The Network Layer
IPv4 and IPv6
Part 1
Jean-Yves Le Boudec
2017
Contents

1. The Two Principles of IP Unicast
   2. IPv4 addresses
   3. IPv6 addresses
   4. NATs
   5. Subnets and Masks

Textbook

Chapter 5: The Network Layer
IP Principle #1 =
Structured addresses + Longest prefix match

Recall goal of Internet Protocol (IP) = interconnect all systems in the world

Principle #1:
- every interface has an IP address
- IP address is structured to reflect where the system is in the world
- every packet contains IP address of destination
- every system has a forwarding table (= routing table) and performs longest prefix match on destination address
IP Principle #2 = Don’t use routers inside a LAN

B ↔ E and W ↔ P should not go through router
W ↔ E goes through router
Terminology: LAN = subnet
IP principle 2 says: between subnets use routers, inside subnet don’t
We observe a packet from W to P at 1. Which IP destination address do we see?

A. The IP address of P
B. The IP address of an Ethernet interface of the Ethernet concentrator
C. There is no destination IP address in the packet since communication is inside the subnet and does not go through a router
D. I don’t know
The Internet Protocol (IP)

Communication between IP hosts requires knowledge of IP addresses

An IP address is unique across the whole network (= the world in general)

An IP address is the address of an interface

There are two versions: IPv4 (current version) and IPv6 (next version)

- There are two network layers: Internet4 and Internet6

Terminology:

- packet = IP data unit
- intermediate system = system that forwards data units to another system; an IP intermediate system is called a “router”
- an IP system that does not forward is called a “host”
2. IPv4 addresses

IPv4 address
- Uniquely identifies one interface in the world (in principle)
- An IPv4 address is 32 bits, usually noted in dotted decimal notation

**dotted decimal**: 4 integers (one integer = 8 bits)
- example 1: 128. 191. 151. 1
- example 2: 129. 192. 152. 2

**hexadecimal**: 8 hexa digits (one hexa digit = 4 bits)
- example 1: x80 bf 97 01
- example 2: x81 c0 98 02

**binary**: 32 bits
- example 1: b1000 0000 1011 1111 1001 0111 0000 0001
- example 2: b1000 0001 1100 0000 1001 1000 0000 0010
Binary, Decimal and Hexadecimal

Given an integer B “the basis”: any integer can be represented in “base B” by means of an alphabet of B symbols

Usual cases are

- decimal: 234
- binary: 1110 1010
- hexadecimal: ea

Mapping binary <-> hexa is simple: one hexa digit is 4 binary digits

- $e_{hex} = 1110_{bin}$
- $a_{hex} = 1010_{bin}$
- $ea_{hex} = b1110 1010_{bin}$

Mapping binary <-> decimal is best done by a calculator

- $1110 1010_{bin} = 128 + 64 + 32 + 8 + 2 = 234$

Special Cases to remember

- $f_{hex} = 1111_{bin} = 15$
- $ff_{hex} = 11111111_{bin} = 255$
Network Prefix

Network prefixes are used in routing tables

/24 is the prefix length in bits

Extract from routing table at sw-la-0

<table>
<thead>
<tr>
<th>Destination</th>
<th>Next hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.178/16</td>
<td>128.178.47.5</td>
</tr>
<tr>
<td>0/0</td>
<td>130.59.23.2</td>
</tr>
</tbody>
</table>

Extract from routing table at ed0-swi

<table>
<thead>
<tr>
<th>Destination</th>
<th>Next hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.178.29/24</td>
<td>128.178.100.2</td>
</tr>
<tr>
<td>128.178/16</td>
<td>128.178.100.3</td>
</tr>
<tr>
<td>0/0</td>
<td>128.178.47.3</td>
</tr>
</tbody>
</table>
## Special Addresses

<table>
<thead>
<tr>
<th>Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0.0.0</td>
<td>absence of address</td>
</tr>
<tr>
<td>127.0.0/24, for example 127.0.0.1</td>
<td>this host (loopback address)</td>
</tr>
<tr>
<td>10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16</td>
<td>private networks (e.g. in IEW) cannot be used on the public Internet</td>
</tr>
<tr>
<td>169.254.0.0/16</td>
<td>link local address (can be used only between systems on same LAN)</td>
</tr>
<tr>
<td>224/4</td>
<td>multicast</td>
</tr>
<tr>
<td>240/5</td>
<td>reserved</td>
</tr>
<tr>
<td>255.255.255.255/32</td>
<td>link local broadcast</td>
</tr>
</tbody>
</table>
We will see the functions of the fields other than the addresses in a following module.
3. IPv6 Addresses

The old version of IP is IPv4. IPv6 is the newer version of IP

*Why* a new version?
- IPv4 address space is too small (32 bits → \( \approx 4 \cdot 10^9 \) addresses)

*What* does IPv6 do?
- Redefine packet format with a larger address: 128 bits (\( \approx 3 \cdot 10^{38} \) addresses)
- Otherwise essentially the same as IPv4

IPv6 is incompatible with IPv4; routers and hosts must handle both separately
- A can talk to W, B can talk to W, A and B cannot communicate at the network layer
# Routing Tables at ed0-swi

<table>
<thead>
<tr>
<th>Destination</th>
<th>Next hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001:620:618:1a4/64</td>
<td>fe80::1%1</td>
</tr>
<tr>
<td>2001:620:618/48</td>
<td>fe80::4%2</td>
</tr>
<tr>
<td>::/0</td>
<td>fe80::1%2</td>
</tr>
</tbody>
</table>

**IP address of next hop**

**Interface number**

<table>
<thead>
<tr>
<th>Destination</th>
<th>Next hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.178.29/24</td>
<td>128.178.100.2</td>
</tr>
<tr>
<td>128.178/16</td>
<td>128.178.100.3</td>
</tr>
<tr>
<td>0/0</td>
<td>128.178.47.3</td>
</tr>
</tbody>
</table>

**Interface number**

**IP address of next hop**
IPv6 addresses are 128 bit long and are written using hexadecimal digits.

- An EPFL public address: `2001:620:618:1a6:0a00:20ff:fe78:30f9`
- An EPFL private address: `fd24:ec43:12ca:1a6:0a00:20ff:fe78:30f9`

This is a private address.

EPFL private
Compression Rules for IPv6 Addresses

1 *piece* = 16 bits = [0-4 ]hexa digits; leading zeroes in one piece are omitted;
prefer lower case
pieces separated by “:” (colon)
one IPv6 address uncompressed = 8 pieces
:: replaces any number of 0s in more than one piece;
appears only once in address

<table>
<thead>
<tr>
<th>uncompressed</th>
<th>compressed</th>
</tr>
</thead>
</table>
A Few IPv6 Global Unicast Addresses

The block 2000/3 (i.e. 2xxx and 3xxx) is allocated for global unicast addresses

<table>
<thead>
<tr>
<th>Address</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001:620::/32</td>
<td>Switch</td>
</tr>
<tr>
<td>2001:620:618::/48</td>
<td>EPFL</td>
</tr>
<tr>
<td>2001:620:8::/48</td>
<td>ETHZ</td>
</tr>
<tr>
<td>2a02:1200::/27</td>
<td>Swisscom</td>
</tr>
<tr>
<td>2001:678::/29</td>
<td>provider independent address</td>
</tr>
<tr>
<td>2001::/32</td>
<td>Teredo</td>
</tr>
<tr>
<td>2002::/16</td>
<td>6to4</td>
</tr>
</tbody>
</table>
### Examples of Special Addresses

<table>
<thead>
<tr>
<th>Address Structure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>::/128</td>
<td>absence of address</td>
</tr>
<tr>
<td>::1/128</td>
<td>this host (loopback address)</td>
</tr>
<tr>
<td>fc00::/7 (i.e. fcxx: and fdxx:)</td>
<td>Unique local addresses = private networks (e.g. in IEW) cannot be used on the public Internet</td>
</tr>
<tr>
<td>For example fd24:ec43:12ca:1a6:a00:20ff:fe78:30f9</td>
<td></td>
</tr>
<tr>
<td>fe80::/10</td>
<td>link local address (can be used only between systems on same LAN)</td>
</tr>
<tr>
<td>ff00::/8</td>
<td>multicast</td>
</tr>
<tr>
<td>ff02::1:ff00:0/104</td>
<td>Solicited node multicast</td>
</tr>
<tr>
<td>ff02::1/128</td>
<td>link local broadcast</td>
</tr>
<tr>
<td>ff02::2/128</td>
<td>all link local routers</td>
</tr>
</tbody>
</table>
IPv6 Packet Format

We will see the functions of the fields other than the addresses in a following module.
The dotted decimal notation for $80c1: ffff$ is ...

A. 128.193.255.255
B. 228.393.255.255
C. I don’t know
The hexadecimal notation «2001::bad:babe» denotes a string of ...

A. 32 bits
B. 44 bits
C. 48 bits
D. 64 bits
E. 128 bits
F. None of the above
G. I don’t know
4. NATs: Why invented?
(Network Address Translation boxes)

Goal: re-use same IP address for several devices / use private addresses
This is a special type of «middle box», that is violating the TCP/IP architecture
Used in residential networks («ADSL Modem») / in smartphones / in companies to save IP addresses
What does Network Address Translation do?

NAT box modifies IP address and port numbers (port numbers are in TCP and UDP headers – see transport protocol module)

Exact matching from NAT Table

Implemented by iptables on Linux -- iptables modifies the TCP and IP headers before forwarding (“prerouting”) or after (“postrouting”)

IPv 4 NAT box

<table>
<thead>
<tr>
<th>LAN</th>
<th>Internet</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.10.10 udp 1029</td>
<td>130.104.228.200 udp 3441</td>
</tr>
<tr>
<td>192.168.10.11 udp 1029</td>
<td>130.104.228.200 udp 3442</td>
</tr>
</tbody>
</table>
Creating a NAT table entry: on the fly

<table>
<thead>
<tr>
<th>LAN</th>
<th>Internet</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.10.10 udp 1029</td>
<td>130.104.228.200 udp 3441</td>
</tr>
<tr>
<td>192.168.10.11 udp 1029</td>
<td>130.104.228.200 udp 3442</td>
</tr>
<tr>
<td>192.168.10.11 tcp 1765</td>
<td><strong>130.104.228.200 tcp 2343</strong></td>
</tr>
</tbody>
</table>

From: **192.168.10.11 TCP : 1765**
To: 201.19.32.45 Port 80

From: **130.104.228.200 TCP : 2343**
To: 201.19.32.45 Port 80
Creating a NAT table entry: on the fly

From: 192.168.10.11 TCP : 1765
To: 201.19.32.45 Port 80

From: 130.104.228.200 TCP : 2343
To: 201.19.32.45 Port 80

IPv 4 NAT box

NAT table

LAN
192.168.10.10 udp 1029
192.168.10.11 udp 1029
192.168.10.11 tcp 1765

Internet
130.104.228.200 udp 3441
130.104.228.200 udp 3442
130.104.228.200 tcp 2343

LAN
Internet
Why some applications don’t work with NATs

Assume A behind a NAT and S in the internet
Communication between A and S must be initiated by A
► « punch a hole through the NAT »
If A and S are both behind a NAT (e.g. with voice over IP), we have a bootstrap problem
► A does not know its IP address as seen by S
► Solving this requires a third party – this is what made Skype’s fortune
When a NAT has a packet to forward and an association exists in the NAT table...

A. The NAT looks for a longest prefix match
B. The NAT looks for an exact match
C. None of the above
D. I don’t know
From WAN to LAN, the NAT may modify...

A. The source port
B. The destination port
C. None of the above
D. I don’t know
5. Network Masks

All machines that are in the same LAN are said to be in the same subnetwork.

The IP addresses of all machines in one subnetwork must have the same prefix (called “network part”).

ex: 128.178.71

The size (in bits) of the network part is not always the same; it must be specified in the machine together with the address; at EPFL-IPv4, size of network part is always 24 bits:
Example: 128.178.71.34 /24

For historical reasons the size of the network part is often specified using a network mask.

Mask = sequence of bits where 1s indicate the position of the prefix. At EPFL-IPv4, network mask is always 1111 1111 1111 1111 1111 1111 0000 0000 which is written in decimal notation as 255.255.255.0;
Example: address = 128.178.71.34, mask = 255.255.255.0
One IP subnet must correspond to one network part

The size of the network part may vary

- EPFL IPv4 network part is 24 bits
- ETHZ IPv4 network part 26 bits
- IPv6 network part is very often 64 bits

128.178.151.24
2001:620:618:1a6:0a00:20ff:fe78:30f9
Same as saying
Mask = ffff:ffff:ffff:ffff::
IPv4 address classes

Long ago, IPv4 addresses had a class subnet mask was not necessary

This is now obsolete...

... but some people continue to use it.
Can Host A have this address?

Masks are all 255.255.255.0

A. Yes
B. No
C. I don’t know
The IPv4 Subnet Mask at ETHZ is ...

A. 255.255.255.0
B. 255.255.255.1
C. 255.255.255.2
D. 255.255.255.192
E. 255.255.255.198
F. 255.255.255.332
G. ffff:ffff:ffff:ffff::
H. ffff:ffff:ffff:c000::
I. I don’t know
The IPv6 Subnet Mask at ETHZ is ...

A. 255.255.255.0
B. 255.255.255.1
C. 255.255.255.2
D. 255.255.255.192
E. 255.255.255.198
F. /48
G. ffff:ffff:ffff:ffff::
H. ffff:ffff:ffff:ffff:c000::
I. I don’t know
What is the subnet broadcast address for subnet 129.132.100.0/26?

A. 129.132.100.0
B. 129.132.100.15
C. 129.132.100.63
D. 129.132.100.192
E. 129.132.100.255
F. I don’t know
Conclusion

The network layer (= IP) is the center piece of communication networks.

IP is built on two principles:
- one IP address per interface and longest prefix match; this allows to compress routing tables by aggregation
- inside subnet, don’t use routers

There are (unfortunately) two versions of IP, IPv4 and IPv6; they are not compatible – interworking requires some tricks (see later).

NATs came as an after-thought and add a different principle than IP unicast (exact match versus longest prefix match) – are widely deployed

NATs hide IP addresses and complicate the operation of networks.