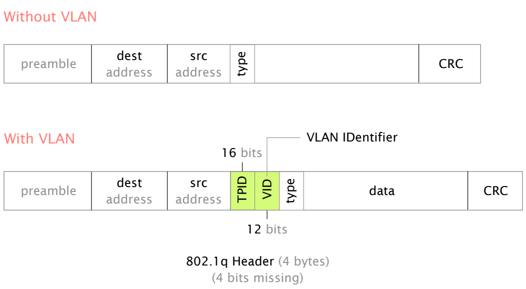
That frame should be received by all staff members:
i.e. C and F, and *only* them



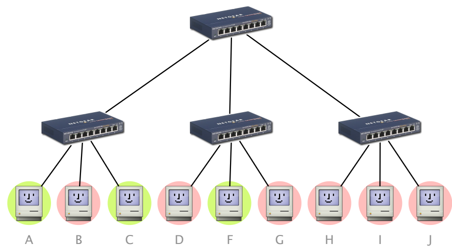




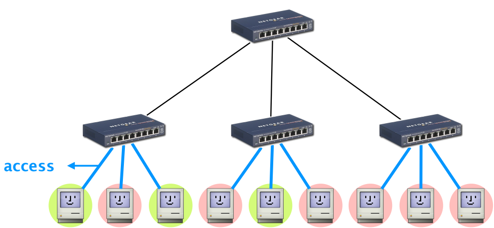
To identify VLAN, switches add new header
when forwarding traffic to another switch



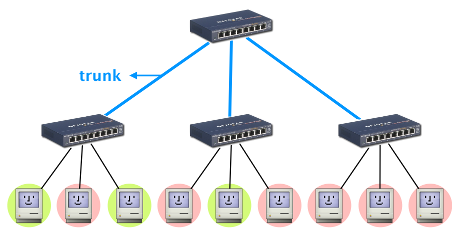
With VLANs, Ethernet links are divided in two sets:
access and trunks (inter switches) links



Access links belong to one VLAN
they do not carry 802.1q headers



Trunk links carry traffic for more than one VLAN
and as such carry 801.1q tagged frames



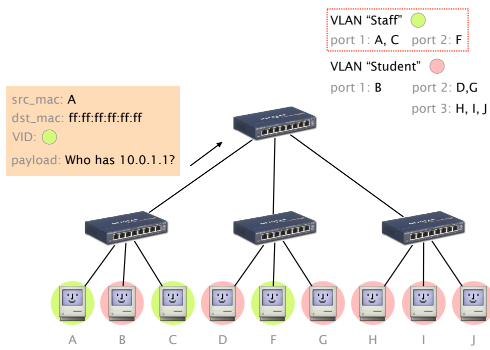
Each switch runs
one MAC learning algorithm for each VLAN

When a switch receives a frame with
an unknown or a broadcast destination,

it forwards it over all the ports that
belong to the same VLAN

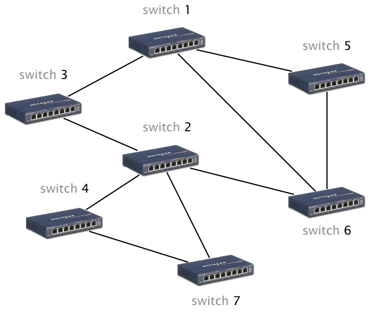
When a switch learns a source address on a port

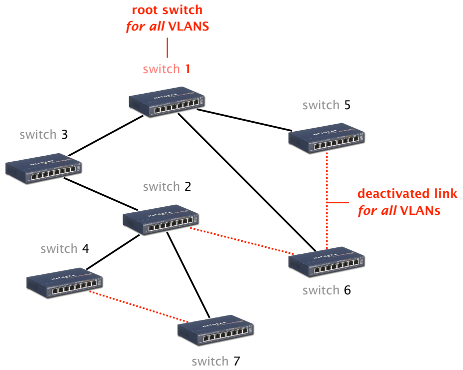
it associates it to the VLAN of this port and
only uses it when forwarding frames on this VLAN

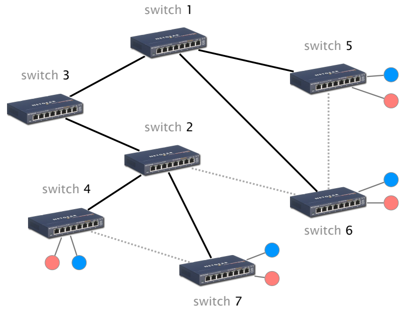


Switches can also compute per-VLAN spanning-tree allowing a distinct SPT for each VLAN

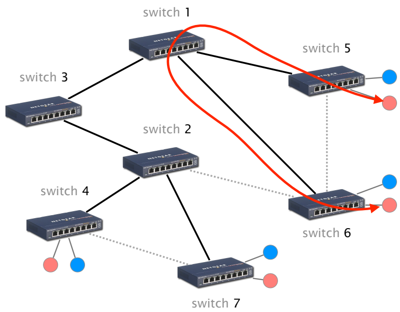
allow the operators to use more of their links

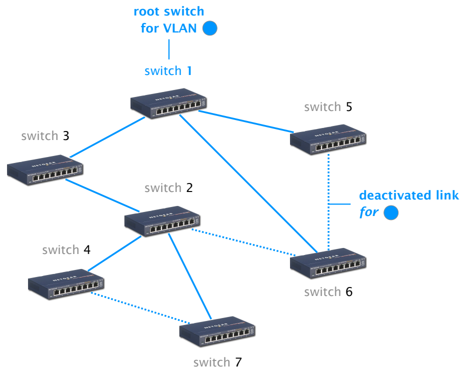


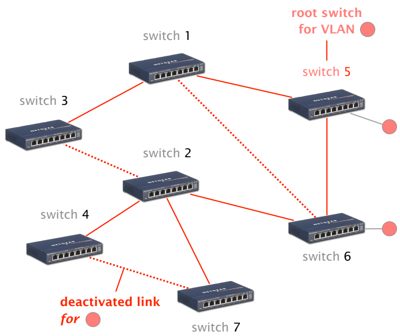




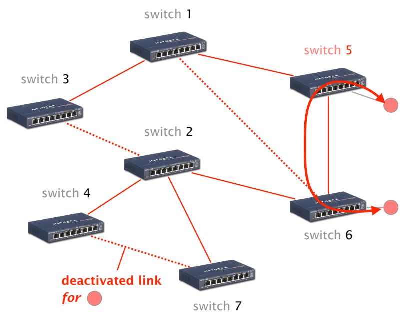
Any communication between the red hosts on switch 5 and 6 need to go via switch 1…







Now any communication between the red hosts on switch 5 and 6 go via the direct link

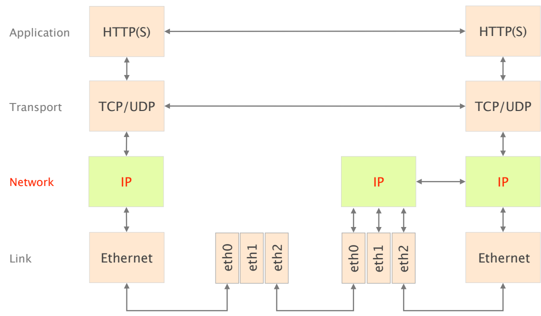


Link Layer

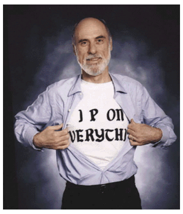
Network Layer

The Beginning

Moving on to IP and the network layer



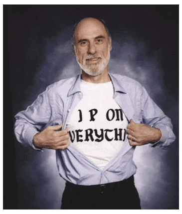
Internet Protocol and Forwarding



source: Boardwatch Magazine

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

Internet Protocol and Forwarding



1. IP addresses
   use, structure, allocation

IP forwarding
longest prefix match rule

IP header
IPv4 and IPv6, wire format

IPv4 addresses are unique 32-bits number associated to a network interface (on a host, a router, …)

IP addresses are usually written
using dotted-quad notation

82.130.102.10

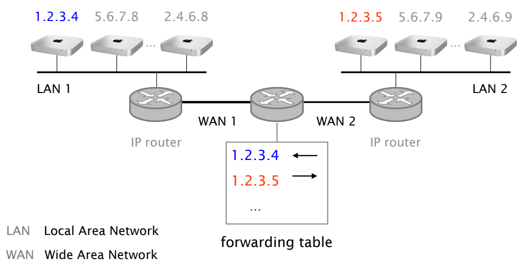
01010010 10000010 01100110 00001010

IPv6 addresses are unique 128-bits number associated to a network interface (on a host, a router, …)

Notation	8 groups of 16 bits each separated by colons (:) Each group is written as four hexadecimal digits	
Simplification	Leading zeros in any group are removed One section of zeros is replaced by a double colon (::) Normally the longest section	
Examples	1080:0:0:0:8:800:200C:417A	→ 1080::8:800:200C:417A
	FF01:0:0:0:0:0:0:0101	→ FF01::101
	0:0:0:0:0:0:0:1	→ ::1

Routers forwards IP packets based on their destination IP address

If IP addresses were assigned arbitrarily, routers would require forwarding entries for all of them



Two universal tricks you can apply to any computer sciences problem

When you need… more flexibility,
you add… a layer of indirection

When you need… more scalability,
you add… a hierarchical structure

When you need… more scalability,
you add… a hierarchical structure

IP addresses are hierarchically allocated, similarly to the postal service

Address	
Zip	8092
Street	Gloriastrasse
Building	35 (ETZ)
Location in building	G 90
Name	Laurent Vanbever

Nobody in the Swiss mail system knows where every single house or building is

principle: Routing tables are separated at each level of the hierarchy

each one with a manageable scale

Forwarding in the Swiss mail
in 4 steps

1. Deliver the letter to the post office responsible for the zip code
2. Assign letter to the mail person covering the street
3. Drop letter into the mailbox attached to the building
4. Hand in the letter to the appropriate person

IP addressing is hierarchical, composed of a prefix (network address) and a suffix (host address)

32 bits

01010010.10000010.01100110.00001010

prefix
identifies the network

suffix
identifies the hosts *in* the network

Each prefix has a given length, usually written using a "slash notation"

IP prefix 82.130.102.0 /24

prefix length (in bits)

Here, a /24 means that we have 8 bits left to address hosts address, enough for 256 hosts

82.130.102.0 /24

prefix part	host part	IP address
01010010.10000010.01100110.	00000000	82.130.102.0
01010010.10000010.01100110.	00000001	82.130.102.1
01010010.10000010.01100110.	00000010	82.130.102.2
01010010.10000010.01100110.	11111110	82.130.102.254
01010010.10000010.01100110.	11111111	82.130.102.255

In practice, the first and last IP address of a prefix are not usable

prefix part	host part	IP address
01010010.10000010.01100110.	00000000	82.130.102.0
01010010.10000010.01100110.	11111111	82.130.102.255

The address with the host part being all 0s identifies the network itself

prefix part	host part	IP address
01010010.10000010.01100110.	00000000	82.130.102.0

The address with the host part being all 1s identifies the broadcast address

prefix part	host part	IP address
01010010.10000010.01100110.	11111111	82.130.102.255

A /24 has therefore only 254 addresses that can be allocated to hosts

Prefixes are also sometimes specified using an address and a mask

Address 82.130.102.0

01010010.10000010.01100110. 00000000

11111111.11111111.11111111. 00000000

Mask 255.255.255.0

ANDing the address and the mask gives you the prefix

Address 82.130.102.0

01010010.10000010.01100110. 00000000

11111111.11111111.11111111. 00000000

Mask 255.255.255.0

Given this IP prefix 82.130.0.0/17

Compute

# of addressable hosts

the prefix mask

network address

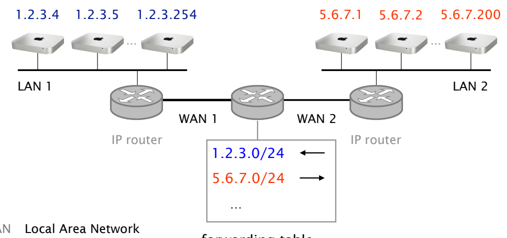
1st host address

last host address

broadcast address

Routers forward packet to their destination according to the network part, *not* the host part

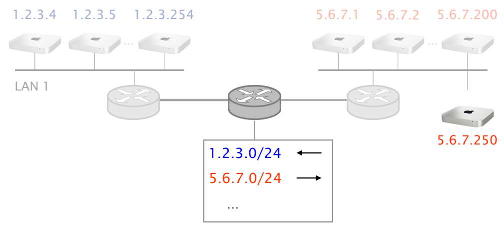
Doing so enables to scale the forwarding tables



forwarding table

LAN Local Area Network
WAN Wide Area Network

Hierarchical addressing enables to add new hosts without changing or adding forwarding rules



forwarding table

Originally, there were only 5 fixed allocation sizes, (or classes)—known as classful networking

	leading bits	prefix length	# hosts	start address	end address
class A	0	8	2^{24}	0.0.0.0	127.255.255.255
class B	10	16	2^{16}	128.0.0.0	191.255.255.255
class C	110	24	2^{8}	192.0.0.0	223.255.255.255
class D multicast	1110			224.0.0.0	239.255.255.255
class E reserved	1111			240.0.0.0	255.255.255.255

Classful networking was quite wasteful leading to IP address exhaustion

problem — Class C was too small, so everybody requested class B
which where: *i)* too big and *ii)* too few (wasted space)

solution — Classless Inter-Domain Routing (CIDR)
introduced in 1993

CIDR enabled flexible division between network and hosts addresses

CIDR must specify both the address and the mask
classful was communicating this in the first address bits

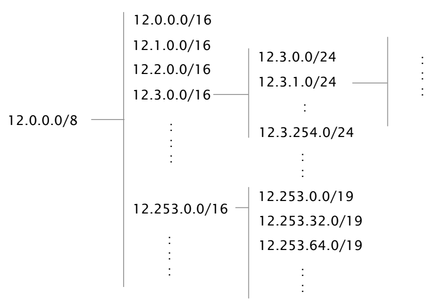
Masks are carried by the routing algorithms
it is *not* implicitly carried in the address

Say that an organization needs 500 addresses…

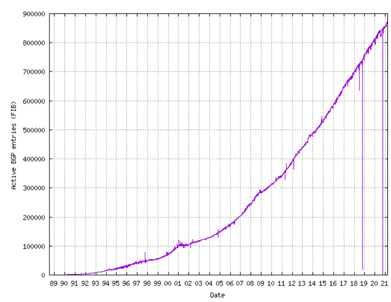
with…	it gets a…	leading to a waste of…
classful	class B (/16)	99%
CIDR	/23 (=2 class C's)	2%

With CIDR, the max. waste is bounded to 50% (why?)

Today, addresses are allocated in contiguous chunks

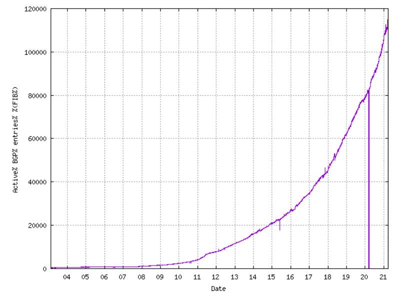


As of last week, the Internet has around 900,000 IPv4 prefixes



source http://www.cidr-report.org/

As of last week,
the Internet has around 115,000 IPv6 prefixes



source https://www.cidr-report.org/v6/as2.0/

The allocation process of IP address is also hierarchical

The root is held by Internet Corporation for Assigned Names and Numbers, aka ICANN

ICANN allocates large prefixes blocks to Regional Internet Registries (RIRs)

ARIN	LACNIC	RIPE NCC	APNIC	AFRINIC
America	Latin America	Europe	Asia-Pacific	Africa

RIRs allocates parts of these prefixes blocks to Internet Service Providers (ISPs) and large institutions

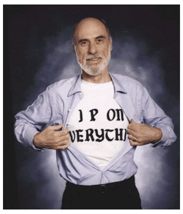
ISPs and large institutions may, in turn, allocate even smaller prefixes to their own customers

ICANN	ICANN gives RIPE Prefix	82.0.0.0/8 01010010
RIPE NCC	RIPE gives ETHZ Prefix	82.130.64.0/18 010100101000001001
ETH	ETHZ gives ITET/TIK Prefix	82.130.102.0/23 01010010100000100110011
DITET	ITET gives me Address	82.130.102.254 01010010100000100110011011111110

IP prefixes @ ETH

1	82.130.64.0/18	6	192.33.88.0/21
2	129.132.0.0/16	7	192.33.96.0/21
3	148.187.192.0/19	8	192.33.104.0/22
4	195.176.96.0/19	9	192.33.108.0/23
5	192.33.87.0/24	10	192.33.110.0/24

Internet Protocol and Forwarding

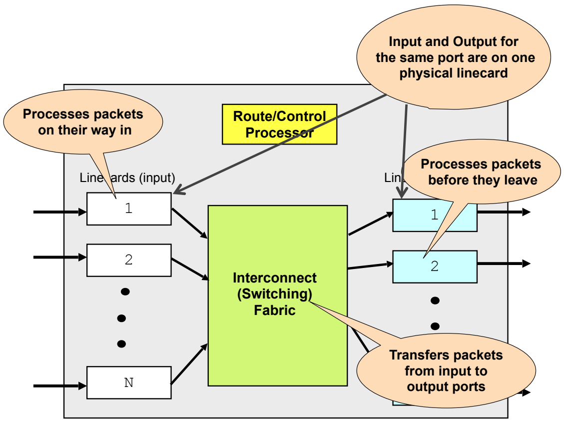


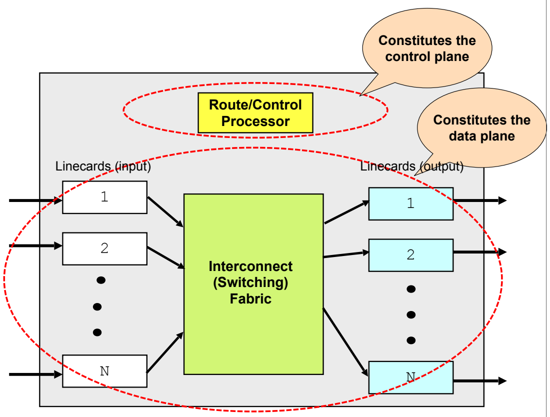
IP addresses
use, structure, allocation

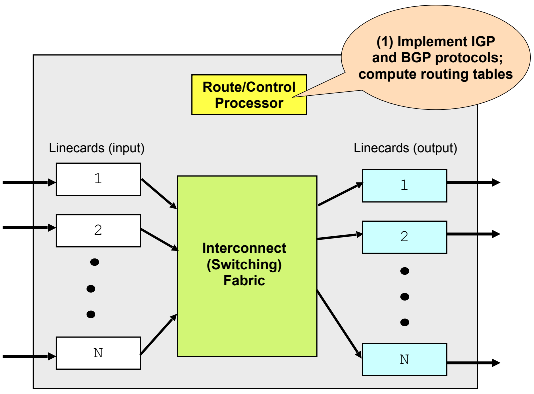
2 IP forwarding
longest prefix match rule

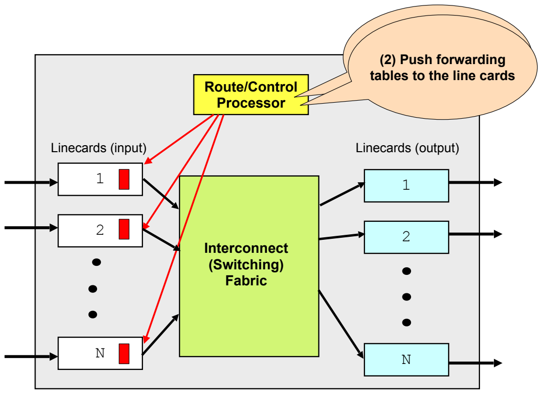
IP header
IPv4 and IPv6, wire format

What's inside an IP router?





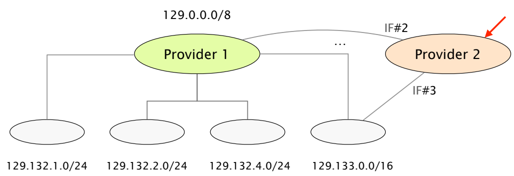




Routers maintain forwarding entries for each Internet prefix

Provider 2's Forwarding table

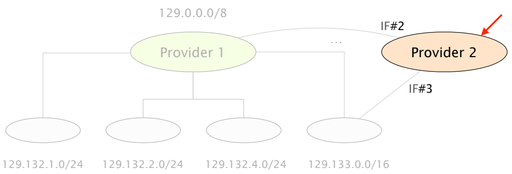
IP prefix	Output
129.0.0.0/8	IF#2
129.132.1.0/24	IF#2
129.132.2.0/24	IF#2
129.133.0.0/16	IF#3



Let's say a packet for 129.0.1.1 arrives at Provider 2

Provider 2's Forwarding table

IP prefix	Output
129.0.0.0/8	IF#2
129.132.1.0/24	IF#2
129.132.2.0/24	IF#2
129.133.0.0/16	IF#3



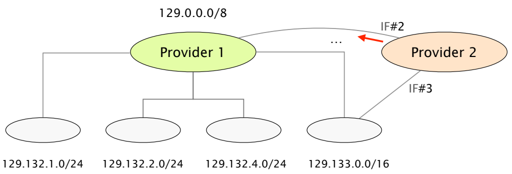
When a router receives an IP packet, it performs an IP lookup to find the matching prefix

Let's say a packet for 129.0.1.1 arrives at Provider 2

> Provider 2 forwards it to IF#2

Provider 2's Forwarding table

IP prefix	Output
129.0.0.0/8	IF#2
129.132.1.0/24	IF#2
129.132.2.0/24	IF#2
129.133.0.0/16	IF#3

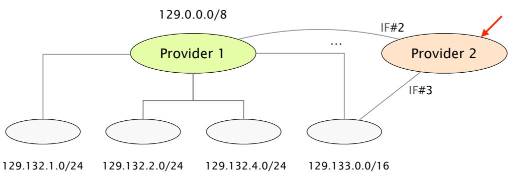


CIDR makes forwarding harder though, as one packet can match many IP prefixes

Let's say a packet for 129.133.0.1 arrives at Provider 2

Provider 2's Forwarding table

IP prefix	Output
129.0.0.0/8	IF#2
129.132.1.0/24	IF#2
129.132.2.0/24	IF#2
129.133.0.0/16	IF#3

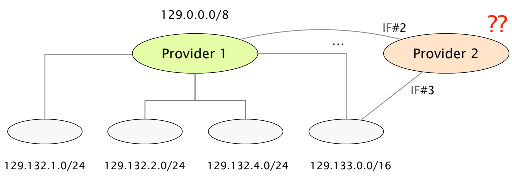


Let's say a packet for 129.133.0.1 arrives at Provider 2

We have two matches!

Provider 2's Forwarding table

IP prefix	Output
129.0.0.0/8	IF#2
129.132.1.0/24	IF#2
129.132.2.0/24	IF#2
129.133.0.0/16	IF#3



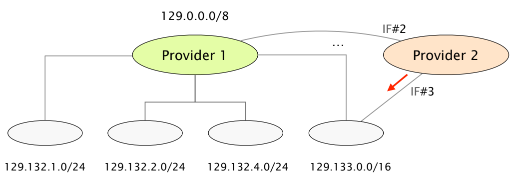
To resolve ambiguity, forwarding is done along the *most specific* prefix (*i.e.*, the longer one)

Let's say a packet for 129.133.0.1 arrives at Provider 2

> Provider 2 forwards it to IF#3

Provider 2's Forwarding table

IP prefix	Output
129.0.0.0/8	IF#2
129.132.1.0/24	IF#2
129.132.2.0/24	IF#2
129.133.0.0/16	IF#3

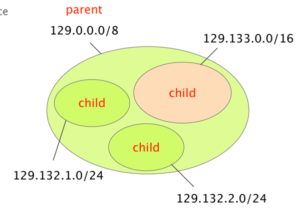


Could we do something better than maintaining one entry per prefix? *Yep!*

A child prefix can be filtered from the table whenever it shares the same output interface as its parent

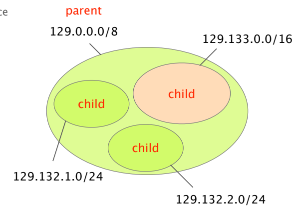
Routing Table

IP prefix	Output Interface
…	
129.0.0.0/8	IF#2
129.132.1.0/24	IF#2
129.132.2.0/24	IF#2
129.133.0.0/16	IF#3
…	



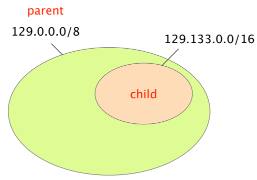
Routing Table

IP prefix	Output Interface
…	
129.0.0.0/8	IF#2
~~129.132.1.0/24~~	~~IF#2~~
~~129.132.2.0/24~~	~~IF#2~~
129.133.0.0/16	IF#3
…	



Routing Table

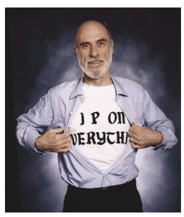
IP prefix	Output Interface
…	
129.0.0.0/8	IF#2
129.133.0.0/16	IF#3
…	



Exactly the same forwarding as before

Check out www.route-aggregation.net, to see how filtering can be done automatically

Internet Protocol and Forwarding



IP addresses
use, structure, allocation

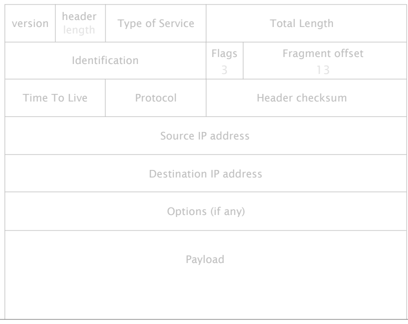
IP forwarding
longest prefix match rule

3 IP header
IPv4 and IPv6, wire format

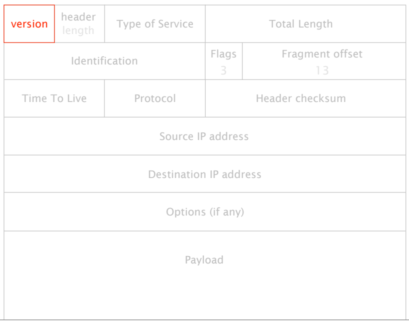
Here is what an IPv4 packet look like on a wire

32 bits

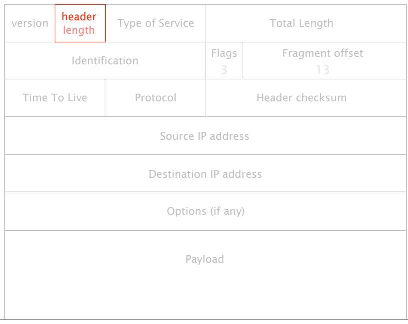
<table>
<tr><td>4</td><td>4</td><td>8</td><td colspan="2">16</td></tr>
<tr><td>version</td><td>header length</td><td>Type of Service</td><td colspan="2">Total Length</td></tr>
<tr><td colspan="3">Identification</td><td>Flags 3</td><td>Fragment offset 13</td></tr>
<tr><td colspan="2">Time To Live</td><td>Protocol</td><td colspan="2">Header checksum</td></tr>
<tr><td colspan="5">Source IP address</td></tr>
<tr><td colspan="5">Destination IP address</td></tr>
<tr><td colspan="5">Options (if any)</td></tr>
<tr><td colspan="5">Payload</td></tr>
</table>



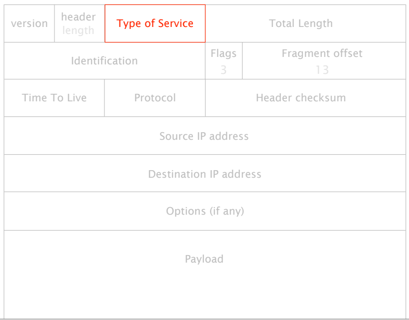
The version number tells us what other fields to expect, typically it is set to “4” for IPv4, or “6” for IPv6



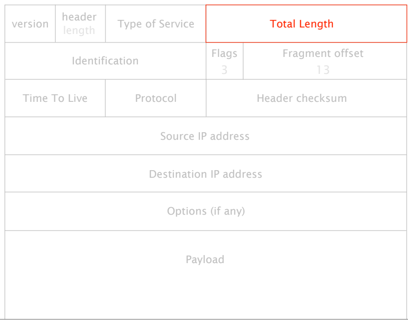
The header length denotes the number of 32-bits word in the header, typically set to 5 (20 bytes header)



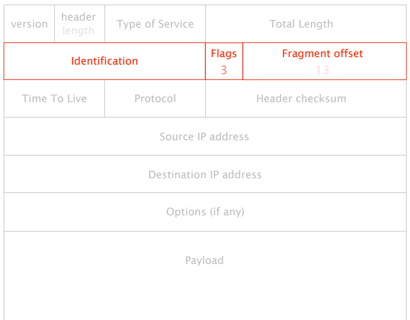
The ToS allows different packets to be treated differently, e.g., low delay for voice, high bandwidth for video



The total length denotes the # of bytes in the entire packet, with a maximum of 65 535 bytes



The next three fields are used when packets get fragmented



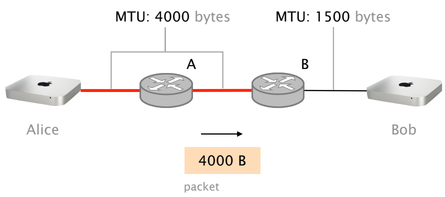
Every link in the Internet has a Maximum Transmission Unit (MTU)

MTU is the max. # of bytes a link can carry as one unit
e.g., 1500 bytes for normal Ethernet

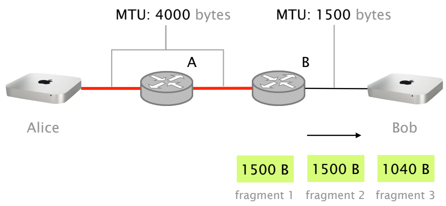
A router can fragment a packet if the outgoing link MTU is smaller than the total packet size

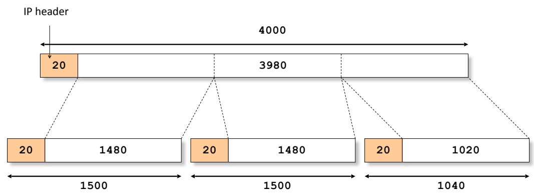
Fragmented packets are recomposed at the destination
why not in the network?

Assume Alice is sending 4000B packets to Bob, who is connected to a 1500B MTU link

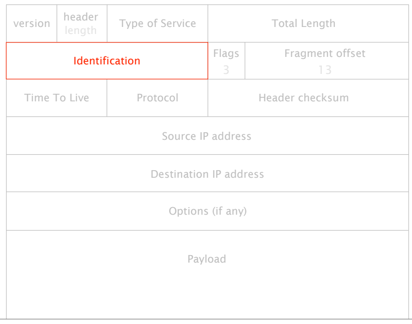


Because the packet is larger than the MTU, router B will split the packet into fragments

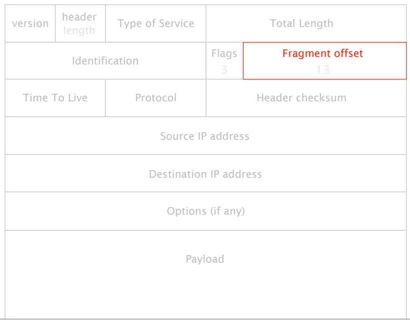




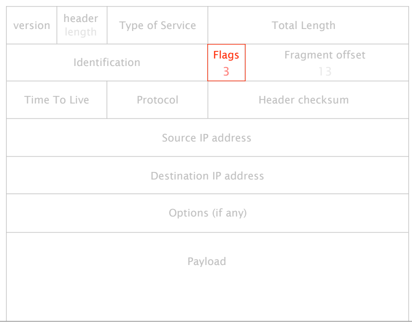
The Identification header uniquely identify the fragments of a particular packet



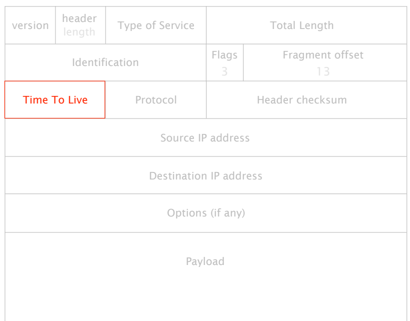
The fragment offset is used to put back the fragments in the right order in case of reordering



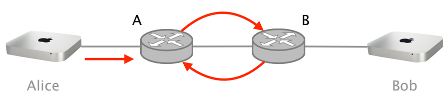
The flags is used to tell whether there are more fragments coming or not



The TTL is used to identify packets trapped in a loop, and eventually discard them



TTL is decremented by 1 at each router, the packet is discarded if it reaches 0

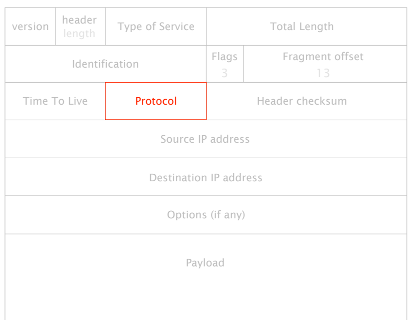


default TTL values

*nix (Linux/Mac)	64
Windows	128

(used for OS fingerprinting)

The protocol field identifies the higher level protocol carried in the packet, “6” for TCP, “17” for UDP



The checksum is the sum of all the 16 bits words in the header (does not protect the payload)

version	header length	Type of Service	Total Length
Identification		Flags 3	Fragment offset 13
Time To Live	Protocol	**Header checksum**	
Source IP address			
Destination IP address			
Options (if any)			
Payload			

The source and destination IP uniquely identifies the source and destination host

version	header length	Type of Service	Total Length
Identification		Flags 3	Fragment offset 13
Time To Live	Protocol	Header checksum	
Source IP address			
Destination IP address			
Options (if any)			
Payload			

Options were initially put to provide additional flexibility.
For security reasons, there are often deactivated.

version	header length	Type of Service	Total Length
Identification		Flags 3	Fragment offset 13
Time To Live	Protocol	Header checksum	
Source IP address			
Destination IP address			
Options (if any)			
Payload			

IP options

- Record route
- Strict source route
- Loose source route
- Timestamp
- Traceroute
- Router alert
- ...

see http://www.networksorcery.com/enp/protocol/ip.htm#Options for a full list

While there are no new IPv4 available,
IPv4 still accounts for most of the Internet traffic (for now)

IPv6 addresses are unique 128-bits number associated to a network interface (on a host, a router, ...)

Notation: 8 groups of 16 bits each separated by colons (:)
Each group is written as four hexadecimal digits

Simplification: Leading zeros in any group are removed
One section of zeros is replaced by a double colon (::)
Normally the longest section

Examples:
1080:0:0:0:8:800:200C:417A → 1080::8:800:200C:417A
FF01:0:0:0:0:0:0:0101 → FF01::101
0:0:0:0:0:0:0:1 → ::1

With respect to IPv4,
IPv6 is simpler

IPv6 was motivated by address exhaustion
IPv6 addresses are 128 bits long, that's plenty!

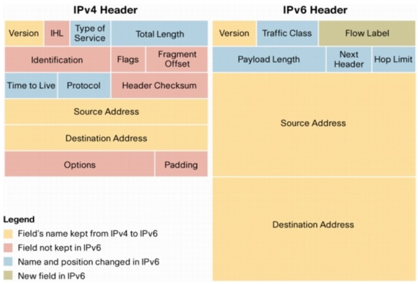
IPv6 got rid of anything that wasn't necessary
spring cleaning

Result is an elegant, if unambitious, protocol

With respect to IPv4,
IPv6 is simpler

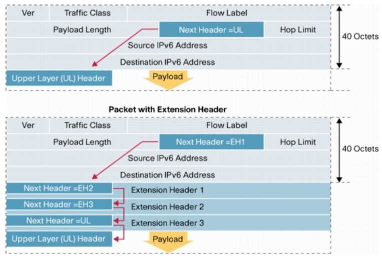
IPv6	reason
removed	
fragmentation, checksum	leave problems to the end host
header length	simplify handling
added...	
new options mechanism	simplify handling
expanded addresses	
flow label	flexibility

IPv4 vs IPv6



source http://bit.ly/1HXc2BS

IPv6 enables to insert arbitrary options in the packet
see RFC 2460



source http://bit.ly/1HXc2BS

The problem with IPv4 options is that all of them must be processed by each router, which is slow

In IPv6, only one type of optional header must be processed by each router

There are three types of IPv6 addresses: unicast, anycast, and multicast

Unicast	Identifies a single interface	Packets are delivered to this specific interface
Anycast	Identifies a set of interfaces	Packets are delivered to the "nearest" interface
Multicast	Identifies a set of interfaces	Packets are delivered to **all** interfaces

Unicast — Identifies a single interface
Packets are delivered to this specific interface

Global unicast addresses are hierarchically allocated

similar to global IPv4 addresses

128 bits

N bits	M bits	128-N-M bits
global routing prefix	subnet ID	Interface ID
Identifies the ISP responsible for the address	A subnet in this ISP or a customer of the ISP	Usually 64 bits Based on the MAC address

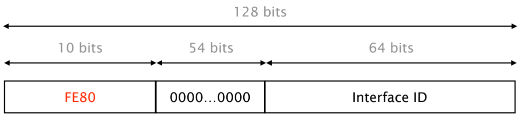
Allocation of IPv6 (global unicast) addresses

The Internet Assigned Numbers Authority (IANA) assigns blocks to Regional IP address Registries (RIR)
For example RIPE, ARIN, APNIC, …

Currently, only 2000::/3 is used for global unicast
All addresses are in the range of 2000 to 3FFF

Link-local addresses are unique to a single link (subnet)

same as private IPv4 addresses



Each host/router **must** generate a link-local address for **each** of its interfaces

An interface therefore can have multiple IPv6 addresses

In addition to global and link-local addresses, some IPv6 unicast addresses have a special meaning

Unspecified address	0:0:0:0:0:0:0:0 Used as src address if no IPv6 address available
Loopback address	0:0:0:0:0:0:0:1 → ::1 127.0.0.1 for IPv4 addresses
IPv4 embedded	The lowest 32 bits contains an IPv4 address useful when deploying IPv6
Important	There are no IPv6 broadcast addresses

Anycast	Identifies a set of interfaces Packets are delivered to the „nearest“ interface

IPv6 anycast addresses

Multiple interfaces with the same address

Packets are sent to the nearest interface

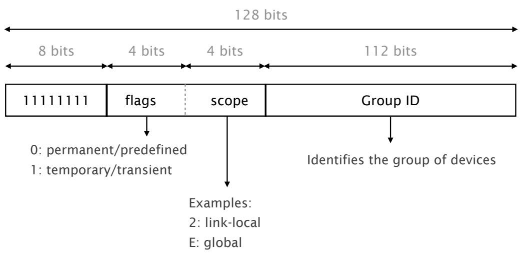
Anycast use the global unicast address range

E.g. for DNS or HTTP services

IPv6 anycast is rarely used

Multicast	Identifies a set of interfaces Packets are delivered to **all** interfaces

Multicast addresses identify a group of receivers/interfaces



Some multicast addresses are well-known and used for auto-discovery, bootstraping, etc.

FF02::1	All IPv6 end-systems E.g. hosts, servers, routers, mobile devices, …
FF02::2	All IPv6 routers All routers automatically belong to this group

Thus far IPv4 has been very persistent, and that's quite understandable

Deploying IPv6 require every device to support it

All routers, middleboxes, end hosts, applications, …

Most of IPv6 new features were back-ported to IPv4

No obvious advantage in using IPv6

Network Address Translation is working well

The pain of address depletion is not obvious

Network Address Translation (NAT)

Sharing a single (public) address between hosts

Port numbers (transport layer) are used to distinguish

One of the main reasons why we can still use IPv4

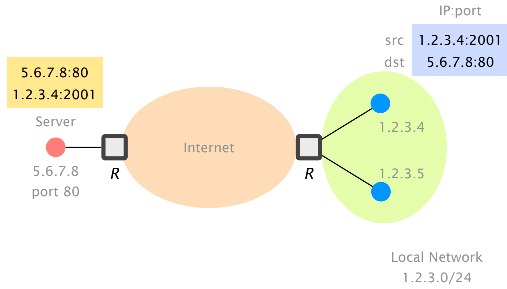
Saved us from address depletion

Violates the general end-to-end principle of the Internet

A NAT box adds a layer of indirection

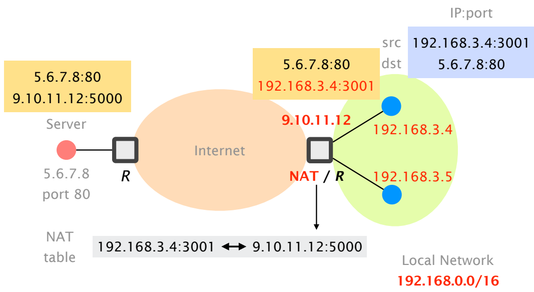
The Internet before NAT

Every machine connected to the Internet had a unique IP



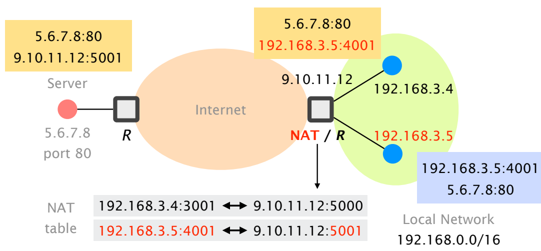
The Internet with NAT

Hosts behind NAT get a private address



The Internet with NAT

The port numbers are used to multiplex single addresses



NAT also provides other (dis-)advantages

Better privacy/anonymization

All hosts in one network get the same public IP

But, cookies, browser version, … still identify hosts

Better security

From the outside you cannot directly reach the hosts

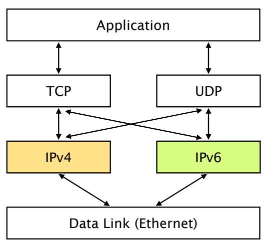
Problematic e.g., for online gaming

Limited scalability (size of the mapping table)

Example: Wi-Fi access problems in public places

(e.g., lecture hall) often due to a full NAT table

Today, a lot of applications and OSes use a dual stack approach



Over the years, a lot of transition mechanisms were developed

6in4

6to4

Teredo

SIIT

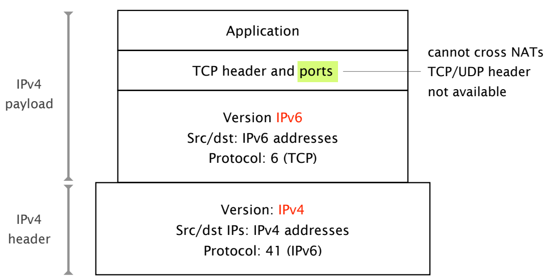
6rd

GRE

AYiYA

…

Tunnel IPv6 packets over static IPv4 links (6in4)



6to4 uses special IPv6 addresses

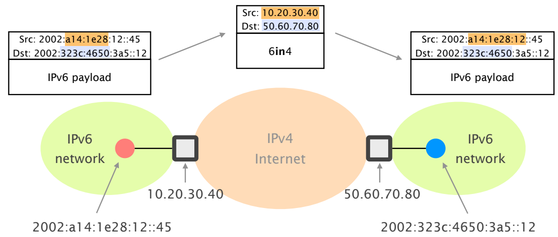
16 bits	32 bits	16 bits	64 bits
2002	IPv4 address	subnet ID	host ID

IPv4: 192.15.3.73

c0.0f.03.49

6to4: 2002:c00f:0349::/48

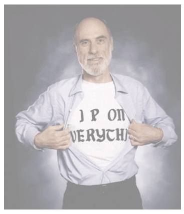
6to4 transmits IPv6 packets over IPv4 networks without explicit tunnels



IPv6 @ home (Swisscom Internet access box)

You will be assigned an IPv4 and IPv6 address

Internet Protocol and Forwarding



IP addresses
use, structure, allocation

IP forwarding
longest prefix match rule

IP header
IPv4 and IPv6, wire format

After the Easter break on
Communication Networks

Internet routing!

Communication Networks
Spring 2021

Laurent Vanbever
nsg.ee.ethz.ch

ETH Zürich (D-ITET)
March 29 2021