Announcing our 2022 “Connectivity Fäscht”

When: Thursday, 07.04.2022, 18:00—21:30
Where: HG E7
Topics:
- Awakening of the mini-Internet
- We'll connect all ASes together!
- Interesting demos and detailed explanations
- Great possibility to work on the project
- A lot of TAs will be there to support you
Attendance: Not mandatory… but try to make it: it’s fun!

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?

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
Who am I?
MAC-to-IP binding

Who are you?
IP-to-MAC binding

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

Given an IP address reachable on a link,
IP-to-MAC binding
MAC-to-IP binding
Address Resolution Protocol

Moving on to IP and the network layer
Application
HTTP(S)
TCP/UDP
IP
Ethernet
Link
eth0
eth1
eth2

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

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

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
82.130.102.10
IPv4 addresses are unique 32-bits number
associated to a network interface (on a host, a router, …)

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.

IP addresses are hierarchically allocated, similarly to the postal service.

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.

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

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).

Each prefix has a given length, usually written using a “slash notation”.

<table>
<thead>
<tr>
<th>IP prefix</th>
<th>82.130.102.0 /24</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prefix length</td>
<td>(in bits)</td>
</tr>
</tbody>
</table>
Here, a /24 means that we have 8 bits left to address hosts address, enough for 256 hosts.

<table>
<thead>
<tr>
<th>prefix part</th>
<th>host part</th>
<th>IP address</th>
</tr>
</thead>
<tbody>
<tr>
<td>01010010.10000010.01100110.00000000</td>
<td>00000000</td>
<td>82.130.102.0</td>
</tr>
<tr>
<td>01010010.10000010.01100110.00000000</td>
<td>00000001</td>
<td>82.130.102.1</td>
</tr>
<tr>
<td>01010010.10000010.01100110.00000000</td>
<td>00000010</td>
<td>82.130.102.2</td>
</tr>
<tr>
<td>01010010.10000010.01100110.11111111</td>
<td>00000001</td>
<td>82.130.102.254</td>
</tr>
<tr>
<td>01010010.10000010.01100110.11111111</td>
<td>00000010</td>
<td>82.130.102.255</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>prefix part</th>
<th>host part</th>
<th>IP address</th>
</tr>
</thead>
<tbody>
<tr>
<td>01010010.10000010.01100110.00000000</td>
<td>00000000</td>
<td>82.130.102.0</td>
</tr>
<tr>
<td>01010010.10000010.01100110.11111111</td>
<td>00000000</td>
<td>82.130.102.255</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>prefix part</th>
<th>host part</th>
<th>IP address</th>
</tr>
</thead>
<tbody>
<tr>
<td>01010010.10000010.01100110.00000000</td>
<td>00000000</td>
<td>82.130.102.0</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>prefix part</th>
<th>host part</th>
<th>IP address</th>
</tr>
</thead>
<tbody>
<tr>
<td>01010010.10000010.01100110.11111111</td>
<td>00000000</td>
<td>82.130.102.255</td>
</tr>
</tbody>
</table>

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

Prefixes are also sometimes specified using an address and a mask.

<table>
<thead>
<tr>
<th>Address</th>
<th>Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>82.130.102.0</td>
<td>255.255.255.0</td>
</tr>
</tbody>
</table>

ANDing the address and the mask gives you the prefix.

<table>
<thead>
<tr>
<th>Address</th>
<th>Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>82.130.102.0</td>
<td>255.255.255.0</td>
</tr>
</tbody>
</table>

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.

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

Classful networking was quite wasteful leading to IP address exhaustion.

CIDR enabled flexible division between network and hosts addresses.

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

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

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 CIDR, the max. waste is bounded to 50% (why?)
As of last week, the Internet has ~150,000 IPv6 prefixes.

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).

RIRs allocate 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 gives RIPE 82.0.0.0/8
Prefix 01010010

RIPE gives ETHZ 82.130.64.0/18
Prefix 01010010 000001000001

ETHZ gives ITET/TIK 82.130.102.0/23
Prefix 01010010 00000011110011

ITET gives me 82.130.102.254
Address 01010010 0000000101101010000000000000000000000000
Internet Protocol and Forwarding

IP addresses
- use, structure, allocation

IP forwarding
- longest prefix match rule

IP header
- IPv4 and IPv6, wire format

What's inside an IP router?

Processes packets on their way in

Route/Control Processor

Interconnect (Switching) Fabric

Processes packets before they leave

Transfers packets from input to output ports

Routers maintain forwarding entries for each Internet prefix

Provider 1

129.133.0.0/16
129.132.1.0/24
129.132.2.0/24
129.133.0.0/16

Provider 2

129.0.0.0/8
129.132.1.0/24
129.132.2.0/24
129.133.0.0/16
129.132.1.0/24
129.132.2.0/24
Let's say a packet for 129.0.1.1 arrives at Provider 2

Provider 2's Forwarding table:

<table>
<thead>
<tr>
<th>IP prefix</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>129.0.0.0/8</td>
<td>IF#2</td>
</tr>
<tr>
<td>129.132.1.0/24</td>
<td>IF#2</td>
</tr>
<tr>
<td>129.132.2.0/24</td>
<td>IF#2</td>
</tr>
<tr>
<td>129.133.0.0/16</td>
<td>IF#3</td>
</tr>
</tbody>
</table>

> Provider 2 forwards it to IF#2

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

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

We have two matches!

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

> Provider 2 forwards it to 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

<table>
<thead>
<tr>
<th>IP prefix</th>
<th>Output Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>129.0.0.0/8</td>
<td>#2</td>
</tr>
<tr>
<td>129.132.1.0/24</td>
<td>#2</td>
</tr>
<tr>
<td>129.132.2.0/24</td>
<td>#2</td>
</tr>
<tr>
<td>129.133.0.0/16</td>
<td>#3</td>
</tr>
</tbody>
</table>

Exactly the same forwarding as before

Check out [www.route-aggregation.net](http://www.route-aggregation.net), to see how filtering can be done automatically.

Here is what an IPv4 packet look like on a wire:

**Internet Protocol and Forwarding**

- **IP addresses**
  - See, structure, allocation
- **IP forwarding**
  - Longest prefix match rule
- **IP header**
  - IPv4 and IPv6, wire format

<table>
<thead>
<tr>
<th>Field</th>
<th>Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>4</td>
</tr>
<tr>
<td>Header length</td>
<td>4</td>
</tr>
<tr>
<td>Type of service</td>
<td>8</td>
</tr>
<tr>
<td>Total length</td>
<td>16</td>
</tr>
<tr>
<td>Identification</td>
<td></td>
</tr>
<tr>
<td>Flags</td>
<td></td>
</tr>
<tr>
<td>Fragment offset</td>
<td></td>
</tr>
<tr>
<td>Time to live</td>
<td></td>
</tr>
<tr>
<td>Protocol</td>
<td></td>
</tr>
<tr>
<td>Header checksum</td>
<td></td>
</tr>
<tr>
<td>Source IP address</td>
<td></td>
</tr>
<tr>
<td>Destination IP address</td>
<td></td>
</tr>
<tr>
<td>Options (if any)</td>
<td></td>
</tr>
<tr>
<td>Payload</td>
<td></td>
</tr>
</tbody>
</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).

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:

MTU: 4000 bytes

MTU: 1500 bytes

1500 B

fragment 1

A B

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

1500 B

fragment 2

1040 B

fragment 3

The Identification header uniquely identifies the fragments of a particular packet:

version: 4
header length: 20
Type of Service (TOS) 0
Total Length 4000
Identification

Flags

Fragment offset

Payload

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

version: 4
header length: 20
Type of Service (TOS) 0
Total Length 4000
Identification

Flags

Fragment offset

Payload

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

version: 4
header length: 20
Type of Service (TOS) 0
Total Length 4000
Identification

Flags

Fragment offset

Payload

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

version: 4
header length: 20
Type of Service (TOS) 0
Total Length 4000
Identification

Flags

Fragment offset

Payload

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

version: 4
header length: 20
Type of Service (TOS) 0
Total Length 4000
Identification

Flags

Fragment offset

Payload

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

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

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

IP options:
- Record route
- Strict source route
- Loose source route
- Timestamp
- Traceroute
- Router alert
- IP options

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

IPv6 is simpler than IPv4:
- IPv6 got rid of anything that wasn’t necessary. Spring cleaning.
- Result is an elegant, if unambitious, protocol.
IPv4 vs IPv6

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

One 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:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unicast</td>
<td>Identifies a single interface. Packets are delivered to this specific interface.</td>
</tr>
<tr>
<td>Anycast</td>
<td>Identifies a set of interfaces. Packets are delivered to the &quot;nearest&quot; interface.</td>
</tr>
<tr>
<td>Multicast</td>
<td>Identifies a set of interfaces. Packets are delivered to all interfaces.</td>
</tr>
</tbody>
</table>

Global unicast addresses are **hierarchically allocated** similar to global IPv4 addresses:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>N bits</td>
<td>Represent the global routing prefix and identifies the ISP</td>
</tr>
<tr>
<td>M bits</td>
<td>Represents the subnet ID and identifies a subnet in the ISP</td>
</tr>
<tr>
<td>128-N-M</td>
<td>bits</td>
</tr>
<tr>
<td></td>
<td>Represent the interface ID and identifies a customer of the ISP</td>
</tr>
</tbody>
</table>

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

128 bits
10 bits 54 bits 64 bits

FE80 0000...0000 Interface ID

Each host/router must generate a link-local address for each of its interfaces. An interface therefore can have multiple IPv6 addresses.

Thus far IPv4 has been very persistent, and that's quite understandable. Most of IPv6 new features were back-ported to IPv4. No obvious advantage in using IPv6. Deploying IPv6 requires every device to support it. All routers, middleboxes, end hosts, applications, …

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.

Network Address Translation (NAT)

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 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.

Application
TCP
UDP
IPv4
IPv6
Data Link (Ethernet)
Over the years, a lot of transition mechanisms were developed:

- 6in4
- 6to4
- Teredo
- SIT
- 6rd
- GRE
- AYiYA
- ...

6in4 transmits IPv6 packets over statically-configured IPv4 tunnels:

- Application
- TCP header and ports
- Version: IPv6
- Src/dst: IPv6 addresses
- Protocol: 6 (TCP)

6to4 transmits IPv6 packets over IPv4 networks without explicit tunnels:

- IPv4 network
- IPv6 network
- 10.20.30.40
- 2002:a14:1e28:12::45
- IPv4 payload
- IPv6 payload
- IPv4 header
- IPv6 header

6to4 uses special IPv6 addresses:

<table>
<thead>
<tr>
<th>16 bits</th>
<th>32 bits</th>
<th>64 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>2002</td>
<td>IPv4 address</td>
<td>subnet ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>192.168</td>
<td>192.15.3.73</td>
<td></td>
</tr>
<tr>
<td>c0.0f.03.49</td>
<td>00.0f.03.49</td>
<td>2002:c00f:0349::/48</td>
</tr>
</tbody>
</table>

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

Communication Networks
Spring 2022

Laurent Vanbever
nsg.ee.ethz.ch

ETH Zürich (D-ITET)
21 March 2022