Poland, Kraków, AGH-UST: a 6-slide intro

Sławomir Zieliński, PhD
slawek@agh.edu.pl

Who am I

- I started to work at AGH University in 1997 (during my 3rd year of study)
- I was graduated (MSc) in 2000
- I do research and teaching in subjects related to computer networking
- I'm the first Cisco Networking Academy instructor in Poland
  - I've got two Instructor Excellence Advanced Level awards,
  - (more importantly) together with Łukasz Czekierda we provide CCNP trainings for future Cisco employees (30+ yearly)
- I finished my PhD in 2009
  - it was about dynamic deployment of peer-to-peer networks

About Kraków & Poland

- Poland: somewhere north-east from Spain (a bit more to the east)
  - more precisely: about 3250km drive
- 7 neighbour countries
- 38,5 mln inhabitants (#34)
- GDP per capita PPP (IMF 2016): #43 (27 764$)
- Kraków?
  Choose a colour, please…

About Kraków & Poland

- Kraków (50N, 20E):
- former capital city of Poland
  (formally until 1795, in fact until 1596)
- no, there are no white bears in the streets
  (in fact, there are no bears in the streets at all)
- 760.000 citizens (growing)
- 31 universities; 180.000 students (growing)

About AGH University

- The university was established in 1919
  - there are about 2000 professors,
    and about 39.000 students (22% of Kraków students)
  - there are 16 faculties (15 of them technical)

- We’ve been constantly ranked #4-6
  in Poland overall…
- … and we’ve been constantly ranked #1 in IT

the networking lab is here

My goals

- To present and discuss the way we teach people
  - I'm going to present a sample lecture
- (maybe) To present and discuss the list of courses offered to our students
- (maybe) To present and discuss the main project (Małopolska Educational Cloud) I've been working on
- To encourage some of you to visit my country, city and university
  - I personally would like to know (and please be bold) what you know/think about Poland, etc.

Internet Protocol version 6

addressing
protocol changes
accompanying mechanisms

Topics for today

(1) IPv4 deficiencies
(2) IPv6 addressing
(3) neighbor discovery
(4) autoconfiguration
(5) tunnelling
(6) IPv6 packet structure
(7) IPv6 protocol optimizations
(8) Mobile IPv6

Sources

- **RFC 2460: Internet Protocol, Version 6 (IPv6) Specification**
- RFC 3587: IPv6 Global Unicast Address Format
- RFC 4291: IPv6 Addressing Architecture
- RFC 3177: IAB/IESG Recommendations on IPv6 Address Allocations to Sites
- RFC 3879: Deprecating Site Local Addresses
- RFC 3697: IPv6 Flow Label Specification
- RFC 2675: IPv6 Jumbograms
- **RFC 4294: IPv6 Node Requirements**
- RFC 3484: Default Address Selection for IPv6
- RFC 4311: Host-to-Router Load Sharing
- RFC 2991: Multipath Issues in Unicast and Multicast Next-Hop Selection
- RFC 2464: Transmission of IPv6 Packets over Ethernet Networks
- **RFC 4861: Neighbor Discovery for IPv6**
- RFC 2462: IPv6 Stateless Address Autoconfiguration
- RFC 4191: Router Preferences and More-Specific Routes
- RFC 4311: IPv6 Host-to-Router Load Sharing
- RFC 4443: Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6)
- www.cisco.com: Implementing IPv6 for Cisco IOS Software
- cisco.customerelearning.com: IPv6

IPv4 address space exhaustion

- What techniques are (or were) used to conserve IPv4 address space?
  - VLSM (Variable Length Subnet Masks),
  - CIDR (Classless Inter-Domain Routing),
  - ~~DHCP (Dynamic Host Configuration Protocol),~~ xDSL was here
  - NAT (Network Address Translation) with overloading and private address space

- If there were no IPv4 address space conservation mechanisms in place, it would have been exhausted in 1995(!)

Hamachi – what's that?

IPv4 deficiencies

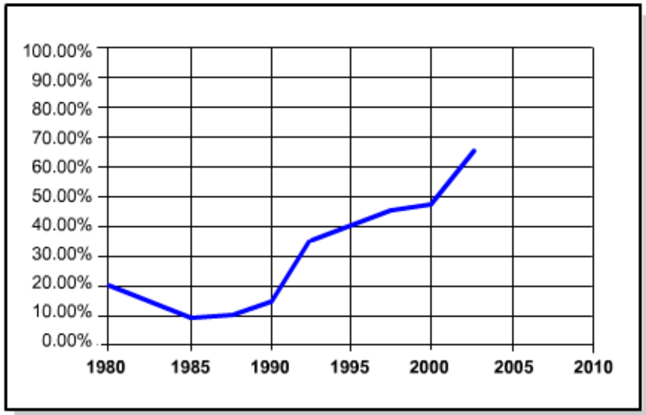
- Was IPv4 designed for „Internet” or „internet”?

IPv4 does not address the following:
- security: privacy, confidentiality, …
- device autoconfiguration,
- diversity of connected devices ← ???

IPv4 PDU hardware processing is not easy:
- routers sometimes need to slice (fragment) packets,
- the definition of IPv4 is de facto fixed
  (e.g., it is hard to add new options)

The decision: the protocol is going to be redesigned

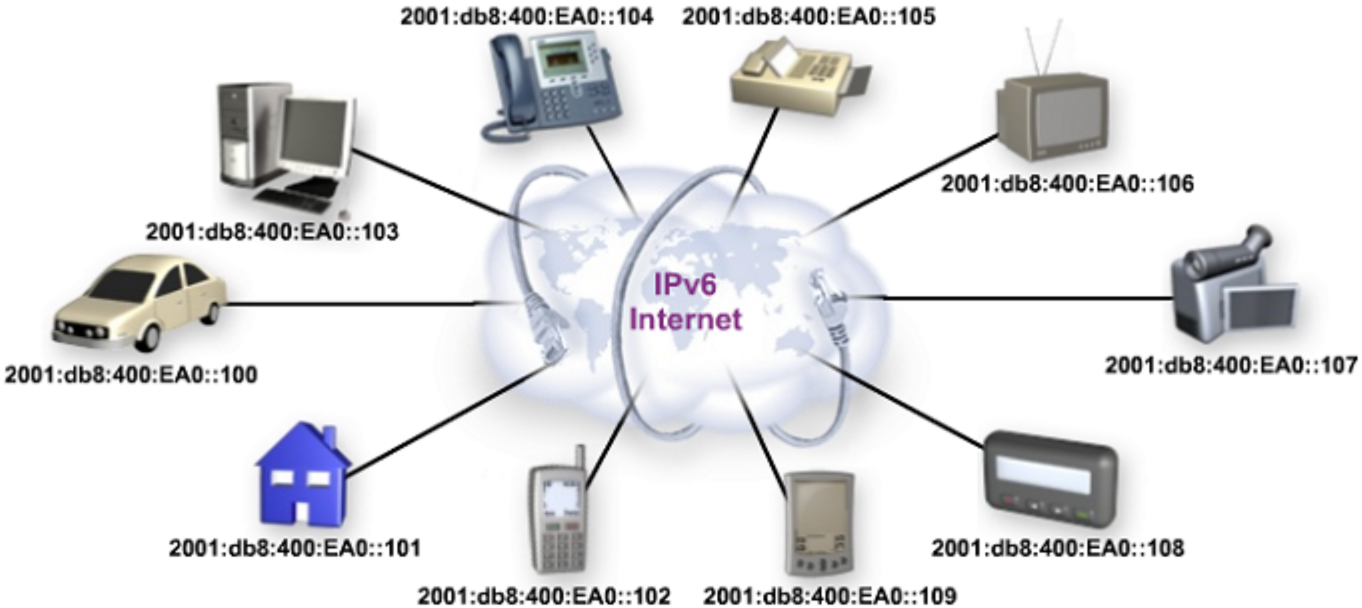
Brain reset



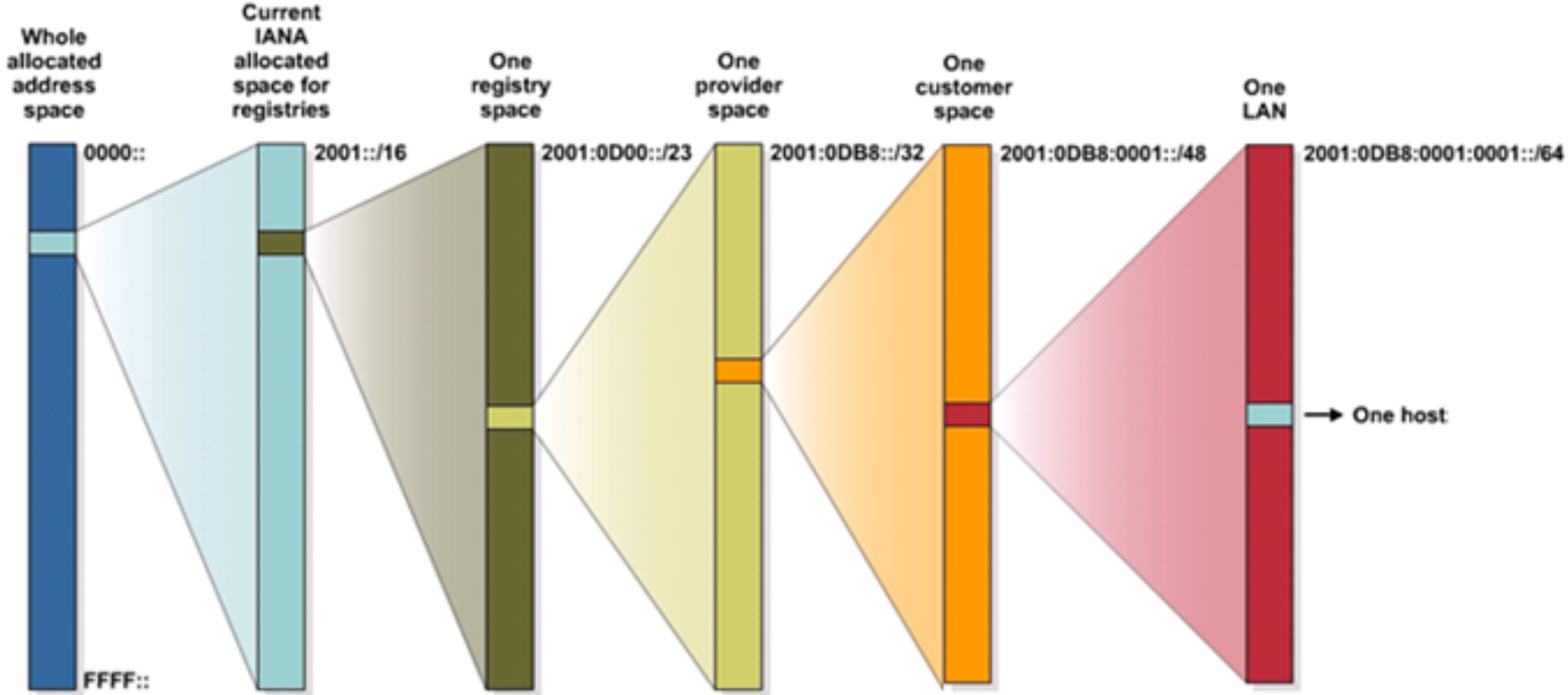
- Which of the following protocols is used to conserve IPv4 address space?
  - CIDR
  - DHCP
  - NAT
  - ARIN
- What options are implemented in IPv4?

Larger address space virtues

- Better reachability of devices
- Ability to use arbitrarily chosen L4 protocol
- Ability to configure direct secured link (without intermediary devices)

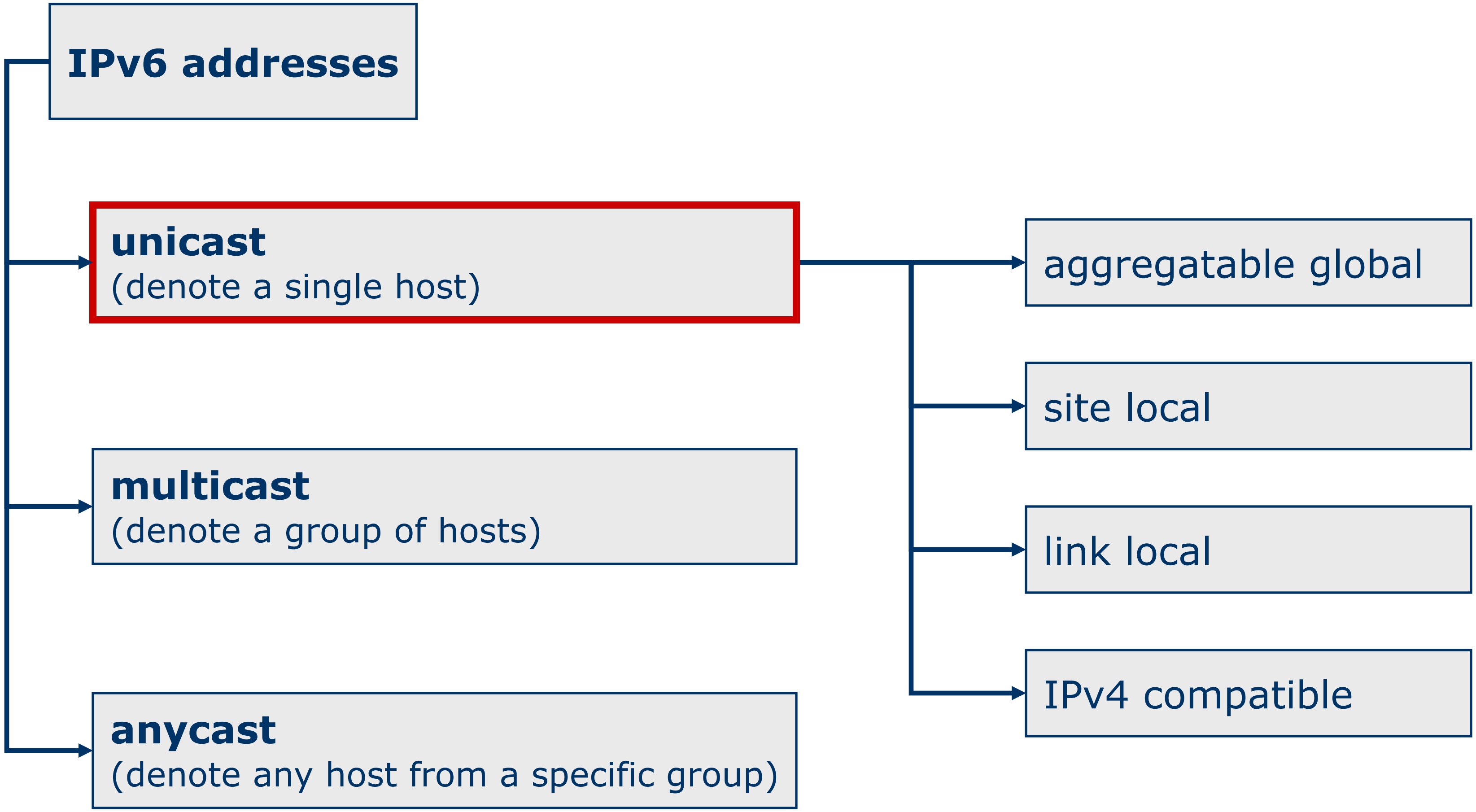


IPv6 address assignment



Archimedes' estimation of the number of sand grains in the universe:
hai myriakismyriostas periodou myriakismyrioston arithmon myriai myriades,
means about **10^{63}** … so – the IPv6 addressing space is insufficient,
because **2^{128}** is a little above **$3,4*10^{38}$** ☹

IPv6 address space



IPv6 address

- Full notation:
  hhhh:hhhh:hhhh:hhhh:hhhh:hhhh:hhhh:hhhh

1	2	3	4	5	6	7	8
64 bits: network				64 bits: host			

  – 128 bits → $2^{128}=3,4*10^{38}$ possible addresses

- Prefix notation: <prefix-hex>/**<len-dec>**
  – example: 2001:0dbd:8000:6561::**/32**

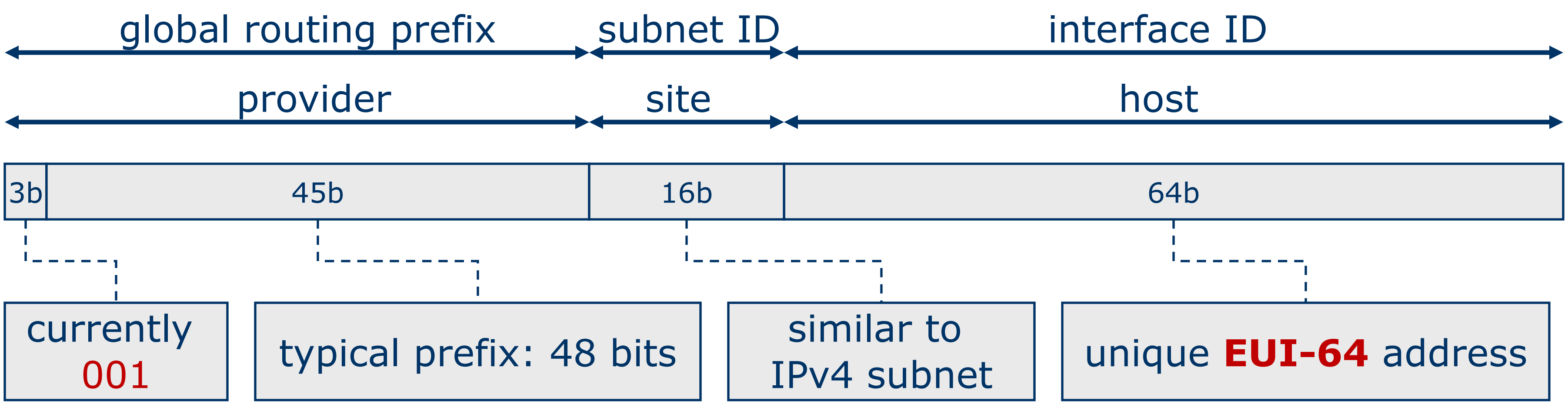
IPv6 address notation

- Full notation:
  2001:0db8:0000:0000:0000:0000:0800:012a
- Without leading zeros:
  2001:db8:0:0:0:0:800:12a
- Abbreviated notation (with ::):
  2001:db8::800:12a
- Erroneous notation:

  2031::130F::9C0:876A:130B

Globally unique addresses

- Structure:



  - SLA = site level aggregator (RFC 2374) → now „subnet ID"
  - EUI = extended universal identifier (IEEE standard)

- Addresses beginning with 2000::/16 – E000::/16 (currently only 2000::/16, 001xxxxx in binary) are assigned by IANA (Internet Assigned Numbers Authority)

Site local addresses

- Functionally equal to IPv4 private (RFC 1918) addresses
  - routers cannot forward packets with source/destination addresses of that pool outside a certain domain
  - prefix: FEC0::/10 (1111 1110 11)
  - although it ispossible to use 54 bits for subnet address, it is recommended to use only 16 – the same bits that are used for subnet address in global addresses
  - the format:

1111 1110 11	0	subnet	interface ID

- **DEPRECATED**: RFC 3879
- RFC 4193: Unique Local IPv6 Unicast Addresses FC00::/8
  - … no comments …

Link local addresses

- Used for autoconfiguration and neighbor discovery
  - local nodes do not need global addresses; *link local* addresses are enough
  - link-local means „no router between”
  - prefix: FE80::/10 (1111 1110 10)
  - format:

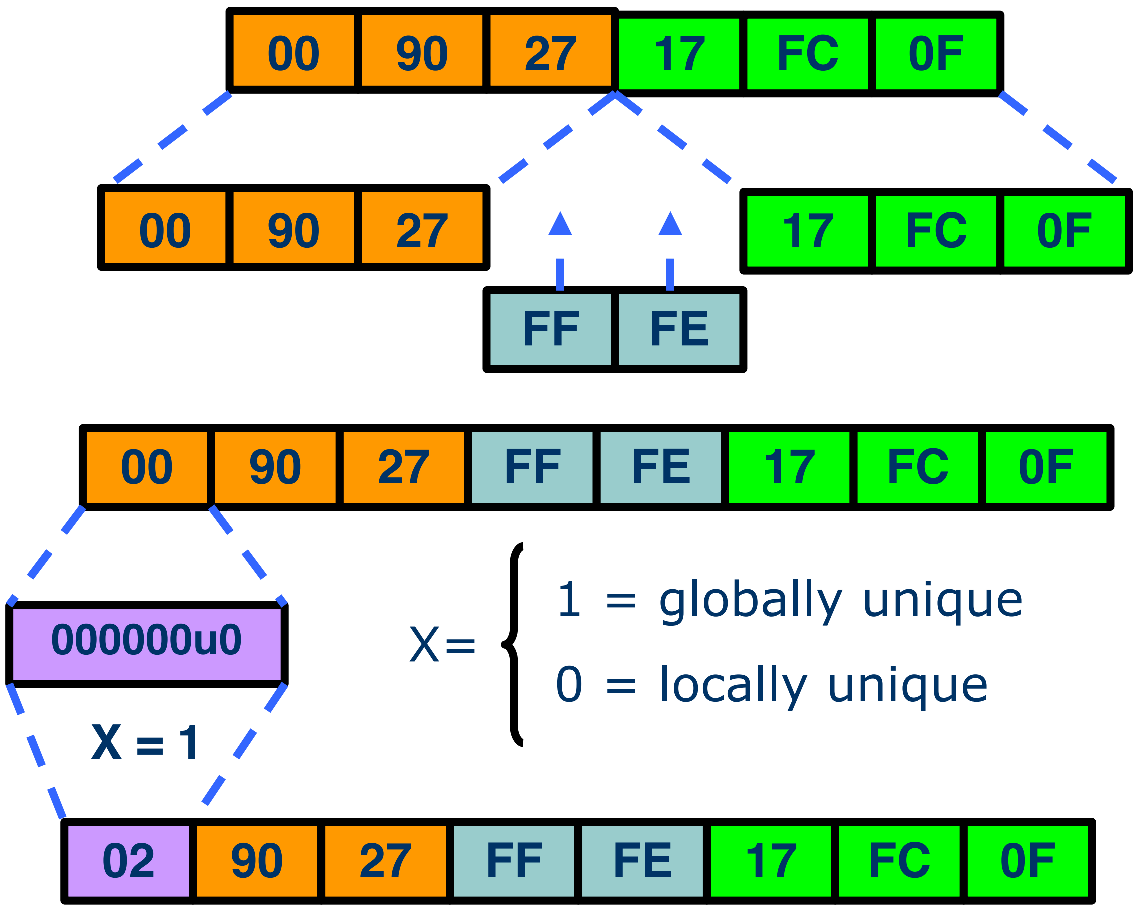
1111 1110 10	0	0	interface ID

  - interface IDs are created in various ways, depending on L2 technology
    - Ethernet – based on MAC
    - ISDN – based on E.164
    - ...

Interface ID generation methods

- Autoconfiguration based on 64-bit L2 address (EUI-64)
- Autoconfiguration based on 48-bit L2 address (MAC)
- Autoconfiguration using DHCP
- Manual configuration
- Autoconfiguration based on pseudo-random number
- Autoconfiguration based on cryptographic methods (CGA = cryptographically generated address)
- …

Ethernet interface ID



Result: **EUI-64** (extended universal identifier)

Current address space usage

- Addresses are assigned from 1/8 of the overall address space
- RFC 6890: Special-Purpose IP Address Registries
  - ::/128 unspecified
  - ::1/128 loopback
  - 64:ff9b::/96 for IPv4 – IPv6 (RFC 6052) translators
  - ::ffff:0:0/96 IPv4 mapped address
  - 100::/64 discard-only (for RTBH, RFC 3882, RFC 5635)
  - 2001:db8::/32 documentation
  - 2002::/16 6to4 tunnels
  - …

Current address space usage

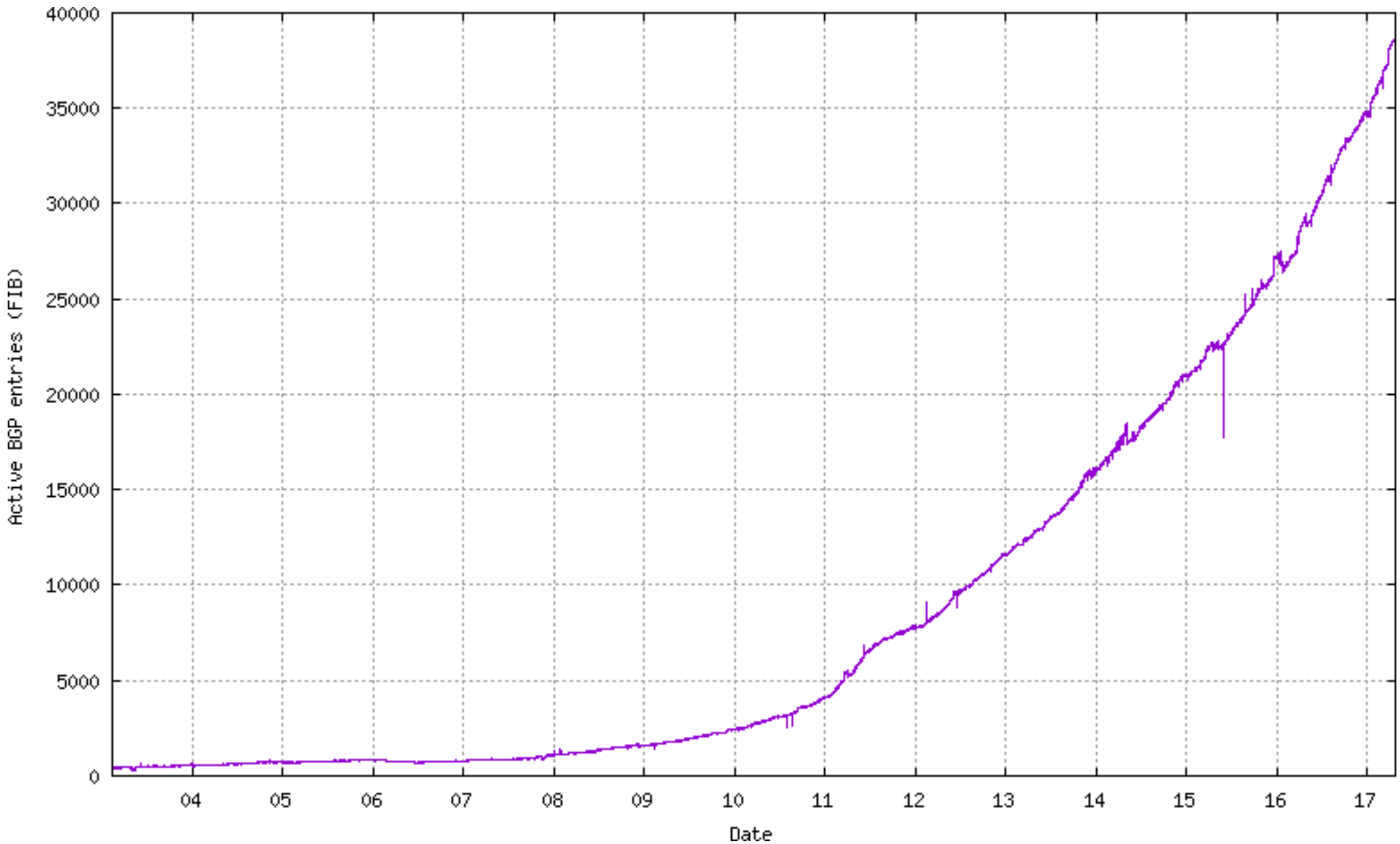
- bgp.potaroo.net (24.04.2017):
  - 987003 IPv6 users in Poland, i.e., **3,53%** of Polish Internet users…
  - other countries:
    Russia: 1,26%,
    Romania: 8,45%,
    Czech Rep.: 10,2%,
    Switzerland: 34,9%,
    Belgium: 55,4%

- What about Spain? and Portugal?

… well: 0,57% (217572), and 26,6% (1840308)

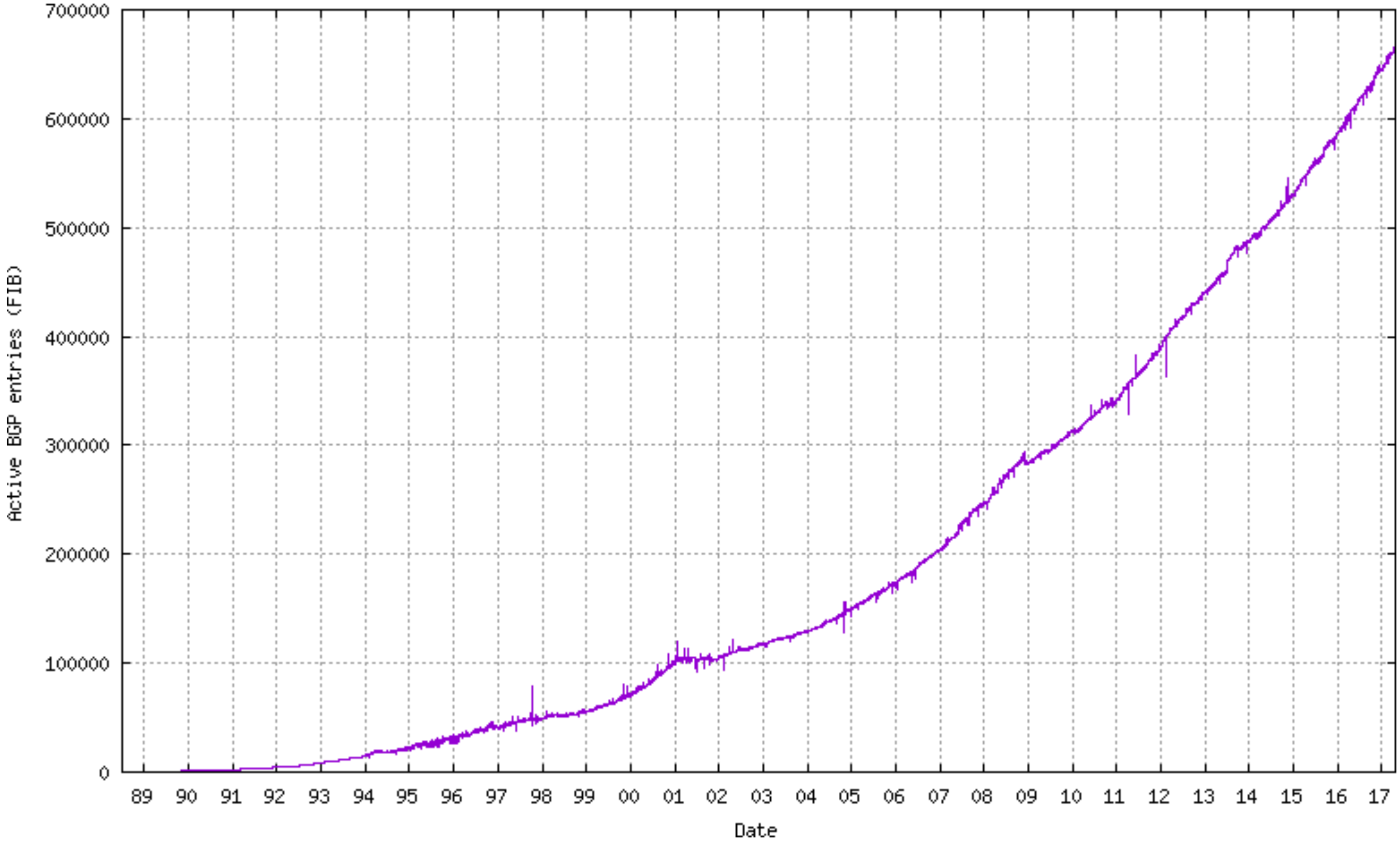
This means: **you've got homework to do**…

Globally routed IPv6 prefixes



Source: http://bgp.potaroo.net/v6/as2.0/index.html

Globally routed IPv4 prefixes



Source: http://bgp.potaroo.net/as2.0/bgp-active.html

Load balancing

- Host can use multiple routers
  - RFC 2461: round-robin
    - problem #1: no route characteristics are taken into account
    - problem #2: synchronization
- Recommended reading: RFC 4311, RFC 2991

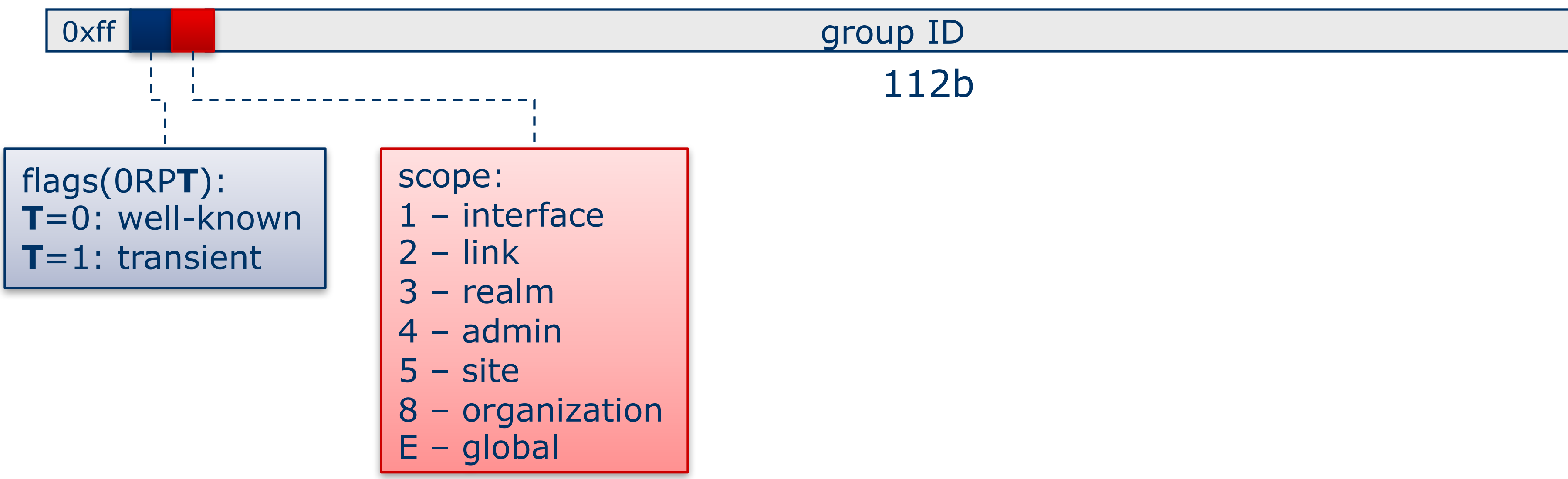
IPv6 address space

IPv6 addresses

- **unicast**
  (single host)
- **multicast**
  (group of hosts)
- **anycast**
  (any host from a specific group)

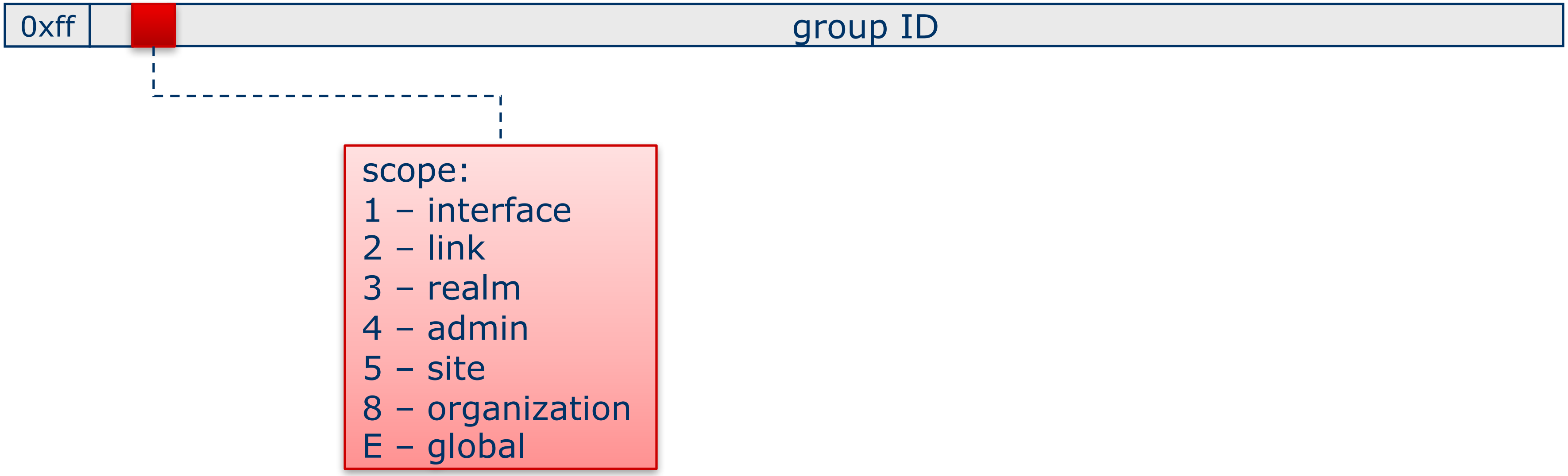
Multicast

- Denotes a group of interfaces
  - typically the interfaces belong to different hosts
  - format:
    - prefix: FF00::/8 (1111 1111)
    - lifetime (4b), scope (4b)



Example:
multicast from the FF02::/16 range is well-known (assigned by IANA), link-local

Multicast addresses



- Scopes:
  - interface-local: something like multicast loopback
  - link-local: inside local LAN (not forwarded by routers)
  - site-local: inside a domain
  - admin-local: something in between link and site local
  - organization-local: inside an organization
  - 0: silently dropped by any device
  - F: reserved, but in practice equivalent to E (global)

Multicast addresses ctd.

- Some permanent (well-known) addresses:
  - „all hosts in the LAN” ff02::1
  - „all routers in the LAN” ff02::2
  - „all RIPng routers” ff02::9
  - …

- The meaning (and application) of some multicast addresses does not depend on their scope, e.g.:
  - FF01::101 – „all NTP servers on the same interface”
  - FF02::101 – „all NTP servers on the same link”
  - FF05::101 – „all NTP servers in the same site”
  - ...

- Temporary (transient) addresses meaning can be different in different networks

Reserved multicast addresses

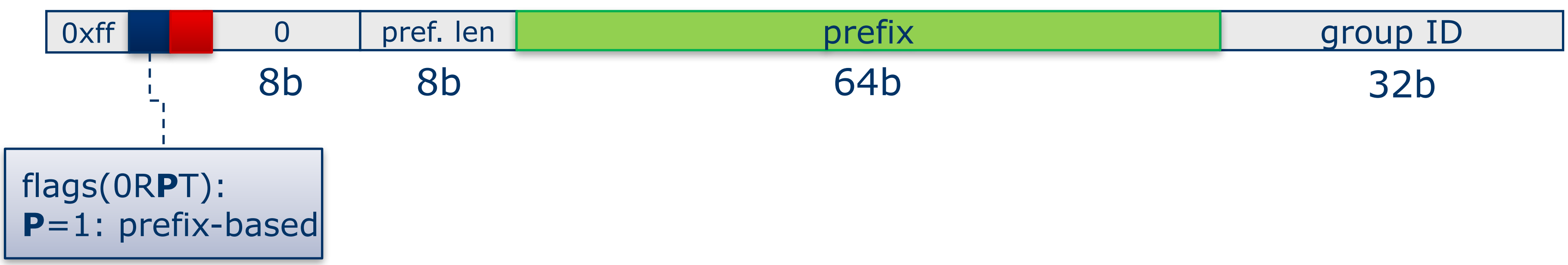
- ff0*::/16 – don’t even try to use them (RESERVED) ☺
- Each node is obliged to receive the following multicasts:
  - ff01::1, ff02::1 – all nodes (interface local & link local),
  - ff02::1:ffxx:xxxx/104 (solicited node multicast address)
    - substitute ‘x’ with 24 least significant bits of any node address
    - example: ff02::1:FF24:2424
      is a solicited node multicast for 2025:01::3624:2424
- Routers have also to listen for
  ff01::2, ff02::2, ff05::2 (all-routers multicast)

„What you have” addresses

- GLOP – what is that?
  - according to uncle Google:
    „sticky and amorphous matter, typically something unpleasant”

- Correct answer: a range of multicast addresses defined by the AS number (e.g., AS 3172 → 233.12.100.0/24)

- IPv6 has a similar concept, prefix-based multicast



- There are more multicast address types to find ☺ …

IPv6 address space

IPv6 addresses

- **unicast**
  (yes, yes, single interface…)
- **multicast**
  (groups, we've just covered that, didn't we?)
- **anycast**
  (any member of a group)

… or any instance of a specific service …

Anycast address

- Syntactically identical to unicasts
- Identifies a set of interfaces
  - the interfaces typically belong to different hosts
- Application:
  - identification, e.g., of a set of database servers,
  - identification of a time synchronization service,
  - …
- IP packet destined for an anycast is forwarded to the nearest interface of the group
  - the distance is defined by routing protocols

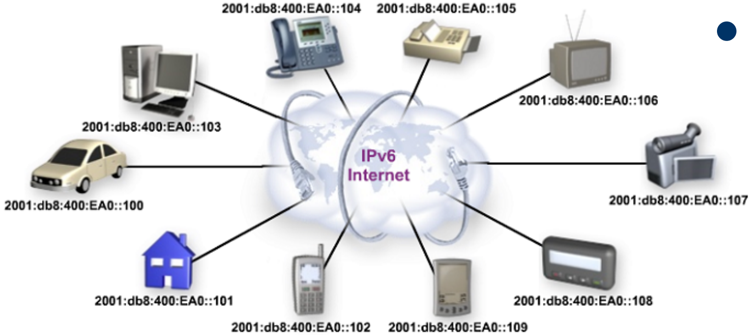
Anycast address

- Anycast cannot be used as a source address (!) for session initiation
  - typical configuration: anycast is configured on a loopback interface, so it is not the first-choice address for new sessions
- Example: subnet-router anycast (e.g., `2001:db8:1:1::/64`)
  - interface ID is „0"
- An interesting application: anycast sinkholes
  - a sinkhole is a place that receives all unwelcome/suspected traffic and drops/analyzes it
  - in a large network the forwarded traffic consumes a significant amount of bandwidth ➔ we need more sinkholes
  - anycast sinkhole simplifies the configuration, because multiple sinkholes can be identified with a single address
- Other applications: DNS, NTP, syslog, RADIUS, Kerberos, …

Anycast addresses

- Problem: how can a host announce its anycast address?
- Solution: using a dynamic routing protocol
- Effect: the host becomes a router
- RFC 4294 definitions:
  - **IPv6 Node**: a device that implements IPv6
  - **IPv6 Router**: a node that forwards IPv6 packets not explicitly addressed to itself
  - **IPv6 Host**: any node that is not a router

Brain reset



- How many times are IPv6 addresses longer than IPv4 ones?
  - 2
  - 4
  - 6
  - 8
- How can a host obtain a *link-local* address?

IPv6 Neighbor Discovery, SLAAC

IPv6 Neighbor Discovery

- Potential uses:
  - IPv6 node discovery
  - e.g., in case of router „disappearance” hosts actively search for another
  - L2 address discovery

- The functionality is similar to:
  - ARP
  - ICMP (subset)

- ND is based on L2 multicasts

Solicited node multicast

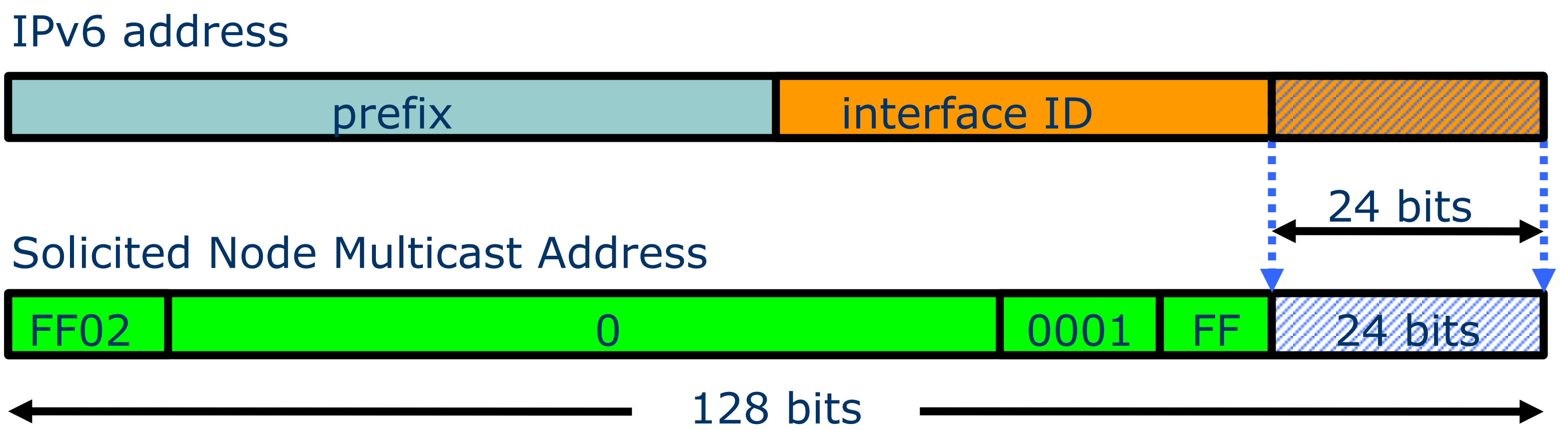
- Every node needs to belong to the following multicast groups:
  - ff01::1, ff02::1 – all nodes multicast,
  - ff02::1:ffxx:xxxx/104
    solicited node multicast
    (replace 'x' with 24 least significant bits of any unicast or anycast address served by the device)

- example: if a host has an address fe80::be:deaf

  it needs to listen on multicast ff02::1:ffbe:deaf

- What MAC address will be used for that IPv6 multicast?

IPv6 Neighbor Discovery (ARP?)



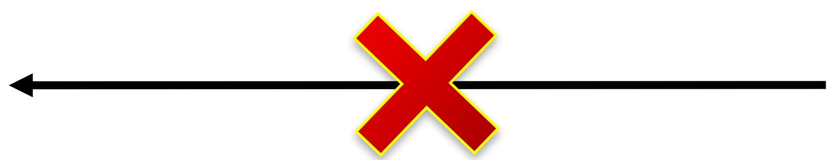
ICMPv6 type = 135
Src = A Dst = B (solicited node multicast)
Data = A's L2 address
Query = what is your L2 address?

ICMPv6 type = 136
Src = B Dst = A
Data = B's L2 address

After the sequence
we can start data transfer

IPv6 ND – DAD (gratuitous ARP?)

ICMP type = 135
Src = 0 (::)
Dst = *A* - solicited node multicast
Data = A's L2 address
Query = what is your L2 address?



No response is good news …

IPv6 Router Discovery (IRDP?)

Router Advertisement:

ICMPv6 type = 134

Src = router's address (link-local)

Dst = all-nodes multicast (FF02::1)

Data= options, **network prefix**, validity, flags

Routers send out their advertisements periodically.

A host does not need to wait for an advertisement
(it can send out „router solicitation").

What will be the destination address of a router solicitation message?

Autokonfiguracja

- Autokonfiguracja bezstanowa (*stateless*)
  - podstawa: duża przestrzeń adresowa pozwala na zrealizowanie mechanizmu „plug and play” dla hostów IPv6 przypisującego adresy IP z zachowaniem globalnej jednoznaczności
  - algorytm działania:
    - ogłoszenie routera zawiera m.in. 64-bitowy prefiks sieci
    - host dopełnia prefiks własnym 64-bitowym identyfikatorem
    - taki proces jest szczególnie użyteczny w przypadku urządzeń mobilnych
  - można w ten sposób dość łatwo przeadresować sieć
    - wystarczy rozesłanie nowego prefiksu przez router
  - jeśli host nie znajdzie routera, próbuje znaleźć serwer DHCP
    - FF02::1:2 „all DHCP agents”
    - FF05::1:3 „all DHCP servers”

Autokonfiguracja - rozszerzenia

- RFC 4941: Privacy Extensions for Stateless Address Autoconfiguration in IPv6
  - problem: generowane adresy IPv6 są niezmienne, co ułatwia pracę podsłuchiwaczom…
  - sposób rozwiązania: zmieniać adresy, tzn. skomplikować procedurę ich automatycznego tworzenia i utrudnić identyfikację urządzenia
  - składowe rozwiązania: pseudorandom bazujący również na globalnym prefiksie + MD5

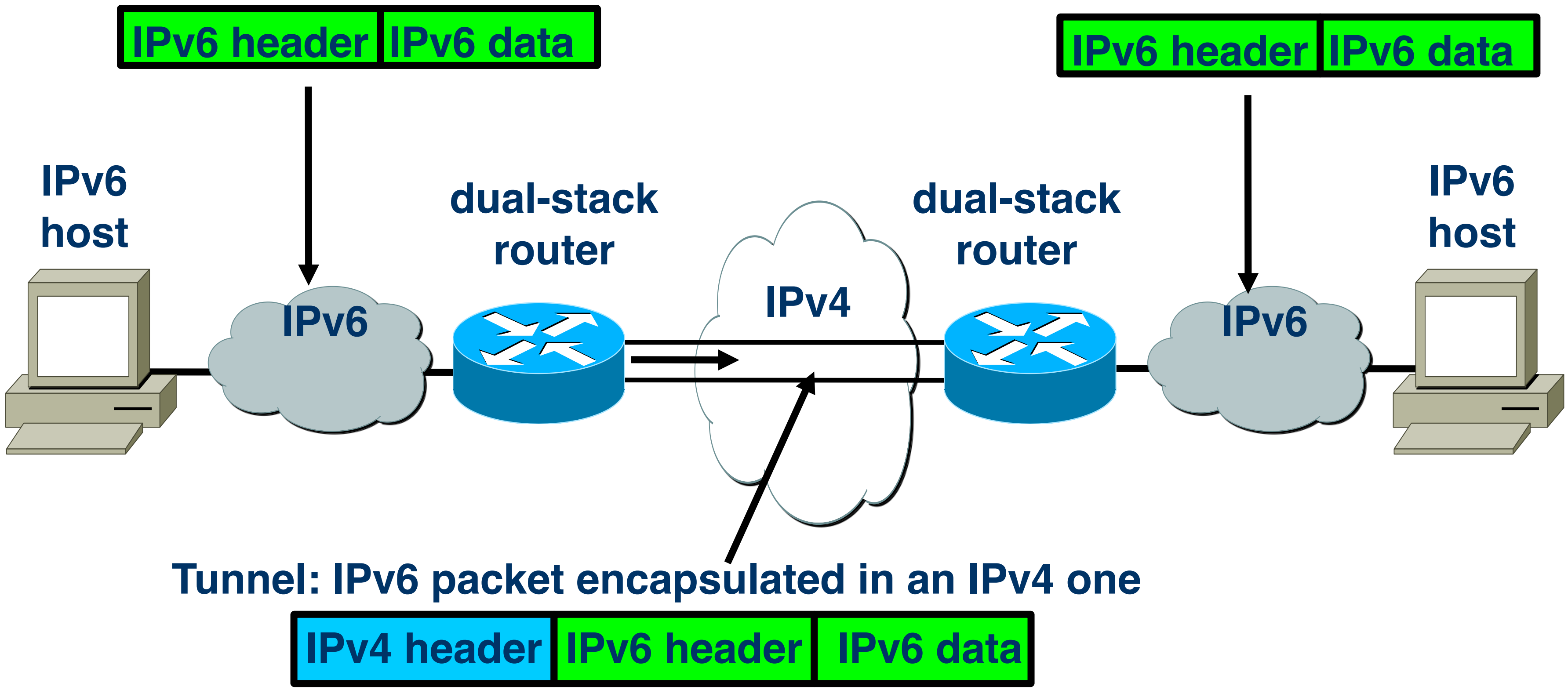
- Ale… po co podsłuchiwać? Scenariusz:
  1. host Telefon jest używany w sieci karczmy Rzym…
  2. właściciel karczmy, niejaki Dzierżymord, instaluje tam co nieco, i zapamiętuje dolne 64 bity adresu IPv6 hosta Telefon
  3. host Telefon wędruje do sieci BardzoWażnejFirmy (BWF) - ma nowy prefiks (znany w okolicy), ale „dół” mu się nie zmienia
  4. Dzierżymord zdalnie aktywuje co nieco i kontaktuje się z szefostwem BWF w celu uzyskania nieuprawnionych korzyści finansowych…

IPv4 & IPv6 coexistence

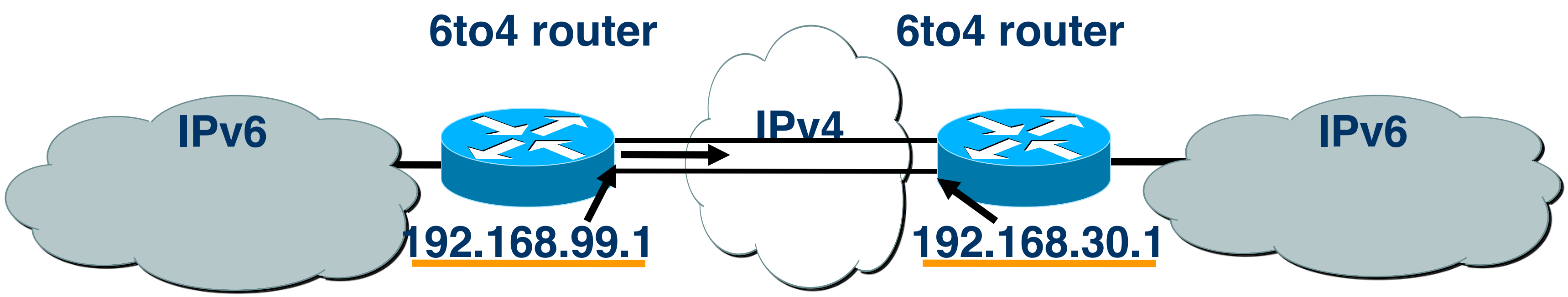
IPv4 ←→ IPv6

- There is no transition date set ☺
- There are quite a few mechanisms for IPv6 ←→ IPv4 communication
  - IPv6 islands and tunnels
  - IPv6 to/from IPv4 protocol translators
  - (hopefully) IPv4 islands …
- The basic element: a dual-stack system

„6in4” tunnels



„6to4" tunnels



IPv6 network prefix:

2002:c0a8:6301::/48

IPv6 network prefix:

2002:c0a8:1e01::/48

- Uses a dedicated, reserved address space (2002::/16)
- 32 bits of prefix contain IPv4 address of the destination router

Translators

- Allow IPv6 to/from IPv4 communication
- They do not require any additional configuration on the hosts
- Problems (header): path MTU, fragmentation
- Example technologies:
  - NAT-PT, NAPT-PT
    RFC 2766; deprecated in RFC 4966,
  - NAT64 (RFC 6146) – stateful (!)
  - SIIT (RFC 6145,6791) stateless translation of IP/ICMP
  - 464XLAT (RFC 6877) IPv4 islands over IPv6 „sea?” communication

IPv6 Packet

[IPv4] Problem #1: fragmentation

- Fragmentation can occur multiple times
  - it can result in <u>really small</u> fragments
  - each of the fragments is a separate IP packet
    → checksum, header, routing…

- IPv6: Hosts are obliged to implement *path MTU discovery*

- IPv6: minimum MTU - 1280 (IPv4: 68), recommended - 1500
  - if the link layer does not support such MTU,
    it needs to provide fragmentation & de-fragmentation mechanisms

[IPv4] Problem #2: QoS

- Basic metrics: bandwidth, delay, jitter, packet loss
- Typical approaches: best effort, IntServ, DiffServ
  - (except the first) they need data stream identification
- A stream is identified with (parts of) information from L3 **and L4** (!)
- Each of the routers processes the packets independently
  - no signalization protocol,
  - separate identification of streams → more CPU power needed

[IPv4] Problem #3: Options

- There is a way to define a proprietary option…
- … but there is also an obligation to process each of the options on every L3 forwarding device…
- Side effect: variable header length

IPv6 packet

- Basic assumption: we need to simplify processing
  - 64-bit alignment,
  - no header recalculation on routers

IPv6 header

- Simplified (compared to IPv4)

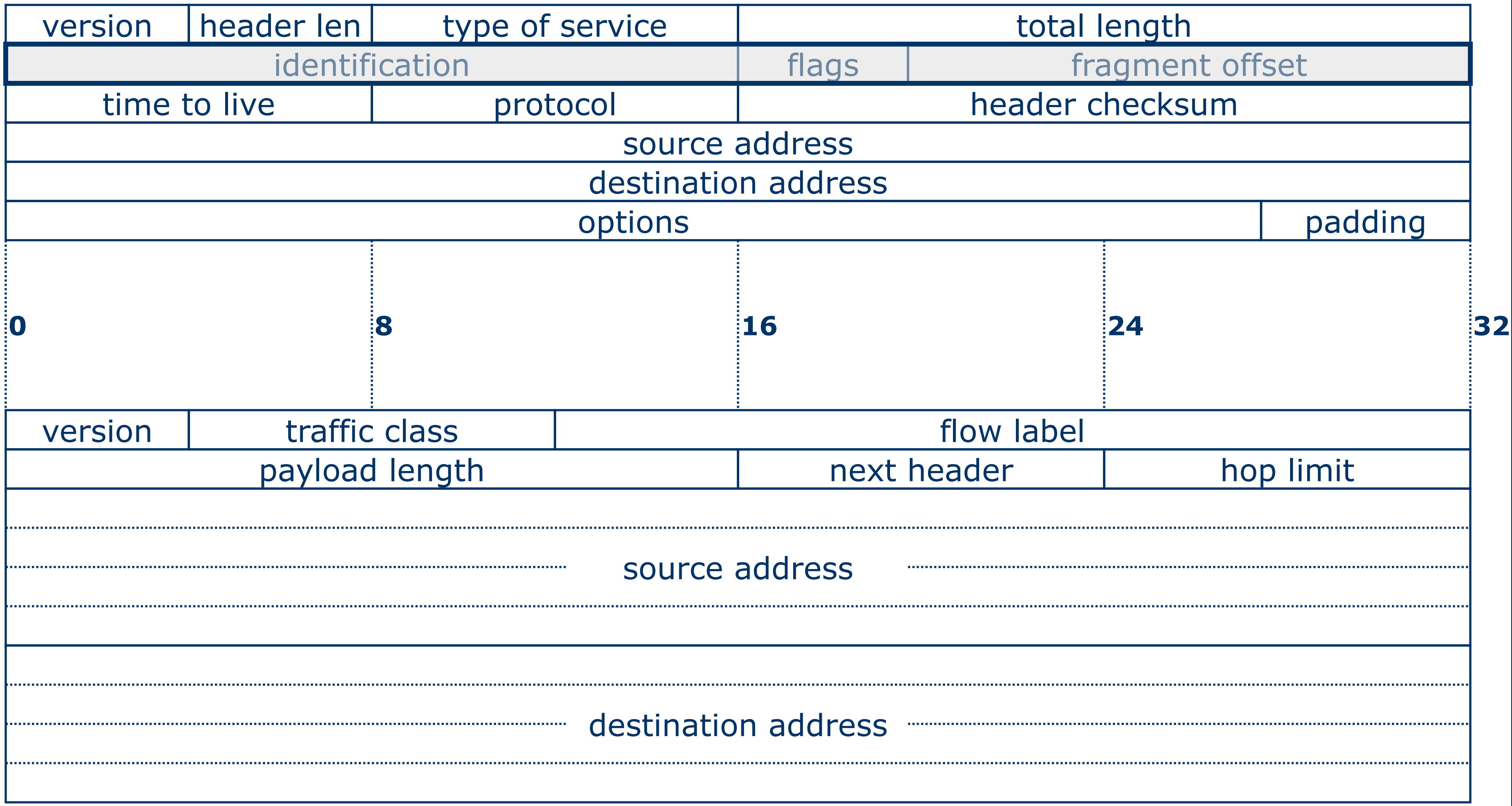
version	header len	type of service	total length
identification	flags	fragment offset	
time to live	protocol	header checksum	
source address			
destination address			
options	padding		

0 8 16 24 32

version	traffic class	flow label
payload length	next header	hop limit
source address		
destination address		

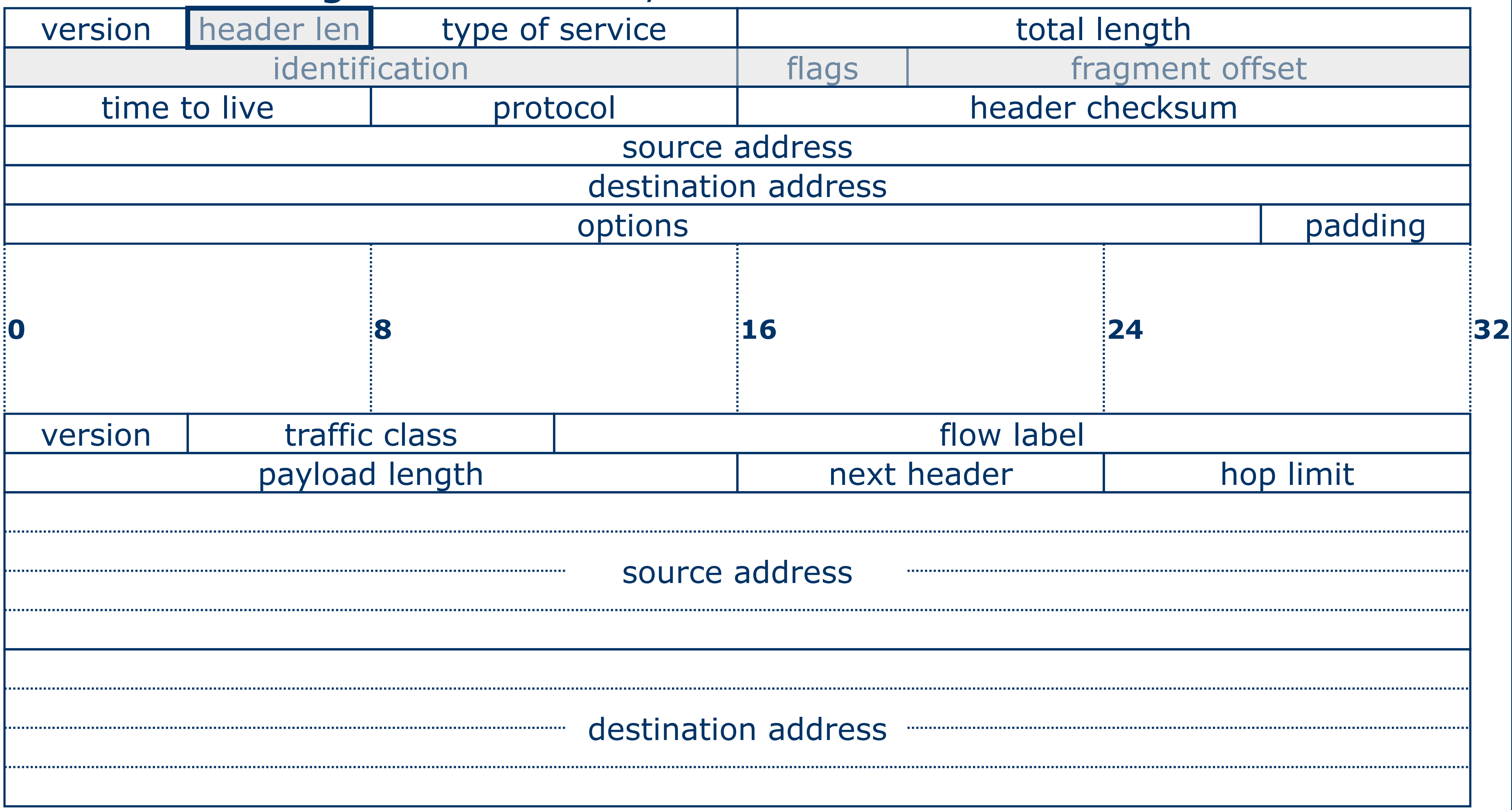
IPv6 header

- Fragmentation – possible, but only at the sender node



IPv6 header

- Header length is constant, 40 octets



IPv6 header

- Options (if any) are placed in extension headers

<table>
<tr><td colspan="4">version</td><td colspan="4">header len</td><td colspan="8">type of service</td><td colspan="16">total length</td></tr>
<tr><td colspan="16">identification</td><td colspan="3">flags</td><td colspan="13">fragment offset</td></tr>
<tr><td colspan="8">time to live</td><td colspan="8">protocol</td><td colspan="16">header checksum</td></tr>
<tr><td colspan="32">source address</td></tr>
<tr><td colspan="32">destination address</td></tr>
<tr><td colspan="28">options</td><td colspan="4">padding</td></tr>
</table>

0 8 16 24 32

<table>
<tr><td colspan="4">version</td><td colspan="8">traffic class</td><td colspan="20">flow label</td></tr>
<tr><td colspan="16">payload length</td><td colspan="8">next header</td><td colspan="8">hop limit</td></tr>
<tr><td colspan="32">source address</td></tr>
<tr><td colspan="32">destination address</td></tr>
</table>

IPv6 header

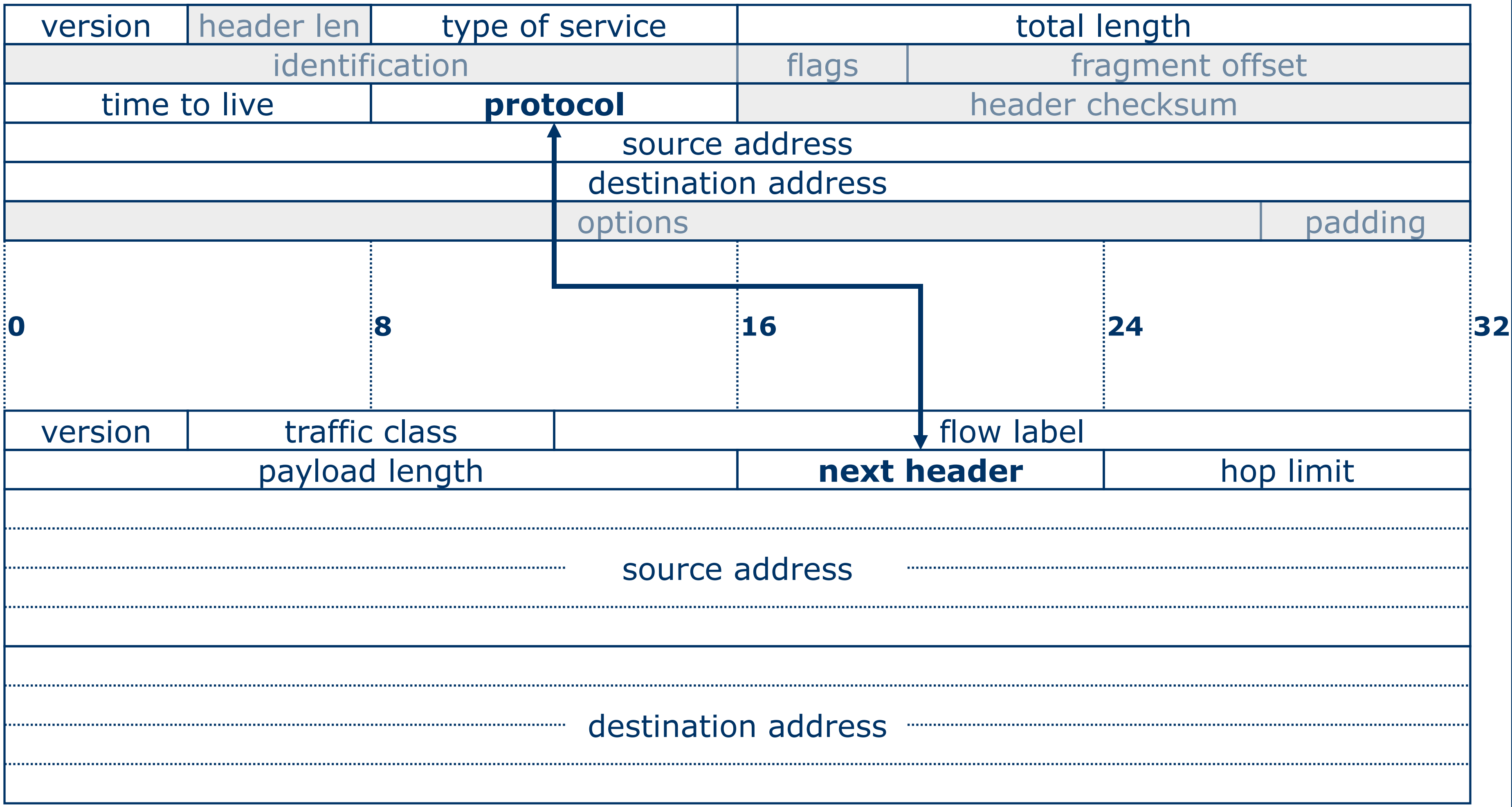
- There is no header checksum

0	8	16	24	32
version / header len	type of service	total length		
identification		flags / fragment offset		
time to live	protocol	header checksum		
source address				
destination address				
options			padding	

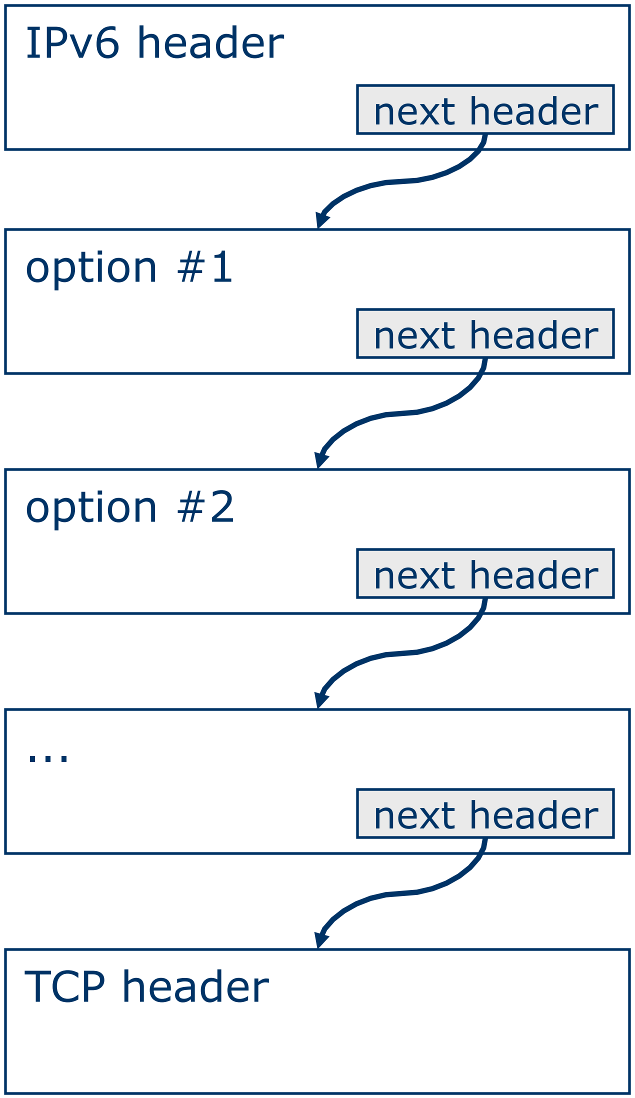
0	8	16	24	32
version / traffic class	traffic class / flow label	flow label		
payload length		next header	hop limit	
source address				
destination address				

IPv6 header

- Some fields have similar (or identical) meaning



IPv6 packet options



- „next header” identifies either an additional option header or a L4 header
- In the unlikely case of IPv6/IP6 tunneling, the field can point even to an additional IPv6 header

IPv6 options

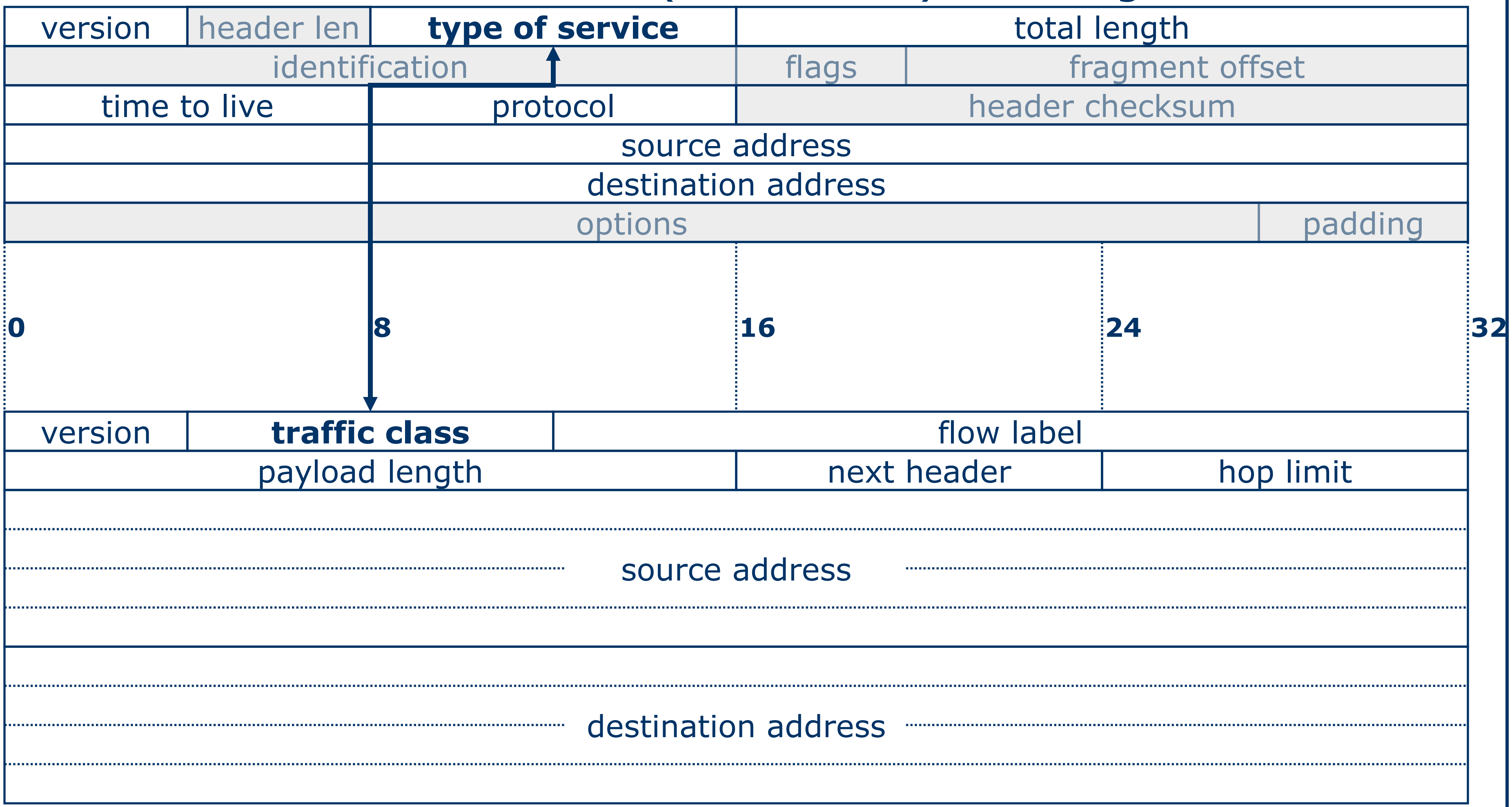
- Currenty a few options are standardized:
  - Hop-by-Hop Options
  - Routing
  - Fragment
  - Destination Options
  - Authentication RFC 2402
  - Encapsulating Security Payload RFC 2406 **IPSec**
  - Mobility Header

- Most of them are not processed by routers
  - exceptions: Hop-by-Hop Options Header, Routing Header

- The sequence of options is also standardized

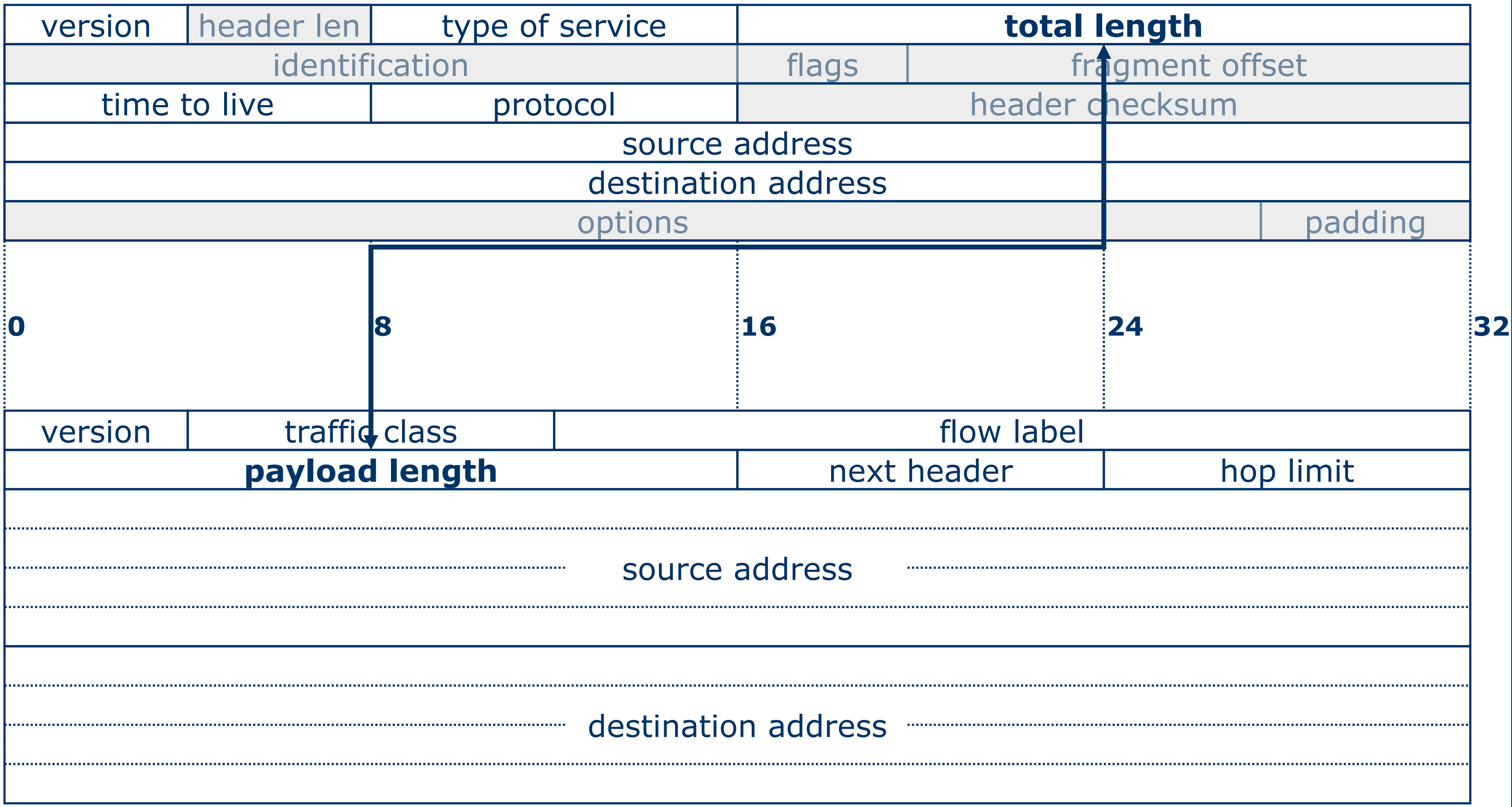
IPv6 header

- Some fields have similar (or identical) meaning



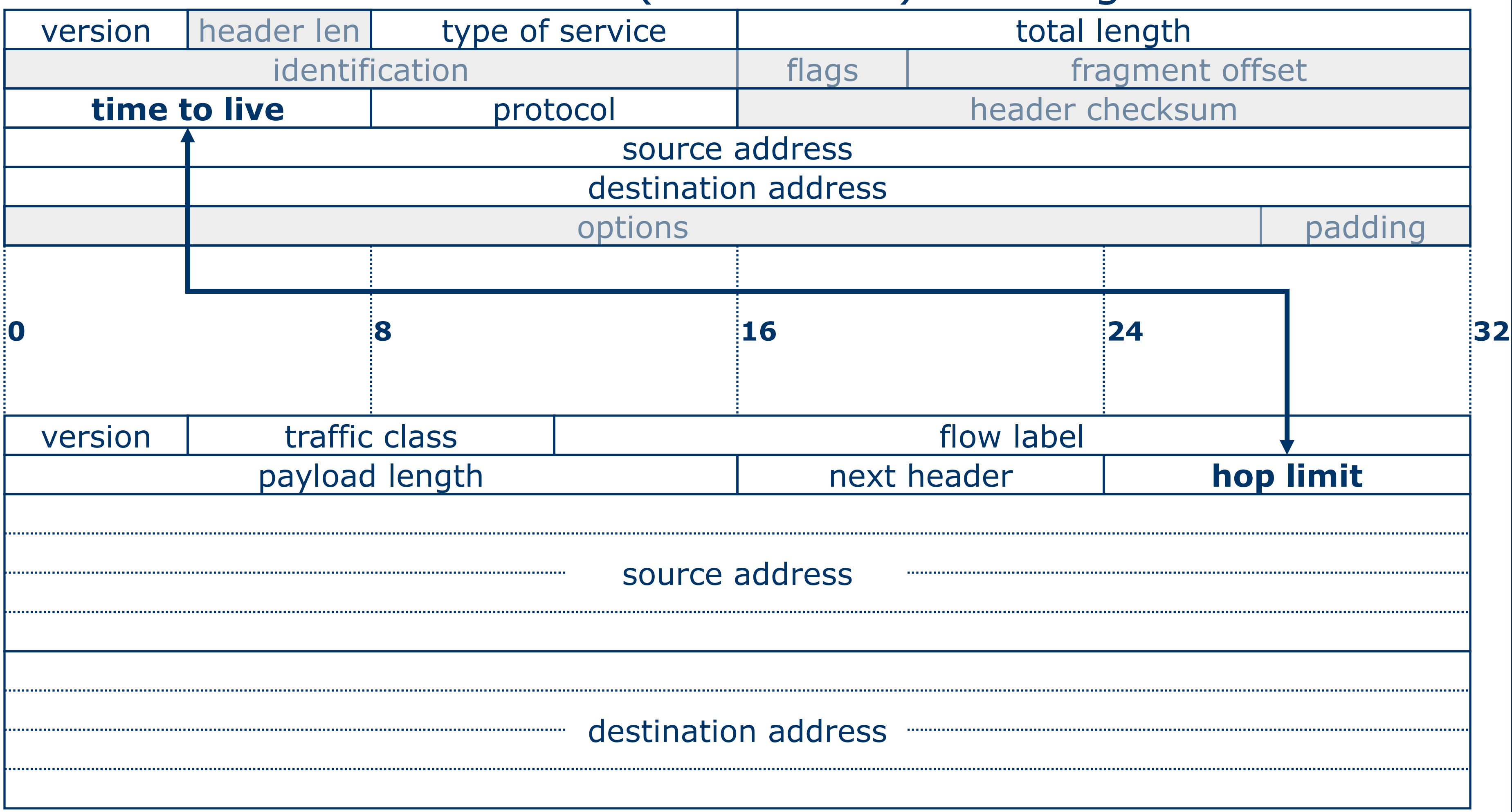
IPv6 header

- Some fields have similar (or identical) meaning



IPv6 header

- Some fields have similar (or identical) meaning



IPv6 header

0	8	16	24	32
version	traffic class	**flow label**		
payload length		next header	hop limit	
source address				
destination address				

- *traffic flow*: a sequence of packets forwarded from a specific source to a specific destination
  - IPv4 routers identify flows based on IP addresses, L4 protocol number and port numbers
    - problems: fragmentation, encryption
  - in case of IPv6 such approach would be even more complicated; we do not have an *all-headers-length* field

IPv6 header

0	8	16	24 32
version	traffic class	**flow label**	
payload length		next header	hop limit
source address			
destination address			

- *flow label* is used by traffic sources to mark the distinct flows
  - IPv6 identifies flows based only on L3 layer information
  - no need to analyze or even know L4 PDU
  - we finally can implement and deploy our own L4 protocol

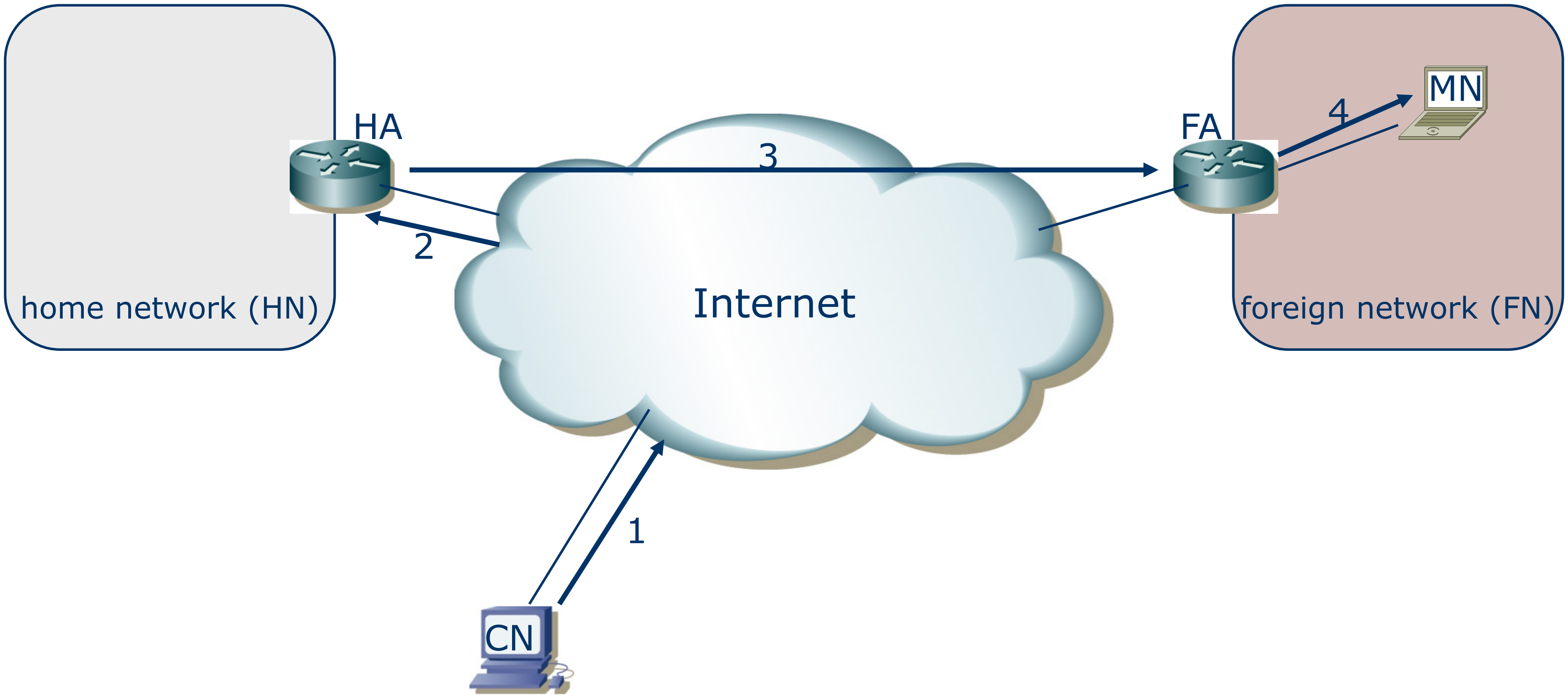
- Source hosts SHOULD assign new labels to the flows
- Routers SHOULD NOT modify the labels

IPv6 Mobility

… only key concepts, really …

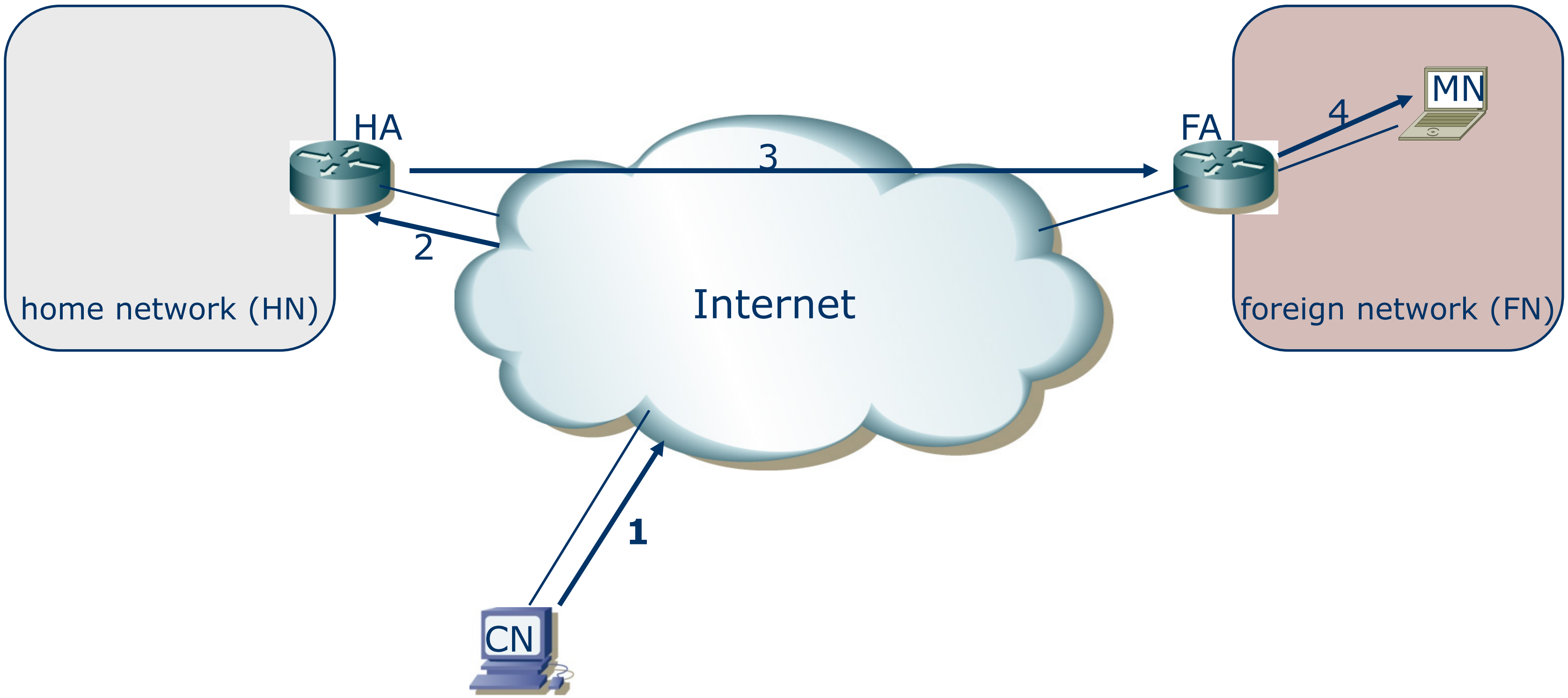
- no agent discovery,
- no registration,
- no hierarchical mobility,
- no …

Mobile IPv4 - sketched



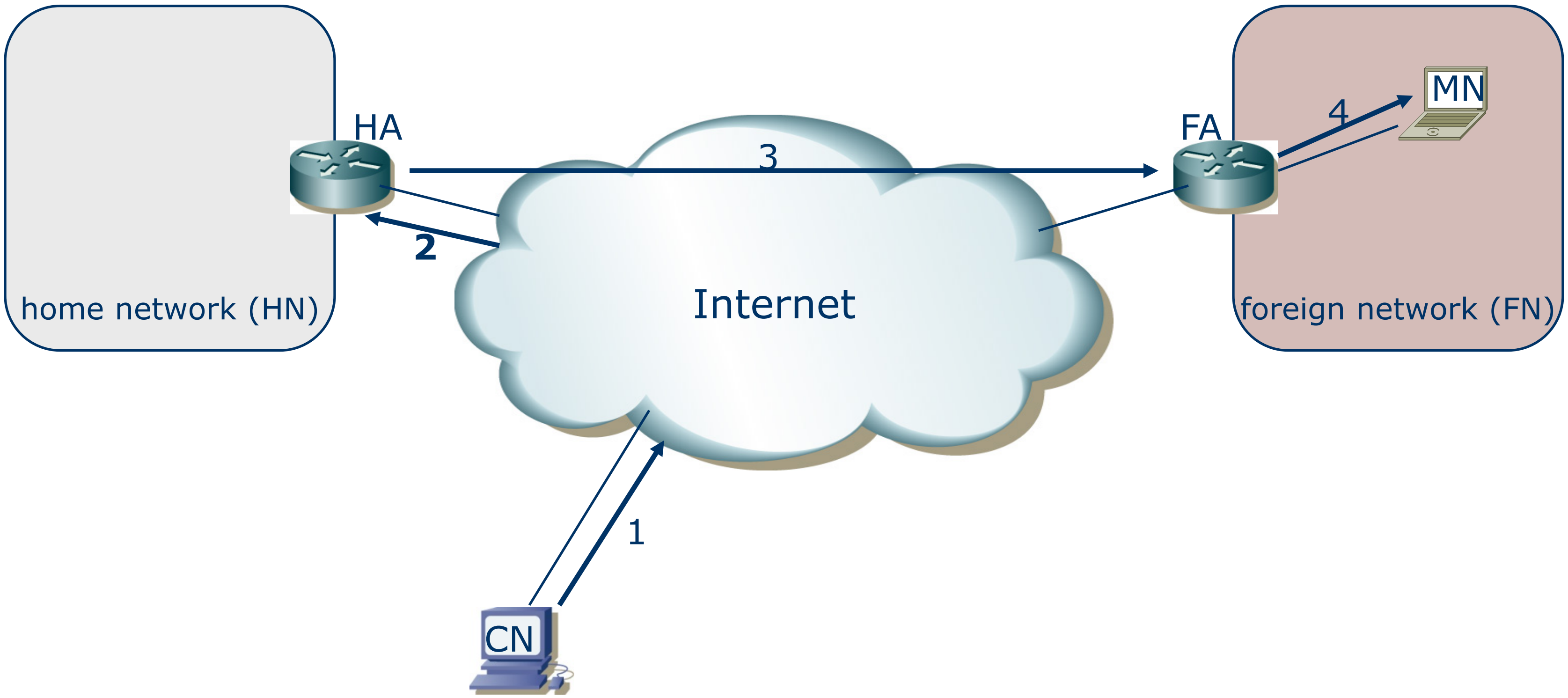
1. CN sends a packet to MN
2. the packet is intercepted by HA
3. HA forwards it to FA
4. FA forwards it to MN

1. CN sends a packet to MN



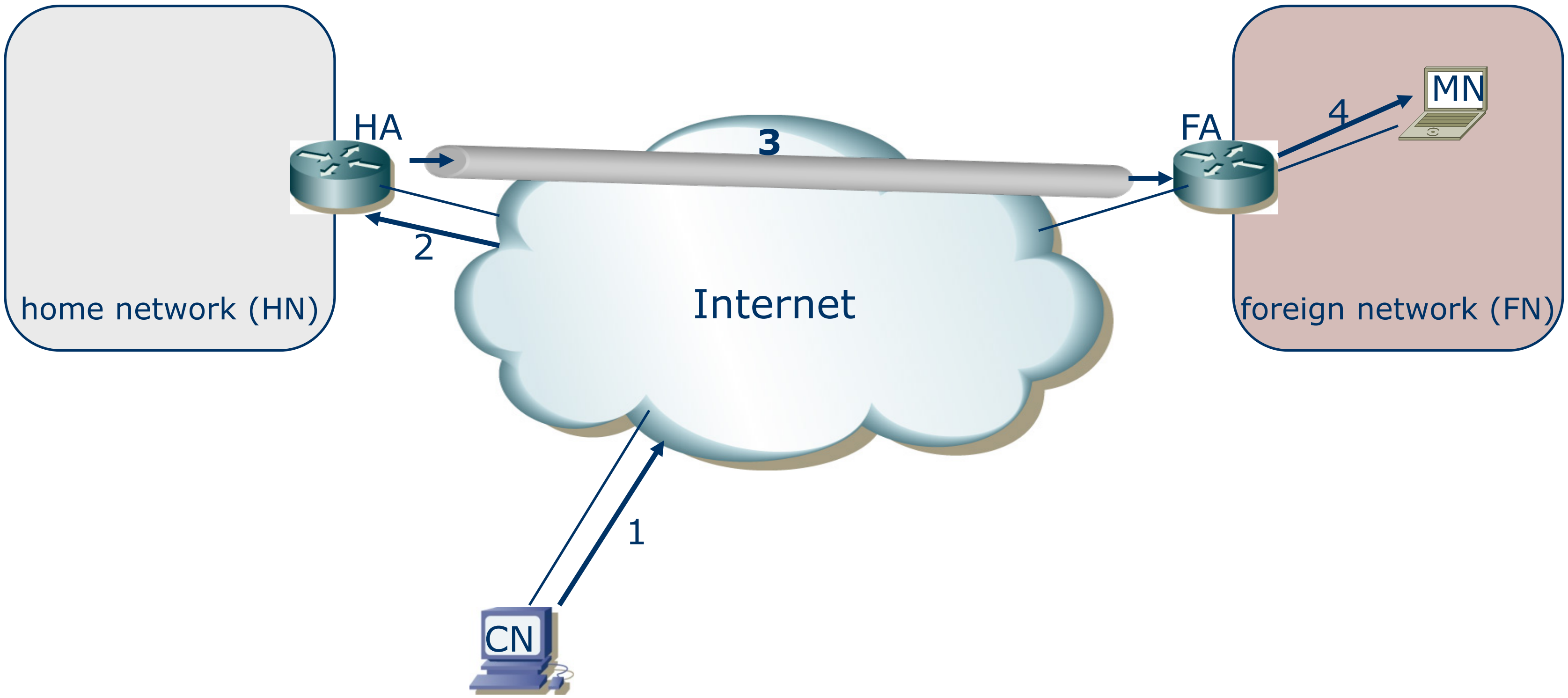
source IP: CN destination IP: MN	data

2. The packet is intercepted by HA



source IP: CN destination IP: MN	data

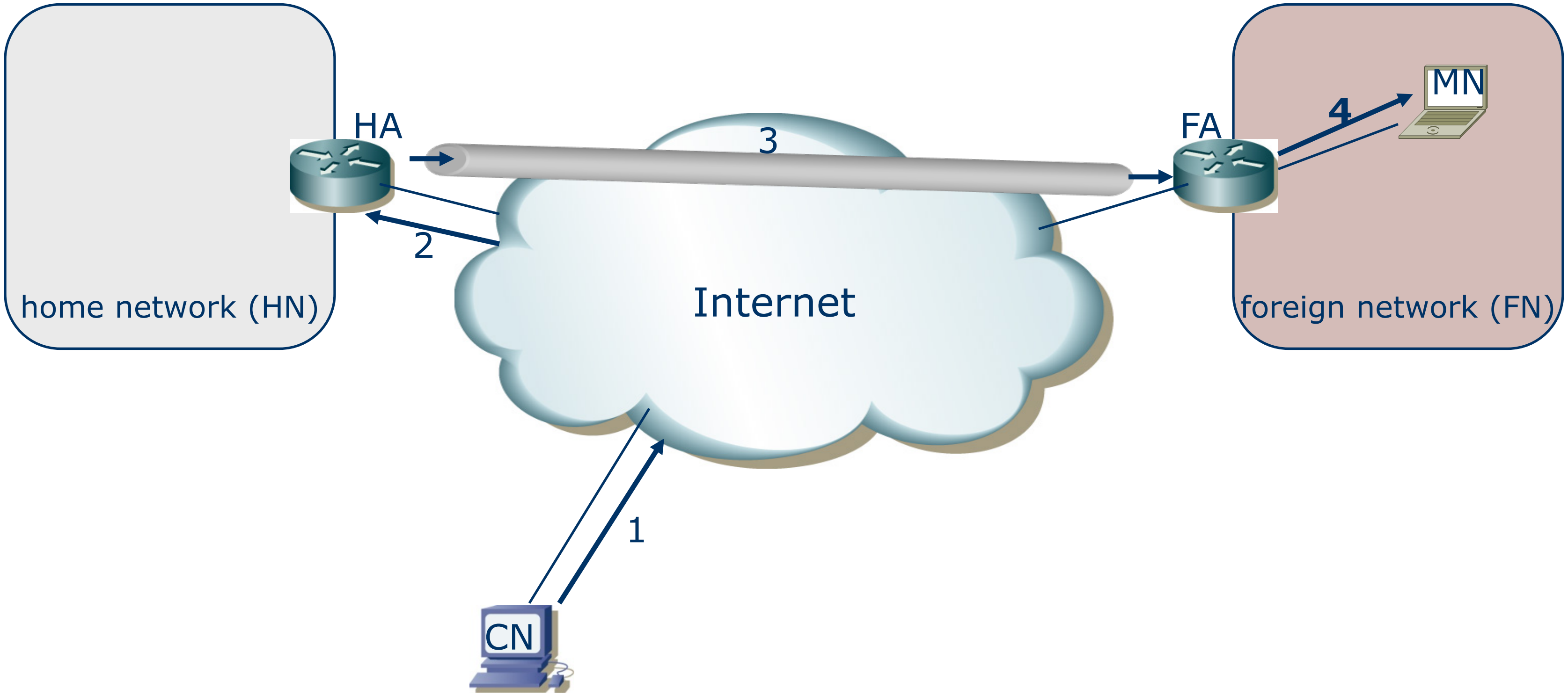
3. HA forwards the packet to FA



source IP: HA destination IP: FA	source IP: CN destination IP: MN	data	data

The packet is tunnelled to FA, more precisely – to the CoA (care-of address).

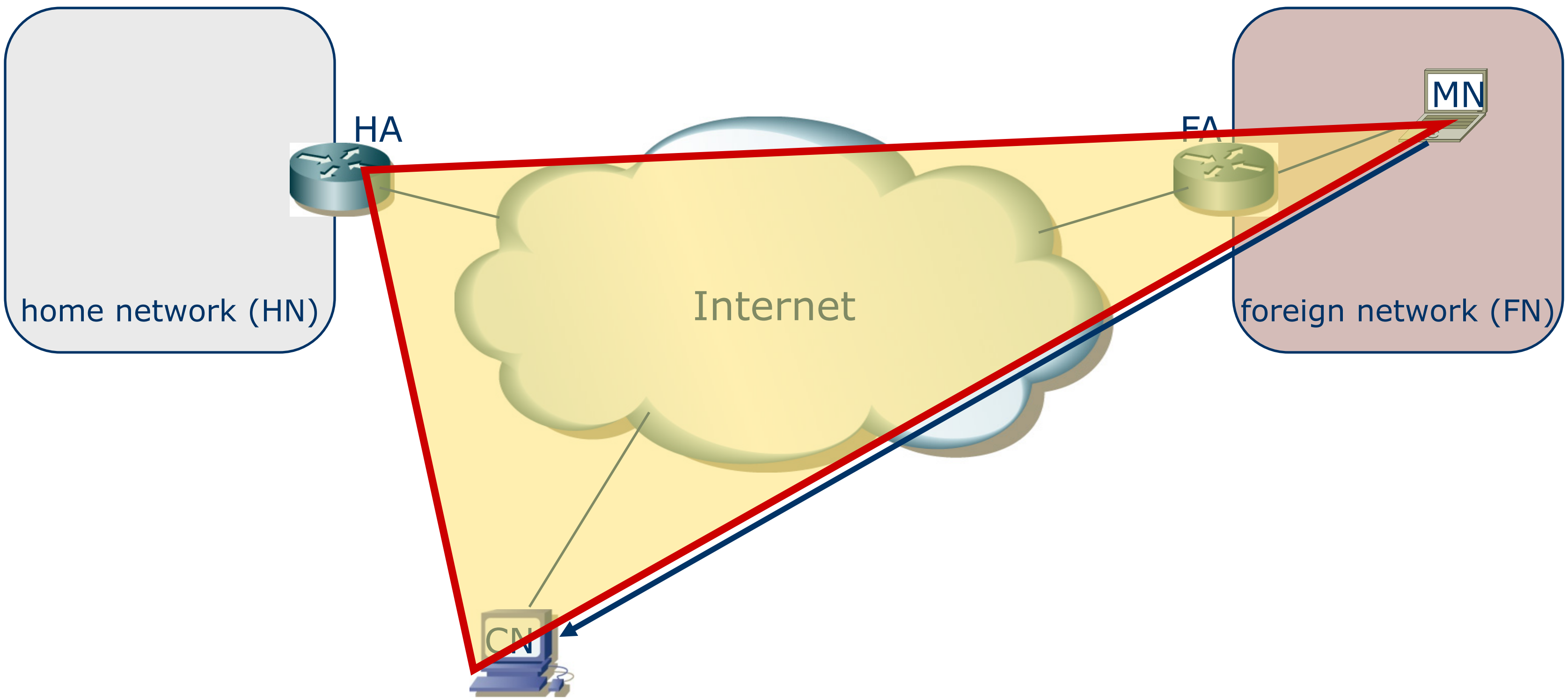
4. FA forwards the packet to MN



source IP: CN destination IP: MN	data

FA decapsulates the packet and forwards it to the destination (MN).

MN → CN



Packets from MN to CN are routed normally, without tunnelling, etc.
FA is a default gateway for MN.

The whole scenario is called „triangle routing”.

Mobile IPv6 (RFC 3775, 3776)

- IPv6:
  - separates „locator" from „identifier",
  - new header, called „mobility header" and a new type of routing header are defined

- Mobile IPv6 does not use FA
  - uses co-located CoA only (the tunnel ends at the MN)

Locator != Identifier

- Locator = localizes the host (tunnel endpoint, CoA)
  – topologically correct in FN,
  – created, e.g., with SLAAC,
  – used for communicating with HN,
  – MN communicates the HA to create a tunnel to itself

typical

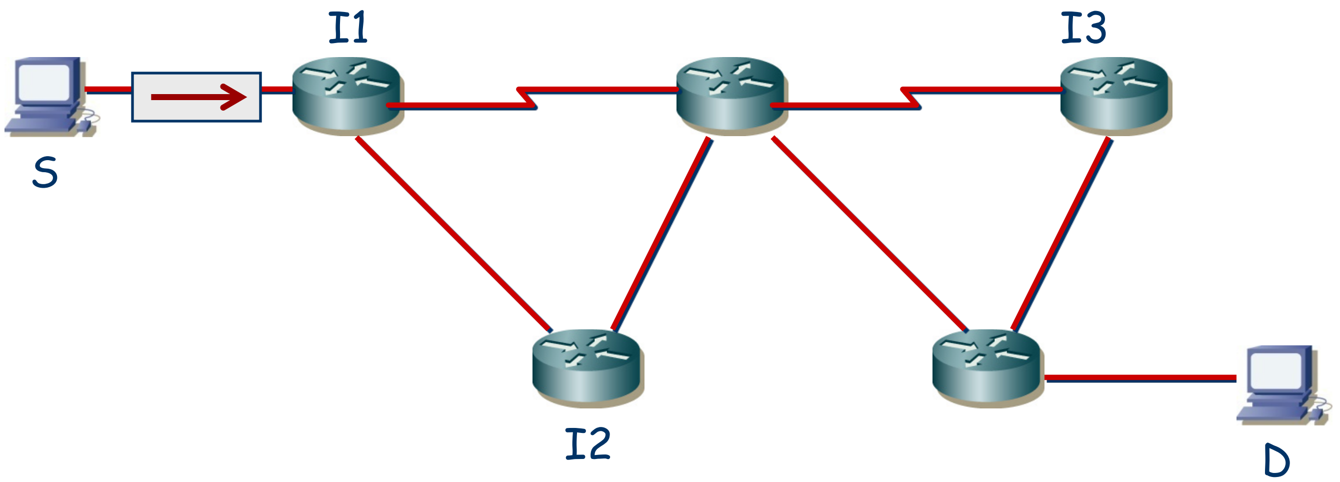
- Identifier = identifies the host (connection endpoint)
  – persistent,
  – globally unique,
  – used for communicating with other hosts

- Could CN forward packets directly to MN?
  – Yes – there is a Routing Header in IPv6…
  – … but CN needs to know the CoA

optional

Routing Header

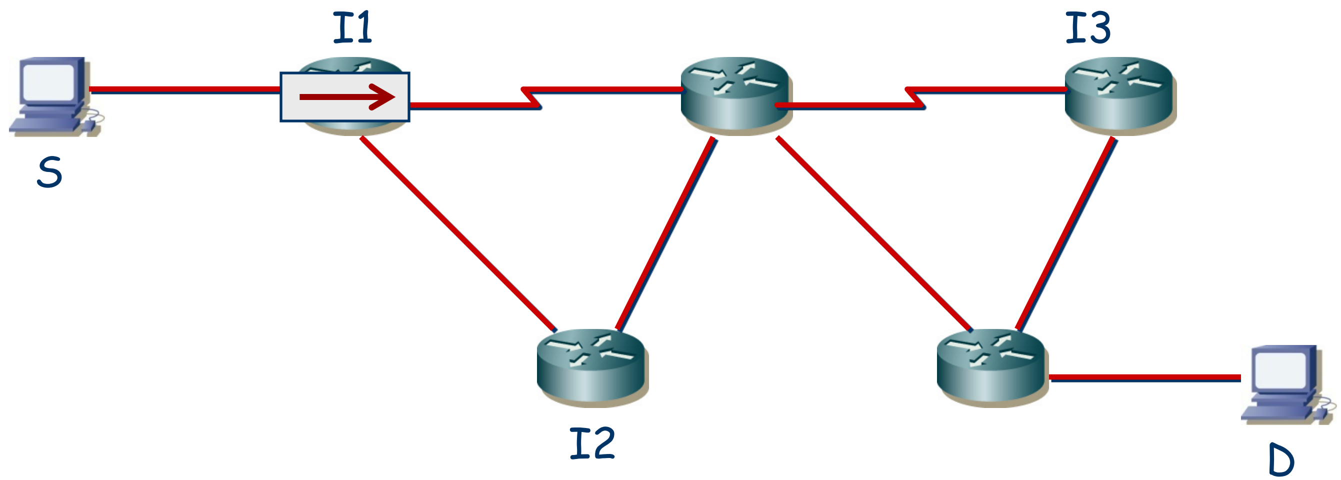
- Example:



Source Address = S			
Destination Address = **I1**			
Next Header = xx	Hdr Ext Len = 6	Routing Type = 0	Segments Left = 3
Address[1] = I2			
Address[2] = I3			
Address[3] = **D**			

Routing Header

- Example:

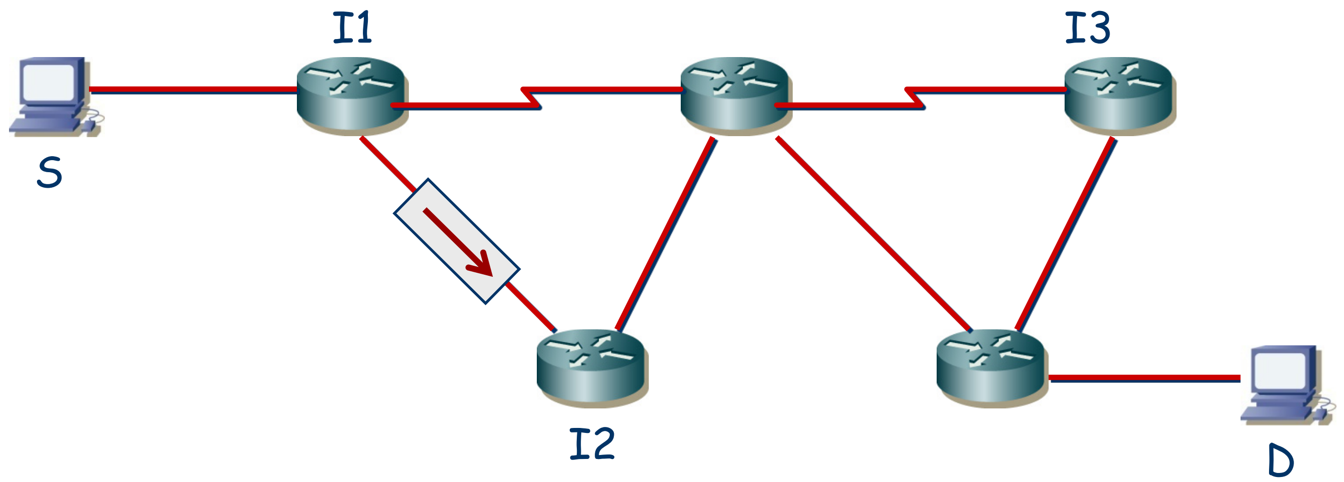


Source Address = S			
Destination Address = **I1**			
Next Header = xx	Hdr Ext Len = 6	Routing Type = 0	Segments Left = **3**
Address[1] = **I2**			
Address[2] = I3			
Address[3] = D			

Router I1 exchanges the first entry with the destination address and decreases the counter value.

Routing Header

- Example:

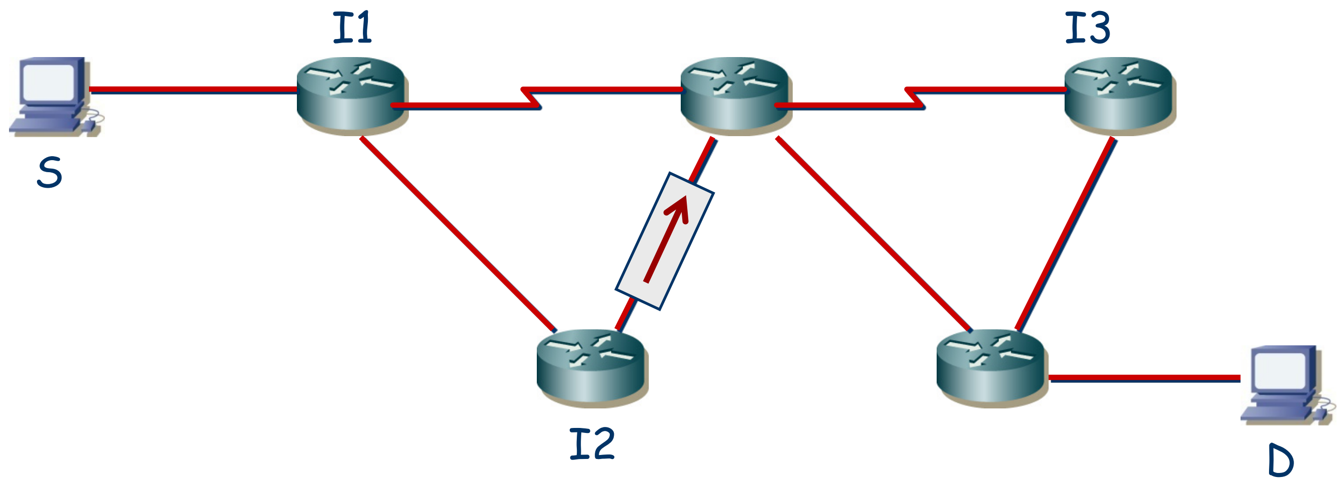


Source Address = S			
Destination Address = **I2**			
Next Header = xx	Hdr Ext Len = 6	Routing Type = 0	Segments Left = **2**
Address[1] = **I1**			
Address[2] = I3			
Address[3] = D			

Router I1 exchanges the first entry with the destination address and decreases the counter value.

Routing Header

- Example:

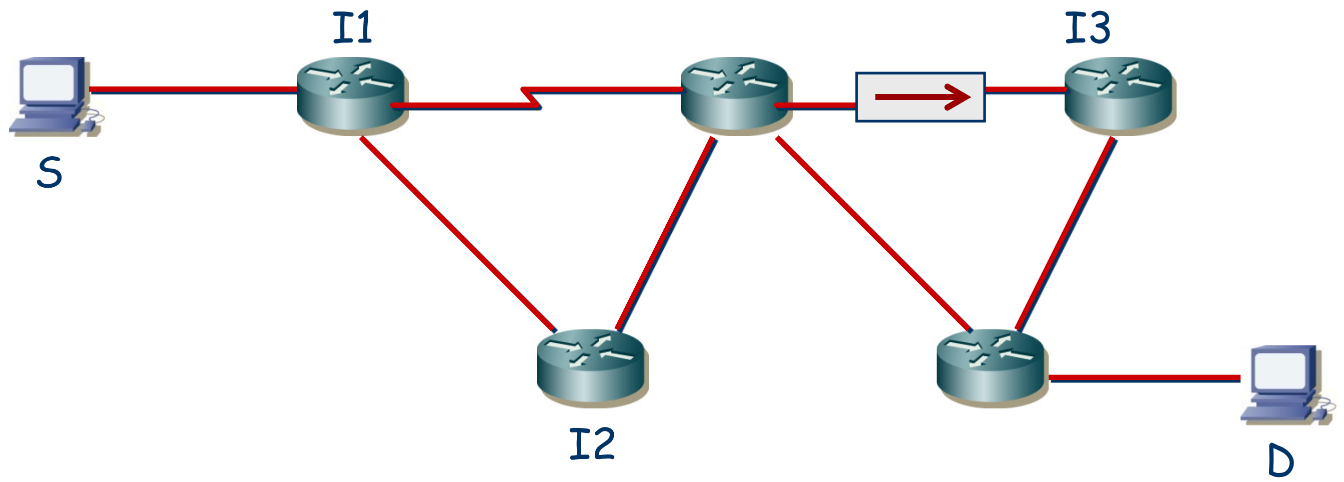


Source Address = S			
Destination Address = **I3**			
Next Header = xx	Hdr Ext Len = 6	Routing Type = 0	Segments Left = **1**
Address[1] = I1			
Address[2] = **I2**			
Address[3] = D			

Router I2 exchanges the destination address with the second entry.

Routing Header

- Example:

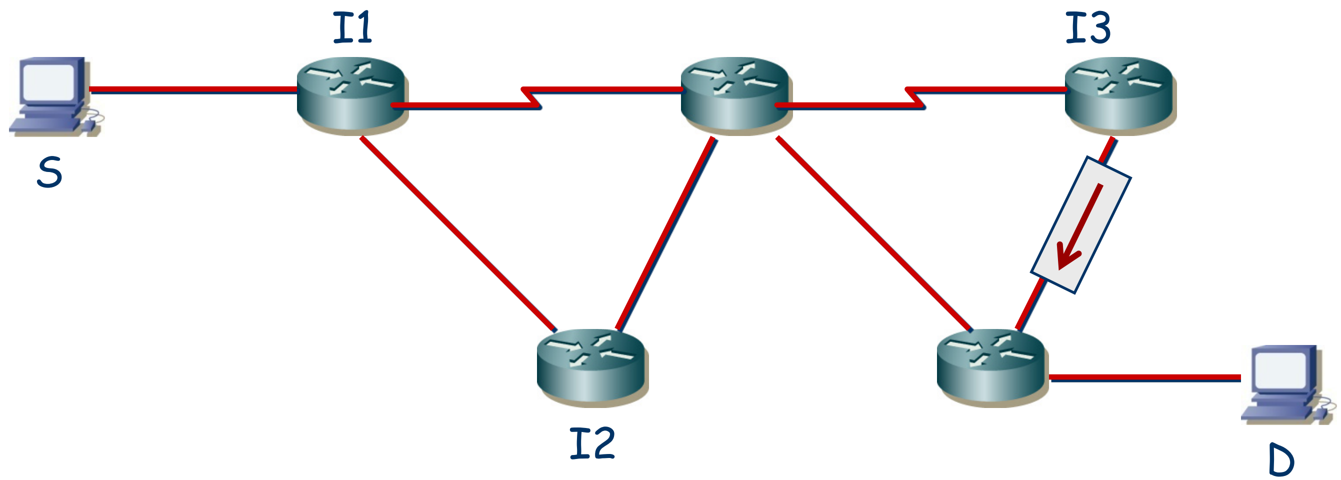


Source Address = S			
Destination Address = **I3**			
Next Header = xx	Hdr Ext Len = 6	Routing Type = 0	Segments Left = **1**
Address[1] = I1			
Address[2] = **I2**			
Address[3] = D			

The next router is not on the list, so it does not modify anything

Routing Header

- Example:



Source Address = S			
Destination Address = **D**			
Next Header = xx	Hdr Ext Len = 6	Routing Type = 0	Segments Left = **0**
Address[1] = I1			
Address[2] = I2			
Address[3] = **I3**			

Router I3 exchanges the destination address with the last entry.
In Mobile IPv6, D could be the MN.

To sum up…

(1) IPv4 deficiencies:

addressing, (in)security, no autoconfiguration;
NAT is a workaround, not a solution

(2) IPv6 addressing

hierarchical, designed with L2 in mind

(3) neighbor discovery

a bunch of mechanisms for various purposes

(4) autoconfiguration

stateless (SLAAC) or stateful (DHCP)

(5) tunnelling

the way to interconnect IPv6 islands through IPv4 ~~mud~~ sea

(6,7) IPv6 packet structure & protocol optimizations

designed so that routers can operate on L3 only

(8) Mobile IPv6

briefly touched; something to read about during summer holidays ;-)

Sources (again)

- **RFC 2460: Internet Protocol, Version 6 (IPv6) Specification**
- RFC 3587: IPv6 Global Unicast Address Format
- RFC 4291: IPv6 Addressing Architecture
- RFC 3177: IAB/IESG Recommendations on IPv6 Address Allocations to Sites
- RFC 3879: Deprecating Site Local Addresses
- RFC 3697: IPv6 Flow Label Specification
- RFC 2675: IPv6 Jumbograms
- **RFC 4294: IPv6 Node Requirements**
- RFC 3484: Default Address Selection for IPv6
- RFC 4311: Host-to-Router Load Sharing
- RFC 2991: Multipath Issues in Unicast and Multicast Next-Hop Selection
- RFC 2464: Transmission of IPv6 Packets over Ethernet Networks
- **RFC 4861: Neighbor Discovery for IPv6**
- RFC 2462: IPv6 Stateless Address Autoconfiguration
- RFC 4191: Router Preferences and More-Specific Routes
- RFC 4311: IPv6 Host-to-Router Load Sharing
- RFC 4443: Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6)
- www.cisco.com: Implementing IPv6 for Cisco IOS Software
- cisco.customerelearning.com: IPv6

To be continued ...

Recommended reading:
- RINA (Recursive Internet Network Infrastructure)
- NDN (Named Data Networking)