The Network Layer
IPv4 and IPv6

Jean-Yves Le Boudec

2018
Contents

1. The Two Principles of IP Unicast
2. IPv4 addresses
3. IPv6 addresses
4. NATs
5. Subnets and Masks
6. ARP
7. Host configuration
8. IP packet format, HL and TTL

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
Lisa
A.H1

Forwarding table

to output
B.* 2
A.* 0

router R1

Bart
B.D.H2

Forwarding table

to output
A.* 1
B.D.* 2
B.* 3

router R2

Homer
B.C.H2

Forwarding table

to output
A.* 1
B.D.* 1
B.C.* 0

router R3

B.C.H2

Forwarding table

to output
A.* 1
B.* 2

router R4

To: B.C.H2

Lisa sends a packet to B.D.H2.
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 (old version) and IPv6 (current version)
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)

eample 1: 128. 191. 151. 1

eample 2: 129. 192. 152. 2

hexadecimal: 8 hexa digits (one hexa digit = 4 bits)

eample 1: x80 bf 97 01

eample 2: x81 c0 98 02

binary: 32 bits

eample 1: b1000 0000 1011 1111 1001 0111 0000 0001

eample 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 $\leftrightarrow$ hexa is simple: one hexa digit is 4 binary digits

\[
e_{hex} = 1110_{bin} \quad a_{hex} = 1010_{bin} \quad ea_{hex} = b110 1010_{bin}
\]

Mapping binary $\leftrightarrow$ decimal is best done by a calculator

\[
1110 1010_{bin} = 128 + 64 + 32 + 8 + 2 = 234
\]

Special Cases to remember

\[
f_{hex} = 1111_{bin} = 15
\]

\[
ff_{hex} = 11111111_{bin} = 255
\]
Example

- **Modem + PPP**
  128.178.84.133

- **sic500cs**
  128.178.84.130

- **stisun1**
  128.178.84.1

- **EPFL-Backbone**
  128.178.47.3
  128.178.47.5
  128.178.100.12

- **stisun1**
  128.178.100.3

- **switch**
  130.59.x.x

- **ed0-ext**
  128.178.100.12

- **in-inr**
  182.5

- **LRC**
  128.178.156.1
  00:00:0C:02:78:36

- **sw-la-01**
  128.178.182.3

- **disun3**
  128.178.79.9
  08:00:20:20:46:2E

- **LEMA**
  128.178.29.64
  08:00:07:01:a2:a5

- **ETHZ-Backbone**
  129.132.100.12
  129.132.100.27
  129.132.35.1

- **Komsys**
  129.132
  66.46

- **ezci7-ethz-switch**
  129.132

- **sw-zu-03**
  129.132.100.27

- **edi2-sw**
  128.178.100.12

- **edi2-el**
  128.178.29.64
Network Prefix

Network prefixes are used in routing tables

/24 is the prefix length in bits

0/0 means the default route

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</td>
<td>this host (loopback address) for example 127.0.0.1</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>
IPv4 Packet Format

Header 20 bytes (+ options, if any)

payload

Higher layer protocol
1 = ICMP, 6 = TCP, 17 = UDP)
3. IPv6 Addresses

The old version of IP is IPv4. IPv6 is the current (and final) 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
v6 Routing Tables at ed0-swi

IP address of next hop

Routing table 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>
</tbody>
</table>

IP addresses and interface numbers:

- `lrcsun`: 128.178.156.24, 08:00:20:71:0D:D4
- `lrcpc3`: 128.178.156.7, 00:00:C0:B8:C2:8D
- `lrcmac4`: 2001:620:618:1ad:0a00:20ff:fe78:30f9, 08:00:20:78:30:F9
- `sw-1a-01`: 128.178.47.3, 2001:620:618:10a::1/96
- `lrcmac4`: 2001:620:618:1ad:0a00:20ff:fe78:30f9, 08:00:20:78:30:F9
- `ed2-el`: 128.178.29.1, 2001:620:618:1a4::1/128
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.
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>Description</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>Prefix</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</td>
<td>Unique local addresses = private networks (e.g. in IEW)</td>
</tr>
<tr>
<td>For example</td>
<td>cannot be used on the public Internet</td>
</tr>
<tr>
<td>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

![IPv6 Packet Format Diagram]

**Header**: 40 bytes (including options, if any)

**Payload**: 16 bytes

- **Source address (128 bits)**
- **Destination address (128 bits)**

**Example Higher layer protocol**
- 1 = ICMP
- 6 = TCP
- 17 = UDP

We will see the functions of the fields other than the addresses in a following module.
The dotted decimal notation for 0102:ffff is ...

A. 1.2.255.255
B. 16.32.255.255
C. 228.393.255.255
D. I don’t know
In full, the hexadecimal notation «2001::bad:babe» means...

A. 2001:0bad:babe
B. 2001:0000:0000:0000:0000:0000:0bad:babe
C. 2001:0000:0bad:babe
D. 2001:0000:bad:babe
E. None of the above
F. 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 translates LAN IP address to (typically) its WAN address.

Collisions inside LAN are avoided by having NAT modify IP address and port number.

(Strictly speaking this should be called Network Address and Port Translation, NAPT)

Forwarding at NAT is by exact matching from NAT Table.

Implemented by iptables on Linux -- iptables modifies the TCP and IP headers before forwarding ("prerouting") or after ("postrouting")
Creating a NAT table entry: on the fly

Mapping (iAddr, iPort) to (eAddr, ePort) can be done automatically, as shown; but also statically (e.g. to support a web server in LAN)

There are many variants on how mapping is done and how it is used on the return direction (Internet → LAN): full cone, restricted cone, symmetric etc ... see Wikipedia page on NATs
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

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 can be automatic with a third party – this is what made Skype’s fortune or can use static configuration of NATs – this is what made Messenger fail. Protocols such as TURN and STUN are used.
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

Machines in same subnetwork must have same “network part”

Size (in bits) of network part must be specified in the machine together with the address;
At EPFL-IPv4, it is 24 bits; at ETHZ-IPv4 it is 26 bits. For IPv6 it is often 64 bits (but not always).

Size of the network part is often specified using a network mask = sequence of bits where 1s indicate the position of the prefix.
At EPFL-IPv4, network mask is 255.255.255.0;
At ETHZ-IPv4, network mask is 255.255.255.192

in binary 11111111 11111111 11111111 11000000
in dotted decimal 255.255.255.192

Example: address =128.178.71.34, mask =255.255.255.0
Example with IPv6

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.

<table>
<thead>
<tr>
<th>Class</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.0.0.0 to 127.255.255</td>
</tr>
<tr>
<td>B</td>
<td>128.0.0.0 to 191.255.255</td>
</tr>
<tr>
<td>C</td>
<td>192.0.0.0 to 223.255.255</td>
</tr>
<tr>
<td>D</td>
<td>224.0.0.0 to 239.255.255</td>
</tr>
<tr>
<td>E</td>
<td>240.0.0.0 to 247.255.255</td>
</tr>
</tbody>
</table>
Can Host A have this address? A. Yes  
B. No  
C. I don’t know  
Masks are all 255.255.255.0
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
6. MAC Address Resolution

Why?
An IP machine A has a packet to send to a next-hop B. A knows B’s IP address (from routing table); A must find B’s MAC address.

How?
On Ethernet, A sends an address resolution packet on the LAN. All hosts that have the IP address of B (in principle only B) respond with their MAC address.
A has a packet to send to B = 2001:620:618:1a6:1:80b2:f66:1
This address is on the same subnet therefore A sends directly to B and looks for B’s MAC address

1. A sends a Neighbour Solicitation (NS) packet using the Neighbour Discovery Protocol (NDP) containing the question: “who has IP address B ?”. The IP destination address of this packet is a special multicast address (Solicited Node Multicast Address). The MAC address is derived from the multicast IP address. The NS packet is received by all hosts whose IP address has the same 24 bits as B (see next slide).
2. B responds with a Neighbour Advertisement (NA) packet, giving its MAC address. This NA packet is sent by B to A.

3. A reads NA, stores MAC address in its neighbour cache (also called ARP table) and can now send the data to B. The cache is refreshed whenever A receives a packet from B; it may expire after some timeout (e.g. 20 mn of inactivity).

NA, NS packets are carried as ICMPv6 packets, next-header = 58 (0x3a), inside IPv6 packets.
The Solicited Node Multicast Address

SLAAC (and other protocols) use this multicast address - obtained by adding last 24 bits of target IP address to ff02::1:ff00:0/104

A packet with such a destination address is forwarded by layer 2 to all nodes that listen to this multicast address

Only for IPv6 – IPv4 uses broadcast instead

<table>
<thead>
<tr>
<th>Target address</th>
<th>Compressed</th>
<th>Uncompressed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solicited Node multicast address</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uncompressed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Solicited Node multicast address</td>
<td>Uncompressed</td>
<td>ff02:0000:0000:0000:0000:0000:0001:ff66:0001</td>
</tr>
<tr>
<td>Compressed</td>
<td></td>
<td>ff02::1:ff66:1</td>
</tr>
</tbody>
</table>
Look Inside an NDP Neighbour Solicitation Packet

ETHER:  Packet size = 86 bytes
ETHER:  Destination = 33:33:ff:01:00:01
ETHER:  Source = 3c:07:54:3e:ab:f2
ETHER:  Ethertype = 0x86dd
ETHER:
IP:     ----- IP Header -----  
IP:
IP:     Version = 6
IP:     Traffic class = 0x00000000
IP:     .... 0000 00... ...... ...... ...... ...... = Default Differentiated Service Field
IP:     .... .... ...0. ................ .... = No ECN-Capable Transport (ECT)
IP:     .... .... ...0 .... ...... ...... .... = No ECN-CE
IP:     ........ .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
IP:     Payload length = 32
IP:     NextHeader= 58
IP:     Hop limit= 255
IP:     Source address = 2001:620:618:197:1:80b2:97c0:1
IP:     Destination address = ff02::1:ff01:1
IP:
ICMPv6:  ----- ICMPv6 Header -----  
ICMPv6:
ICMPv6:  Type = 135
ICMPv6:  Code=0
ICMPv6:  Checksum = 0xb199 [correct]
ICMPv6:  Reserved = 00000000
MAC Address Resolution with IPv4

Similar, except

the protocol is called ARP (Address Resolution Protocol)
ARP packets are not IP packets but directly in Ethernet frame
with Ethertype =ARP (86dd)

NDP NS /NA are called ARP Request /ARP replies

ARP request is broadcast to all nodes in LAN
Look inside an ARP packet

Ethernet II
  Destination: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)
  Source: 00:03:93:a3:83:3a (Apple_a3:83:3a)
  Type: ARP (0x0806)
  Trailer: 00000000000000000000000000000000...

Address Resolution Protocol (request)
  Hardware type: Ethernet (0x0001)
  Protocol type: IP (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (0x0001)
  Sender MAC address: 00:03:93:a3:83:3a (Apple_a3:83:3a)
  Sender IP address: 129.88.38.135 (129.88.38.135)
  Target MAC address: 00:00:00:00:00:00 (00:00:00_00:00:00)
  Target IP address: 129.88.38.254 (129.88.38.254)
M1 sends a packet to M3 for the first time since last reboot.

A. M1 sends an NS /ARP packet for q.h1
B. M1 sends an NS / ARP packet for p.1
C. None of the above
D. I don’t know
Security Issues with ARP/ NDP

ARP requests / replies may be falsified (ARP spoofing). Attacker will capture all packets intended to B (e.g. man in the middle attack)

Dest IP address = 2001:620:618:1a6:1:80b2::f01:1
Dest MAC address = 08:00:20:71:0d:d4

bogus NA

128.178.15.221
2001:620:618:1a6:1:80b2::f01:1
08:00:20:71:0d:d4

128.178.15.7
2001:620:618:1a6:1:80b2::f01:1
00:00:d0:b3:d2:8d
DHCP Snooping and Dynamic ARP Inspection can prevent ARP spoofing in LANs

*DHCP snooping* = switch/Ethernet concentrator/WiFi base station observes all DHCP traffic and remembers mappings IP addr ↔ MAC addresses
(DHCP is used to automatically configure the IP address at system startup)

*Dynamic ARP inspection*: switch filters all ARP (or NDP) traffic and allows only valid answers – removes broadcasts (IPv4) and multicasts (IPv6)

Such solutions are deployed in enterprise networks, rarely in homes or WiFi access points
Secure NDP (SEND)

What?
Make NDP spoofing impossible by cryptographic means

How?
Host B has public/private key pair.
EUI of B is a CGA (cryptographically assigned address) = hash of B’s public key and IPv6 address prefix (and other fields such as counters). This binds B’s address and its public key.

NA message sent in response to A contains a signature (RSA) computed with B’s private key. This proves that the message is sent by a system that has the private key that corresponds to the address, i.e. B (if the private key is kept secret).

Not widespread yet – requires a strong hash function. Problem is the inability to change the crypto function by patching.
A private/public key system such as RSA has two keys: one public and one private (secret). With RSA, a clear text message can be encrypted with the private key and decrypted with the public key (or vice versa).

Let P be the public key that was used to generate B’s address (using a hash function), and let p be the corresponding private key.

Here, A can verify the NA by applying RSA decryption with the public key P. This proves that the NA was originated by a system that knows the private key p. A can also verify that B’s EUI is derived from the public key P, since the hash algorithm is known and public. This proves to A that the NA was originated by a system that owns B’s EUI.
7. Host Configuration

An IP host needs to be configured on each interface with:
- IP address of this interface
- Mask of this interface
- IP address of default router
- IP address of DNS server

Can be done manually, or automatically with:
- IPv4 → DHCP (Dynamic Host Configuration Protocol)
- IPv6 → DHCP stateful, SLAAC (stateless), DHCP stateless

Same applies to routers connected to a provider:
- IPv4 → PPP
- IPv6 → PPP, DHCP with Prefix Delegation
DHCP with IPv4

Configuration info is kept in central DHCP server, contacted by host when it needs an IP address; is commonly used with IPv4. Also works with IPv6 (with mods – called DHCP stateful).

**Problem:** host cannot contact DHCP server since it is still does not have an IP address;

**Solution:** router implement a “DHCP Relay” function.

Two phase commit to avoid inconsistent reservations.

Limited lifetime - renewals

![Diagram showing DHCP client, relay, and server with messages: Discover, Offer, Request, Ack.](image)
The Point to Point Protocol (PPP)

**Why?**

allocate address automatically over telecom lines
(modem, ADSL)
link is point to point, no MAC address, DHCP not suitable

**How?** Similar to (simpler than) DHCP

PPPv4 for IPv4
PPPv6 for IPv6
Stateless Address Autoconfiguration (SLAAC) = Plug and Play --- IPv6 only

Why invented: avoid configuring DHCP servers
Fully automatic

How it works:

1. host auto-configures a link local address; 64 bit host part obtained by one of these methods:
   - manually assigned e.g. ::1;
   - derived from MAC address (modified EUI - next slide)
   - randomly assigned
   - cryptographically generated address (CGA) – by hashing the public key of the host

2. host performs address duplication test by sending a multicast packet (to solicited node multicast address)

3. host tries to add globally valid addresses by obtaining network prefix from routers if any present;
Host Part derived from MAC address:
MAC@ → EUI (Extended Unique Identifier)

Q: Why is bit 7 flipped?
A: Bit 7 is the “g” bit -- an EUI is defined by IEEE; a locally assigned EUI (i.e not derived from MAC address) has g bit equal to 1; this is inconvenient for IPv6 addresses. IPv6 uses the reverse convention: bit 7 of modified EUI is 1 for EUI derived from globally assigned MAC addresses, and 0 for manually assigned addresses (e.g. 2001:620:618:100::1).
Randomly Assigned Host Part

Privacy concern: MAC address allows tracking a mobile node
Randomly assigned Host Part can be used as alternative
  7th bit of address must be 0
Host randomly computes one tentative host part
  Duplicate test is used to avoid (unlikely) collisions
Has a limited lifetime
Limited lifetime, renewed before expiration

- Deprecation status
  - Successful Duplicate test
  - Preferred timer expires
  - Valid timer expires
  - Deprecated address cannot be used to start new TCP connections
  - Host should obtain a new address
Stateless DHCPv6

Why invented: solve problem left by stateless auto-configuration
DNS server address is not provided to host by stateless auto-configuration

How:
Stateless auto-configuration is performed first
Router response contains a flag = USE STATELESS DHCP
Host sends a query to DHCP server to obtain missing info, such as DNS server address

Why called stateless?
A: DHCP servers does not keep state information

A better solution?
Router Advertisements indicate address of DNS server in an optional RA extension (RFC 6106)
DHCP with Prefix Delegation

Why? A home (or enterprise) IPv6 router R0 is configured by ISP using DHCP. Local devices are autoconfigured from home router using e.g. SLAAC. Home router needs an IPv6 prefix for the entire home network.

How? ISP DHCP server (delegating router) provides to home router not just its IPv6 address but also the network prefix that this router can delegate to its devices. This is called prefix delegation. This prefix may include the prefix of the link from ISP to R0 (RFC 6603).

Compare to IPv4!

Simpsoncom delegates 2004:aaaa:bbbb:cc00/56 to Lisa
Lisa can create many subnets; e.g. 2004:aaaa:bbbb:cc02/64 and 2004:aaaa:bbbb:cc03/64
the subnet 2004:aaaa:bbbb:cc01/64 is excluded from the delegation
For ISP, all of Lisa is one prefix: 2004:aaaa:bbbb:cc00/56
Multiple Addresses per Interface are the Rule with IPv6

A host interface typically has
One or several link local addresses
Plus one or several global unicast addresses

The preference selection algorithm, configured by OS, says which address should be used as source address – see RFC 3484

In contrast, there is usually only one IPv4 address per interface
Identifies an interface inside one machine that has several interfaces – typically visible in Windows machines

Never inside an IP packet

E.g. fe80: :1%2 means: the destination IPv6 address fe80: :1 on interface %2
**Ipconfig example**

Wireless LAN adapter Wireless Network Connection:

- Physical Address: 10-0B-A9-A3-91-08
- DHCP Enabled: Yes
- Autoconfiguration Enabled: Yes
- Link-local IPv6 Address: fe80::945c:d22c:b0e2:a885%16 (Preferred)
- IPv4 Address: 123.255.96.194 (Preferred)
- Subnet Mask: 255.255.252.0
- Lease Obtained: mercredi 25 juillet 2012 09:05:03
- Lease Expires: mercredi 25 juillet 2012 09:35:02
- Default Gateway: 123.255.99.254
- DHCP Server: 10.3.1.12
- DHCPv6 IAID: 386927529
- DHCPv6 Client DUID: 00-01-00-01-16-E8-19-59-F0-DE-F1-BE-ED-EB
- DNS Servers: 202.45.188.37, 137.189.192.3, 137.189.196.3
- NetBIOS over Tcpip: Enabled

IAID = logical number of this interface, assigned by client

Ethernet MAC address

Identifies this host in the DHCP database
IPv4 Link Local Addresses

Some form of autoconfiguration also exists with IPv4

When host boots, if no DHCP and no configuration info available, it picks an IPv4 link local address at random in the 169.254/16 block

Address duplicate test is performed by broadcast

Allows to operate in routerless network («Dentist’s Office», à la AppleTalk) but not in a general setting

Implemented in Windows, not supported by the Linux version we use in the lab
a) When an IPv4 host uses DHCP, which of the following information does it acquire:
A. its IP address;
B. its subnet mask
C. its default gateway address
D. its DNS server address

A. A
B. A, B
C. A, B, C
D. A, B, C, D
E. None of the above
F. I don’t know
b) When an IPv6 host uses SLAAC, the host part is...

A. Mapped from MAC address
B. Randomly chosen
C. Both of the above are possible
D. None of the above are possible
E. I don’t know
c) With SLAAC an IPv6 host has...

A. A link local address and, if a router is present in the subnet, also a global unicast address
B. If a router is present in the subnet a global unicast address and no link-local address
C. None of the above
D. I don’t know
8. IPv6
Header

- **Ver**: Version (6)
- **Traffic class**: Traffic priority
- **Flow label**: Flow identifier
- **Payload length**: Length of payload
- **Next header**: Protocol type of next header
- **Hop limit**: Maximum number of hops

**Source address (128 bits)**

**Destination address (128 bits)**

**Total length**: 40 bytes (header + options if any)

**Example higher layer protocols**:
- 1 = ICMP
- 6 = TCP
- 17 = UDP

### Notes
- The IPv6 header is designed to support various advanced features such as hop-by-hop options, route elements, and authentication.
- IPv6 addresses are 128 bits long, compared to 32 bits in IPv4 addresses, significantly increasing the address space.
- The header is 40 bytes long, including the variable-length options section, allowing for flexibility in network design and management.
- The header uses a fixed-length design, unlike IPv4, which allows for more efficient processing and routing.

---

**Diagram**

- A diagram illustrating the structure of the IPv6 header, showing the various fields and their positions within the header. The diagram highlights the fixed and variable parts of the header, emphasizing the importance of understanding these components for network troubleshooting and protocol analysis.
Hop Limit is called TTL (Time-to-live) in IPv4; however it is not a time but a hop count.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>The version number of IPv4</td>
</tr>
<tr>
<td>IHL</td>
<td>Internet Header Length</td>
</tr>
<tr>
<td>Type of Service</td>
<td>The type of service (e.g., TCP, UDP)</td>
</tr>
<tr>
<td>Total Length</td>
<td>The total length of the packet</td>
</tr>
<tr>
<td>Identification</td>
<td>The identification number</td>
</tr>
<tr>
<td>Flags</td>
<td>Flags for protocol</td>
</tr>
<tr>
<td>Fragment Offset</td>
<td>The offset of the fragment</td>
</tr>
<tr>
<td>Source Address</td>
<td>The address of the source</td>
</tr>
<tr>
<td>Destination Address</td>
<td>The address of the destination</td>
</tr>
<tr>
<td>Time to Live</td>
<td>The time to live</td>
</tr>
<tr>
<td>Protocol</td>
<td>The protocol type</td>
</tr>
<tr>
<td>Header Checksum</td>
<td>The checksum of the header</td>
</tr>
<tr>
<td>Header</td>
<td>The header of the packet</td>
</tr>
<tr>
<td>Payload</td>
<td>The payload of the packet</td>
</tr>
</tbody>
</table>

Header: 20 bytes (+ options, if any)

Higher layer protocol:
1 = ICMP, 6 = TCP, 17 = UDP
Hop Limit (HL) / Time to Live (TTL)

Why? Avoid looping packets in transient loops. If propagation time is small compared to transmission time, a single packet caught in a loop can congest the line.

Transient loops may exist due to non instantaneous changes to routing tables.

How? Every IP packet has a field on 8 bits (from 0 to 255) (called HL for IPv6 / TTL for IPv4) that is decremented at every hop. When it reaches 0, packet is discarded. At source, value is 64 by default.
Traceroute

Sends a series of packets (using UDP) with TTL = 1, 2, 3, ...

**tracert** (windows) similar but uses ICMP

Routers on the path discard packets and send ICMP error message back to source

source learns address of router on the path by looking at source address of error message

Tracing route to www.google.com [2a00:1450:4008:800::1012]
over a maximum of 30 hops:

<table>
<thead>
<tr>
<th>Hop</th>
<th>Time (ms)</th>
<th>Time (ms)</th>
<th>Time (ms)</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>&lt;1</td>
<td>&lt;1</td>
<td>&lt;1</td>
<td>cv-gigado-v100.epfl.ch [2001:620:618:164:1:80b2:6412:1]</td>
</tr>
<tr>
<td>3</td>
<td>&lt;1</td>
<td>&lt;1</td>
<td>&lt;1</td>
<td>c6-ext-v200.epfl.ch [2001:620:618:1c8:1:80b2:c801:1]</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>&lt;1</td>
<td>&lt;1</td>
<td>swiEL2-10GE-3-2.switch.ch [2001:620:0:ffdc::1]</td>
</tr>
<tr>
<td>5</td>
<td>&lt;1</td>
<td>&lt;1</td>
<td>&lt;1</td>
<td>swiLS2-10GE-1-2.switch.ch [2001:620:0:c00c::2]</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>swiEZ1-10GE-2-7.switch.ch [2001:620:0:c03c::2]</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>8</td>
<td>7</td>
<td>swiEZ2-P2.switch.ch [2001:620:0:c0c3::2]</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>swiiX2-P1.switch.ch [2001:620:0:c00a::2]</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>swissix.google.com [2001:7f8:24::4a]</td>
</tr>
<tr>
<td>10</td>
<td>38</td>
<td>34</td>
<td>15</td>
<td>2001:4860::1:0:4ca2</td>
</tr>
<tr>
<td>11</td>
<td>14</td>
<td>14</td>
<td>17</td>
<td>2001:4860::8:0:5038</td>
</tr>
<tr>
<td>12</td>
<td>17</td>
<td>50</td>
<td>17</td>
<td>2001:4860::8:0:8f8e</td>
</tr>
<tr>
<td>13</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>2001:4860::8:0:6400</td>
</tr>
<tr>
<td>14</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>2001:4860::1:0:6e0f</td>
</tr>
<tr>
<td>15</td>
<td>25</td>
<td>24</td>
<td>25</td>
<td>2001:4860:0:1::4b</td>
</tr>
<tr>
<td>16</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>ber01s08-in-x12.1e100.net [2a00:1450:4008:800::1012]</td>
</tr>
</tbody>
</table>
Other fields

Type of service / Traffic Class

- Differentiated Services (6bits) – sort of priority eg voice over IP
  Used only in corporate networks
- Explicit Congestion Notification (2bits) see congestion control

Total length / Payload length

- in bytes including header
  \( \leq 64 \text{ Kbytes}; \) limited in practice by link-level MTU (Maximum Transmission Unit)
- every subnet should forward packets of 576 = 512 + 64 bytes

Protocol / Next Header = identifier of protocol

- 6 = TCP, 17 = UDP
- 1 = ICMP for IPv4, 58 = ICMP for IPv6
- 4 = IPv4; 41 = IPv6 (encapsulation = tunnels)
- 50 = ESP (encrypted payload)
  51 = AH (authentication header)

Checksum

- IPv4 only, protects header against bit errors
- Absent in IPv6 \( \Rightarrow \) layer 2 and router hardware assumed to have efficient error detection
  ICMP is used to carry error messages
Look inside an IPv4 packet

Ethernet II

   Destination: 00:03:93:a3:83:3a (Apple_a3:83:3a)
   Source: 00:10:83:35:34:04 (HEWLETT__35:34:04)
   Type: IP (0x0800)

Internet Protocol, Src Addr: 129.88.38.94 (129.88.38.94), Dst Addr: 129.88.38.241 (129.88.38.241)

   Version: 4
   Header length: 20 bytes
   Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
   Total Length: 1500
   Identification: 0x624d
   Flags: 0x04
   Fragment offset: 0
   Time to live: 64
   Protocol: TCP (0x06)
   Header checksum: 0x82cf (correct)
   Source: 129.88.38.94 (129.88.38.94)
   Destination: 129.88.38.241 (129.88.38.241)
Look inside an IPv6 packet

ETHER:       ----- Ether Header ----- 
ETHER:       Packet 1 arrived at 11:55:22.298 
ETHER:       Packet size =  86 bytes 
ETHER:       Destination = 33:33:ff:01:00:01 
ETHER:       Source = 3c:07:54:3e:ab:f2 
ETHER:       Ethertype = 0x86dd 

IPv6

IP:         ----- IP Header ----- 
IP:         Version = 6 
IP:         Traffic class =0x00000000 
IP:         .... 0000 00. .... .... .... .... .... = Default Differentiated Service Field 
IP:         .... .... .... 0. .... .... .... .... .... = No ECN-Capable Transport (ECT) 
IP:         .... .... .... 0 .... .... .... .... .... = No ECN-CE 
IP:         .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000 
IP:         Payload length =  32 
IP:         NextHeader= 58 
IP:         Hop limit= 255 
IP:         Source address = 2001:620:618:197:1:80b2:97c0:1 
IP:         Destination address = ff02::1:ff01:1 

ICMP for IPv6 (this is an NDP packet used for address resolution) 
solicited node multicast address
A host generates a packet with Hop Limit = 1

A. This packet is invalid
B. This packet will never be forwarded by a bridge nor by a router
C. This packet will never be forwarded by a bridge but may be forwarded by a router
D. This packet will never be forwarded by a router but may be forwarded by a bridge
E. None of the above is true
F. I don’t know
Conclusion

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

IPv4 and IPv6 are not compatible – interworking requires tricks

NATs came as an after-thought and are widely deployed

ARP/NDP finds the MAC address corresponding to an IP address

DHCP is used allocates IP address, network mask and DNS server’s IP address to a host

SLAAC automatically allocates IPv6 addresses without DHCP

TTL/HL limits the number of hops of an IP packet