

Introduction to IPv6

March 2016 – ICTP - Trieste

Alvaro Vives (alvaro.vives@nodo6.com)

NODO6 (www.nodo6.com)


Contents

- ▶ 1 Digital Data Transmission
- ▶ 2 Switched Packet Networks
- ▶ 3 Layered Model
- ▶ 4 IPv4 and IPv6 basics
 - ▶ 4.1 IPv4 Header
 - ▶ 4.2 IPv6 Header
 - ▶ 4.3 Differences
- ▶ 5 IP addresses (v4/v6)
 - ▶ 5.1 IPv4 Notation
 - ▶ 5.2 IPv6 Notation
 - ▶ 5.3 IPv6 types of addresses
 - ▶ 5.4 Interface Identifier (IID)
 - ▶ 5.5 IPv6 Addresses Exercise
- ▶ 6 IPv6 Protocols and Autoconfiguration
 - ▶ 6.1 ICMPv6
 - ▶ 6.2 Path MTU Discovery (PMTU-D)
 - ▶ 6.3 NDP
 - ▶ 6.4 Autoconfiguration: DHCPv6 vs. SLAAC

Objectives

- ▶ Give an overview of IP data networks to understand where we are nowadays
- ▶ “Equalize” students knowledge (in order to)
- ▶ Be prepared for the IPv6 concepts we will see during the workshop

Digital Data Transmission (I)

- ▶ Objective is to send some information from one place/device to another
- ▶ Different type of info, through different transport networks
- ▶ You have to codify the info -> digitally 
 - ▶ Three symbols: using 1 transmitted unit of information you could represent 3 different codes (A,B or C) (3^1)
 - ▶ If you transmit 2 units of information: 9 codes (3^2)
- ▶ Binary codification -> uses two characters: 0 / 1
- ▶ Bit (0 or 1) minimal unit of information
- ▶ Byte = 8 bits -> used for ASCII characters => 256 (2^8)

Digital Data Transmission (II)

- ▶ If you want to transmit “hi”:

- ▶ h ->

0	1	1	0	1	0	0	0
---	---	---	---	---	---	---	---

- ▶ i ->

0	1	1	0	1	0	0	1
---	---	---	---	---	---	---	---

- ▶ This codification is defined by ASCII

- ▶ There could be other ones

- ▶ You could codify hexadecimal (16 from 0 to F) numbers using 4 bits ($2^4 = 16$)

- ▶ 0 =

0	0	0	0
---	---	---	---

 -> Represented as 0x0

- ▶ 1 =

0	0	0	1
---	---	---	---

 -> Represented as 0x1

- ▶ 2 =

0	0	1	0
---	---	---	---

 -> Represented as 0x2

- ▶ ...

- ▶ A =

1	0	1	0
---	---	---	---

 -> Represented as 0xA

Switched Packet Networks (I)

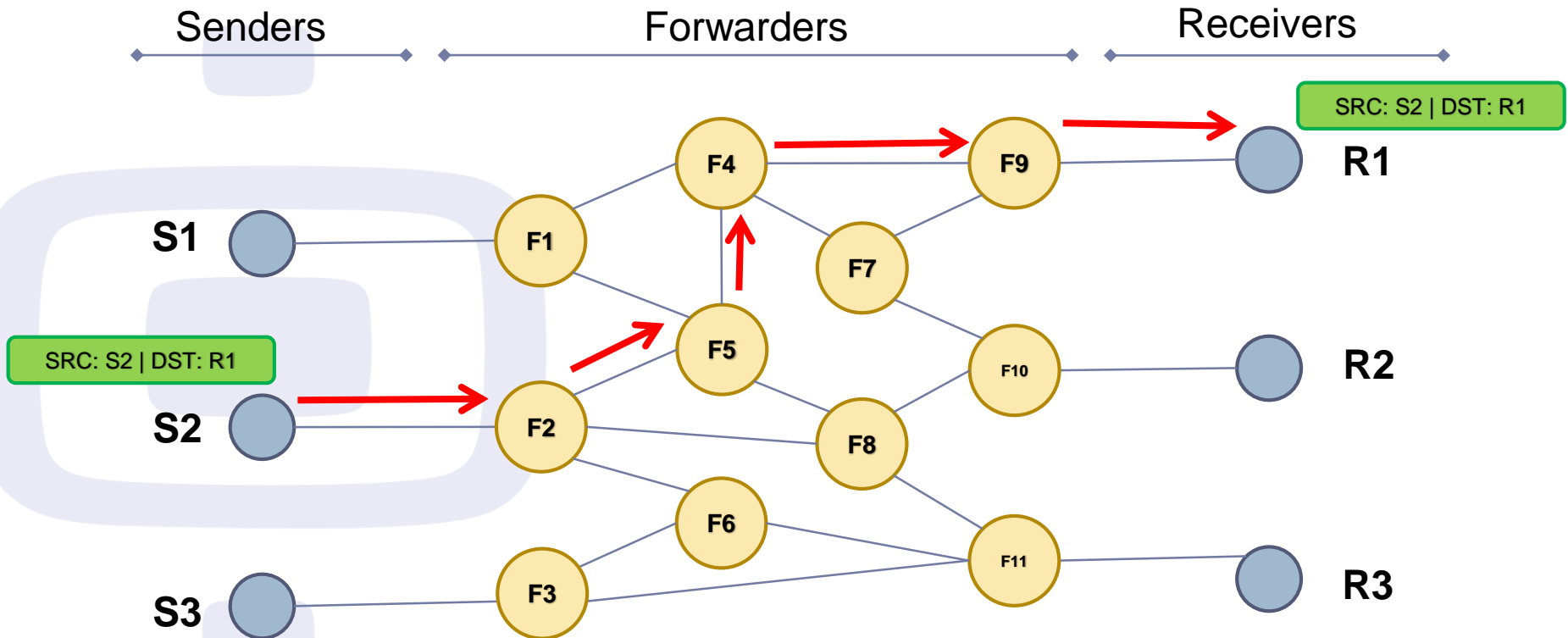
- ▶ Two options to send information:
 1. **Switched Circuits:** fixed paths, reserved resources, communication starts only when circuit is established (example: telephone)
 2. **Packet Switching:** paths can vary, shared resources (best effort), communication can start at any moment (example: postal mail, Internet)

- ▶ **Packet switching is much more efficient and flexible**

Switched Packet Networks (II)

- ▶ Basic elements on a switched network:
 1. **Sender:** Generates the info to be sent to a receiver. Should codify the message.
 2. **Receiver:** Is the destination of the information sent by the sender. Should decode the message.
 3. **Forwarder:** Not the origin or the destination of the information. Just receive and forward the information in its path to the destination
 4. **Identification:** Each element in the switched network should be uniquely identified

Switched Packet Networks (III)



Switched Packet Networks (VI)

▶ Role Play

▶ Three kinds of roles: senders, receivers, forwarders

1. **Receivers:** get an IP destination card -> shows it
2. **Senders:** take an origin IP card and envelopes -> choose one destination IP from receivers showing
3. **Forwarders:** will receive packet envelopes and forward to the best neighbor

▶ Start:

1. Senders: put the first part of the word in an envelope and write the origin and destination IP for it
2. Senders: pass the packet to their "gateway" router
3. Forwarders: get packets, look at the destination IP and pass it to the router they consider is in the shortest path to the destination IP
4. Receiver: get packets and put together word parts, when it has the full word it should say it loud

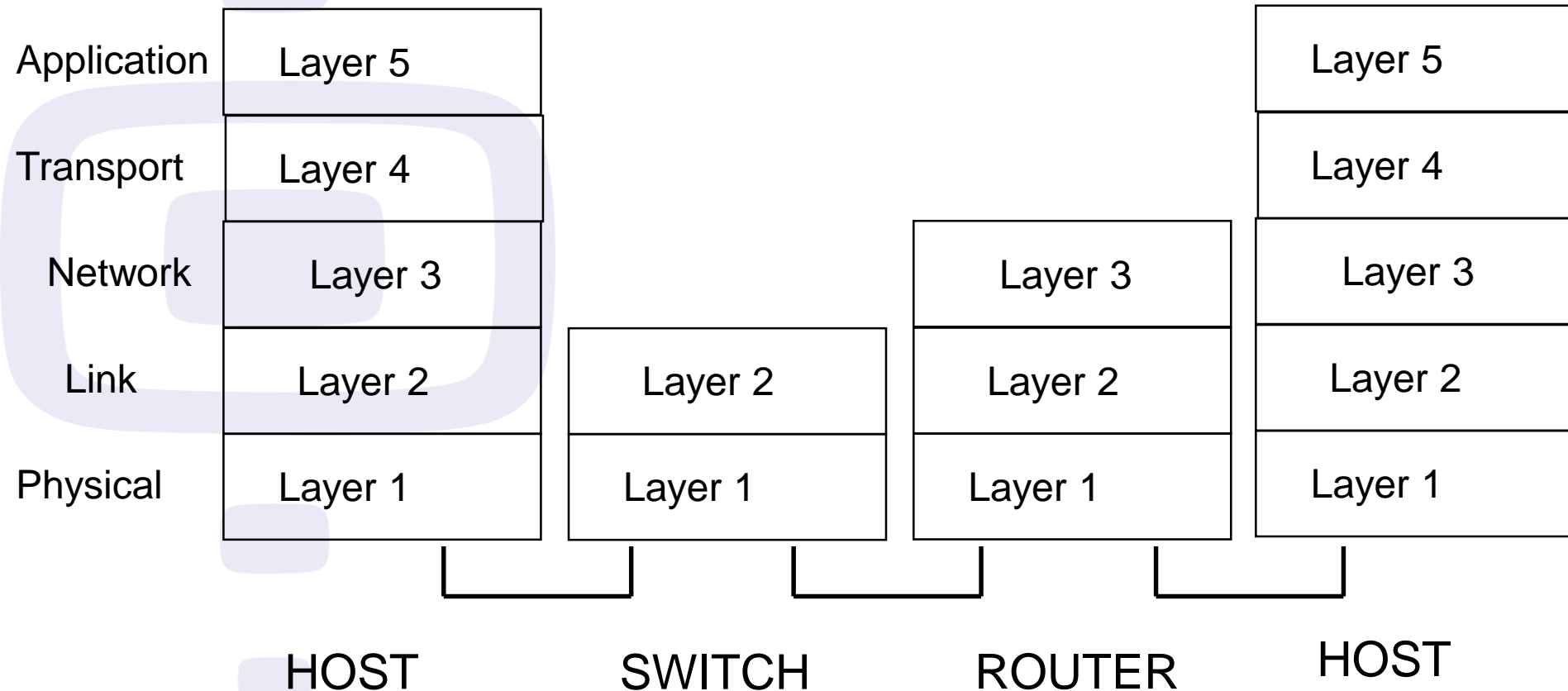
Layered Model (I)

▶ Let's define things:

1. **Layered model:** physical, link, network, etc. each one is in charge of different things/services
2. **Network elements:** Node, host, router, server
3. **Addresses:** link layer, network layer
4. **Protocol:** definition of the format and order of messages exchanged between two or more communicating entities, as well as the actions taken on the transmission and/or reception of a message or other event

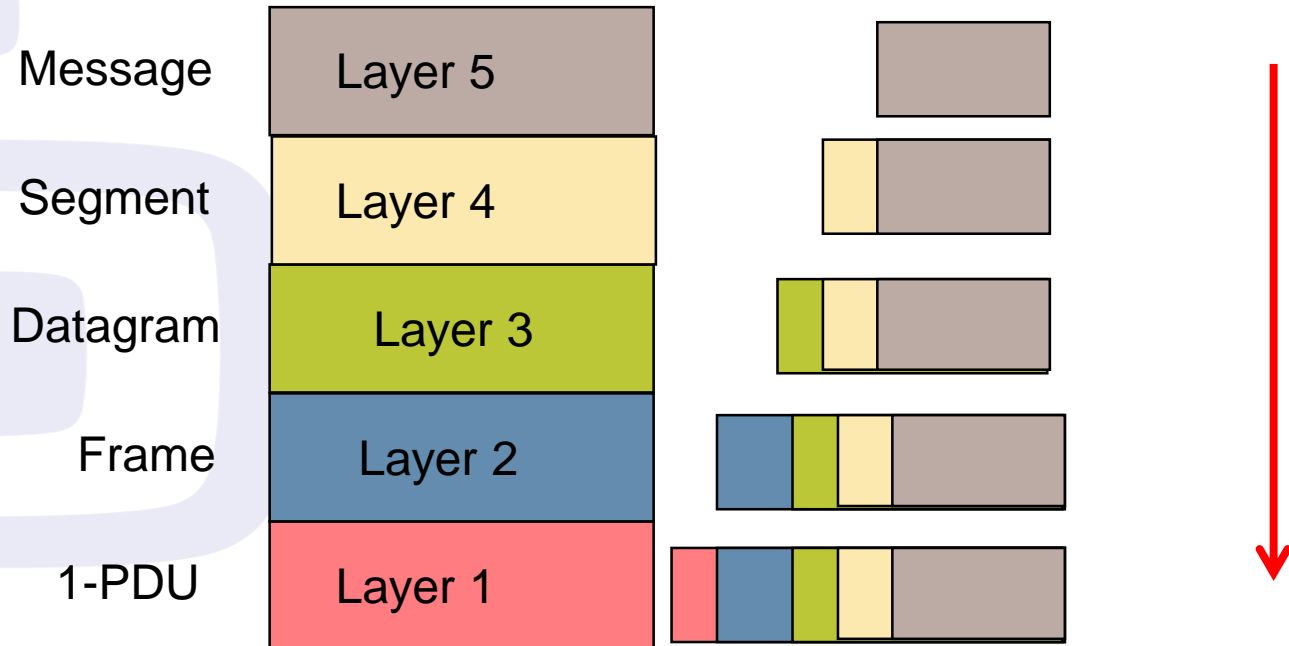
Layered Model (II)

- ▶ TCP/IP layered model -> Used in Internet



Layered Model (III)

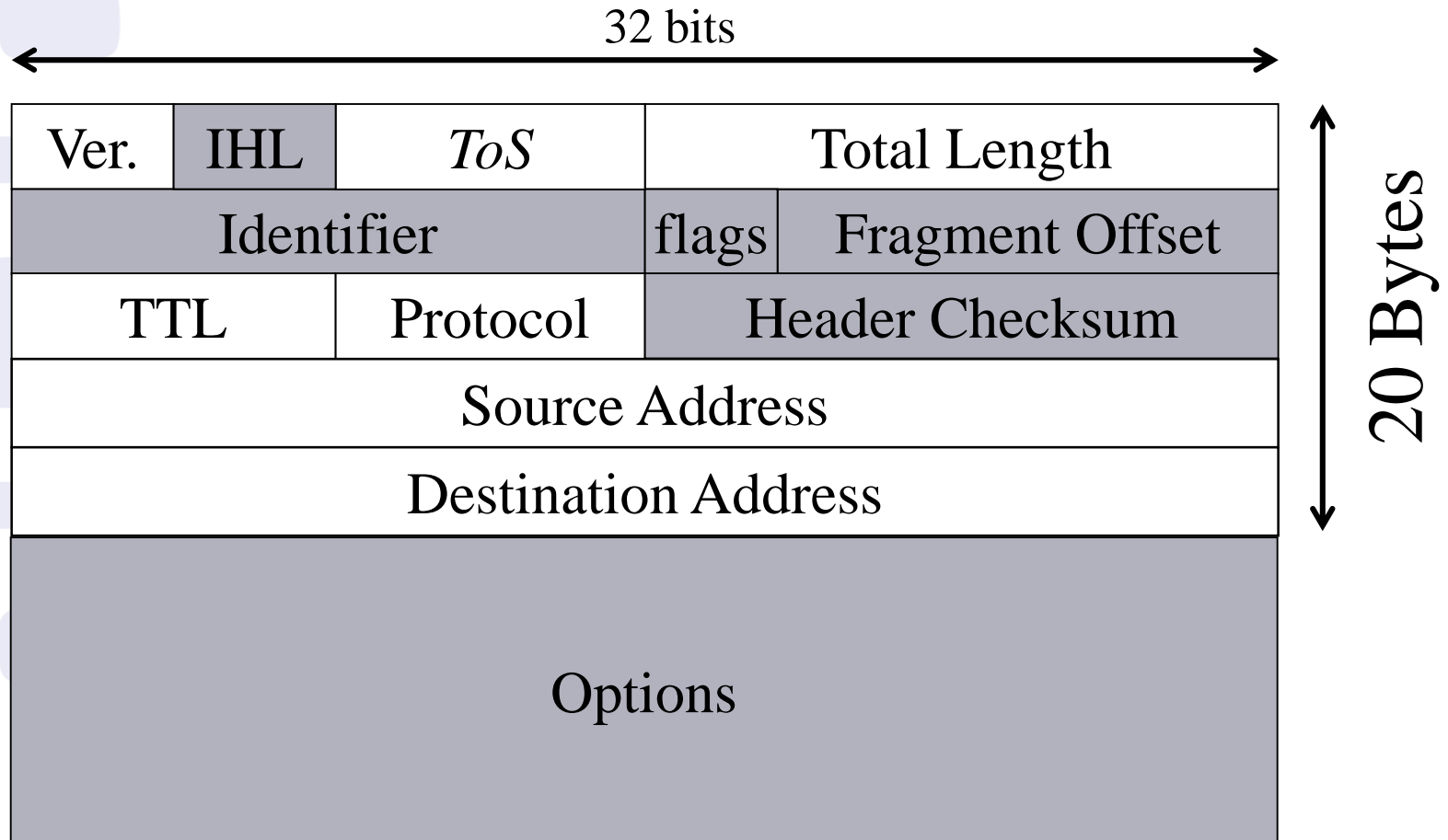
▶ PDU: Protocol Data Unit



- ▶ Layer 3 Header includes Source and destination Network Address (IP Address)
- ▶ Layer 3 is the only common layer in Internet: IP

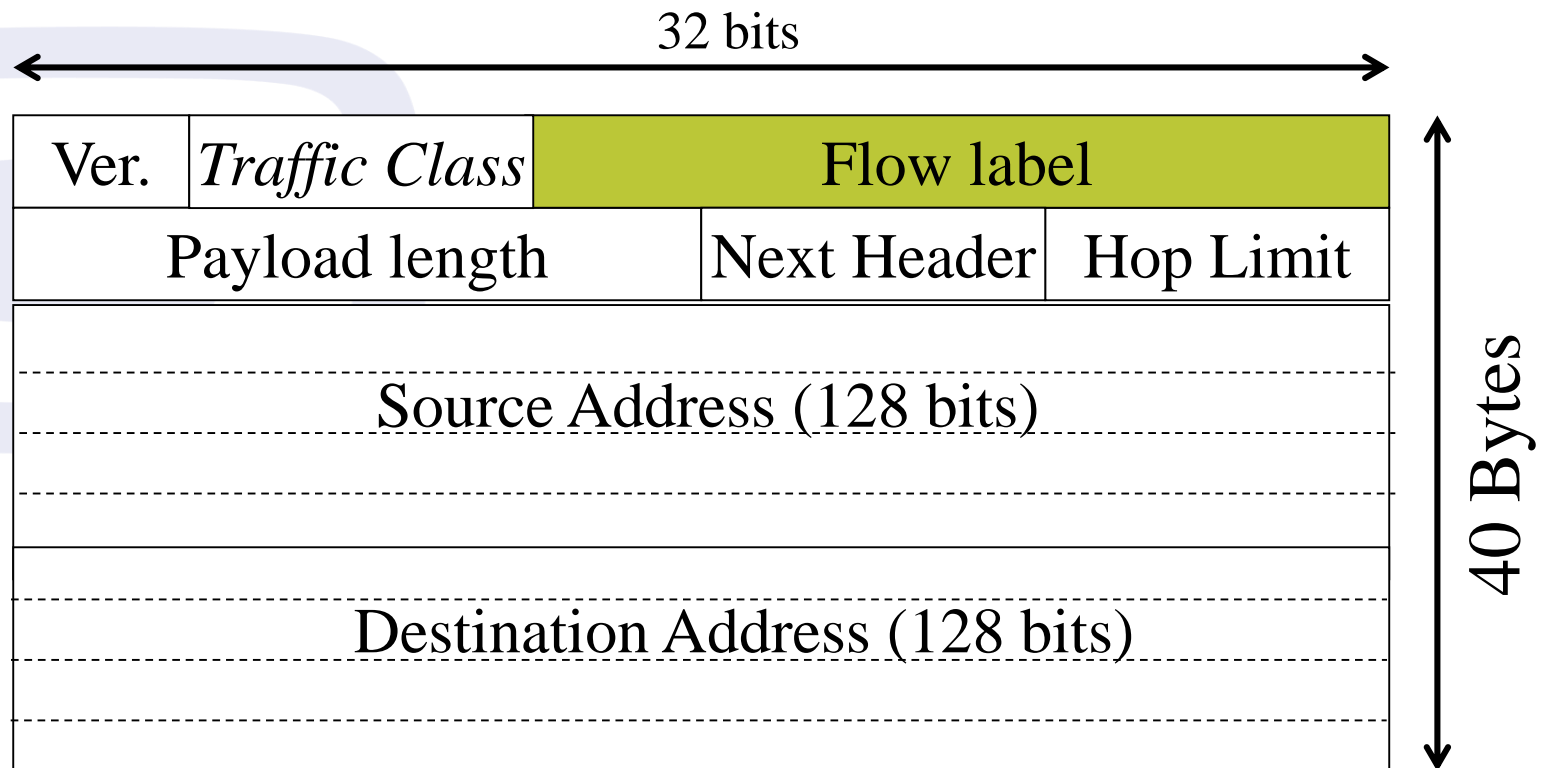
IPv4 and IPv6 basics (I)

- ▶ IPv6 is an evolution of IPv4



IPv4 and IPv6 basics (II)

- ▶ Simplified, fixed-length, 64 bits aligned -> complexity from core to border

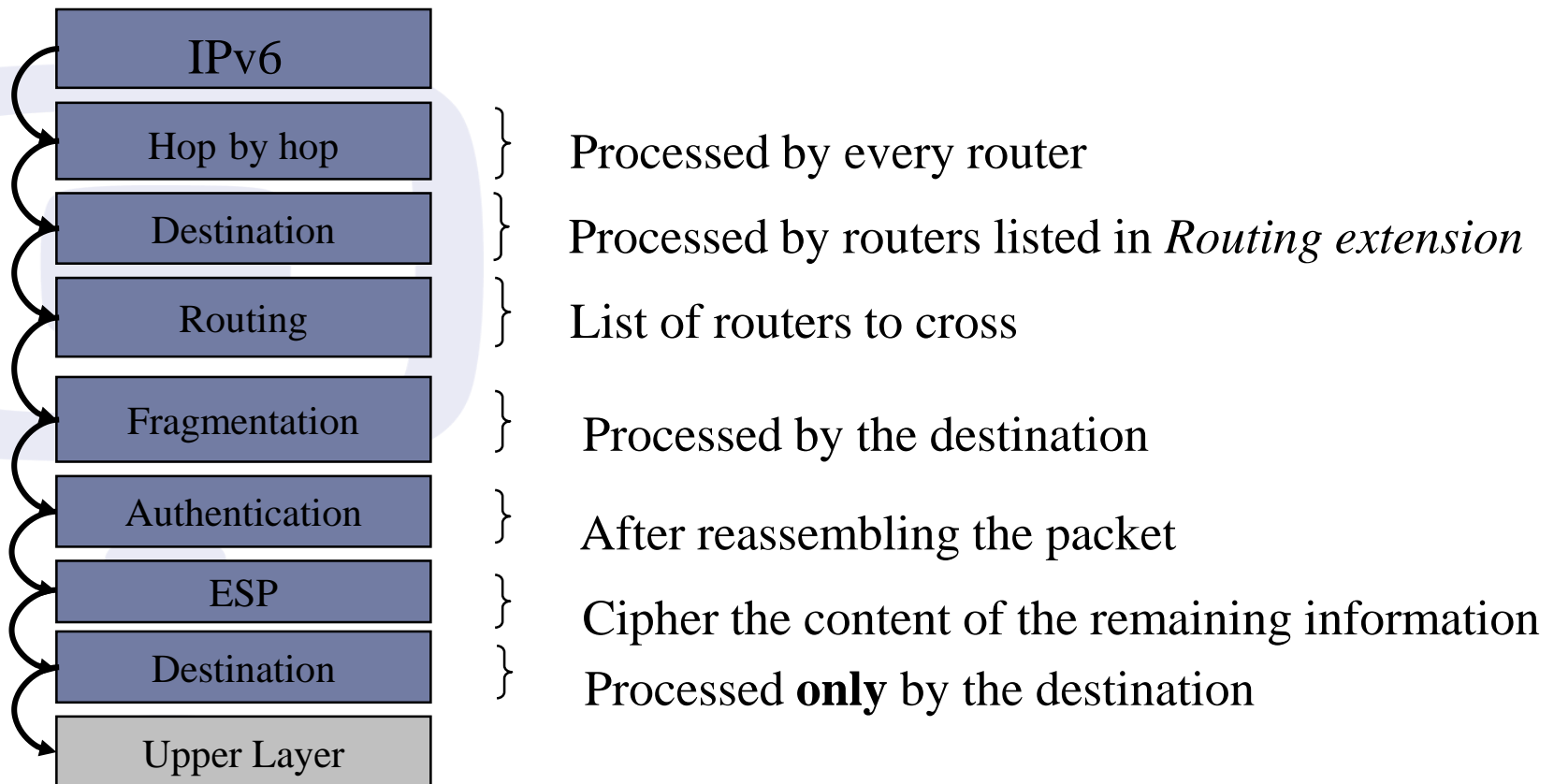


IPv4 and IPv6 basics (III)

- ▶ New IPv6 basic header has advantages:
 - ▶ Simplified, fixed length, and aligned to 64 bits -> routers can process it faster --> Scalable
 - ▶ Redundant or not needed features are eliminated: checksum, header length (IHL)
 - ▶ New QoS field (IntServ): Flow Label
 - ▶ Much more addresses

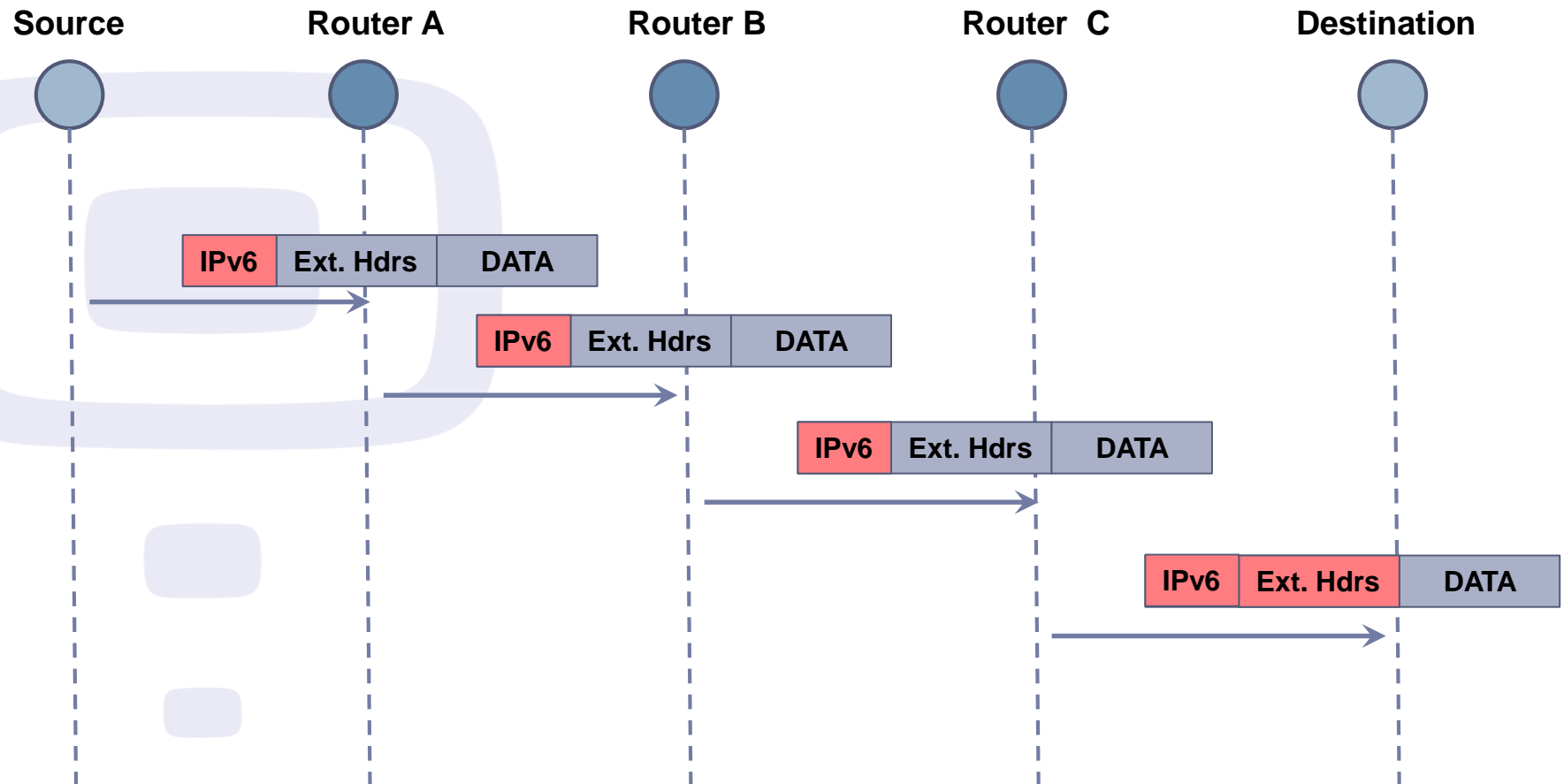
IPv4 and IPv6 basics (IV)

- ▶ Extension Headers: To cover IP layer needs -> flexible
- ▶ Limited and ordered: used only once (exception Destination)



IPv4 and IPv6 basics (V)

- ▶ Basic IPv6 header is processed in all hops
- ▶ Extension headers are processed in destination (exception Hop-by-hop)



IPv4 addresses (I)

- ▶ IPv4 addresses have 32 bits
- ▶ Represented using decimal notation of each byte (8 bits) separated by .
- ▶ Examples: 10.1.1.2, 192.168.11.1
- ▶ Each decimal number corresponds to 8 bits, for example: 10 -> 00001010
- ▶ Do you remember/know about binary to decimal conversions?

IPv4 addresses (II)

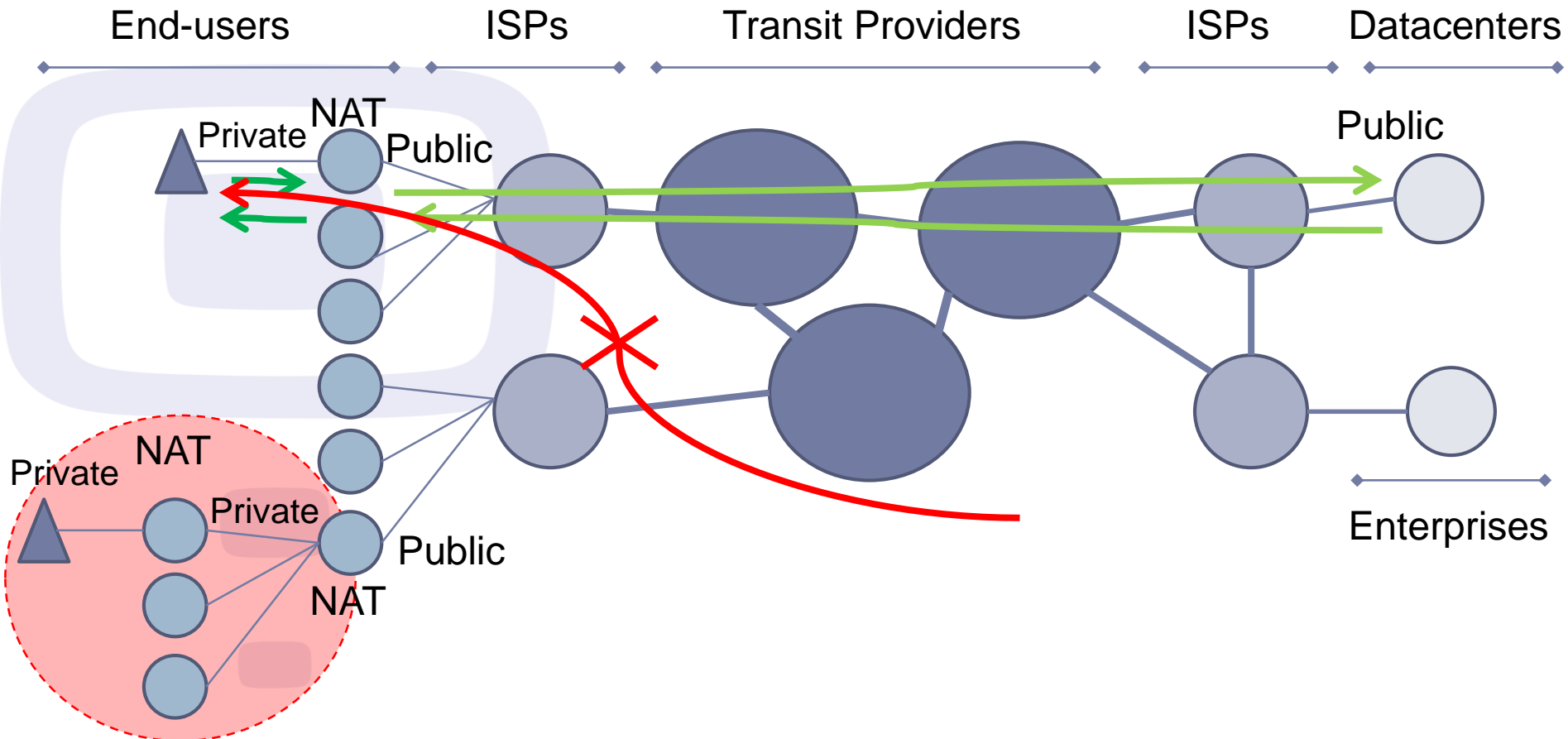
- ▶ At the beginning different “classes” were defined:
 - ▶ Class A: 8 bits mask (/8) -> first byte 0 to 127
 - ▶ Class B: (/16) -> first byte 128 to 191
 - ▶ Class C: (/24) -> first byte 192 to 223
- ▶ Later, classes were abandoned by CIDR (Classless Inter Domain Routing) Notation: prefix / length
- ▶ Example 10.1.2.0/24:
 - ▶ 24 bits network prefix
 - ▶ 8 bits for hosts
 - ▶ 254 possible host addresses (all 0s (network) and all 1s (broadcast) could not be used)

IPv4 addresses (III)

- ▶ **Private addresses were defined:**
 - ▶ 10.0.0.0/8 (1 x A): 10.0.0.0 to 10.255.255.255
 - ▶ 172.16.0.0/12 (16 x B): 172.16.0.0 to 172.31.255.255
 - ▶ 192.168.0.0/16 (256 x C): 192.168.0.0 to 192.168.255.255
- ▶ **Private addresses are used behind a NAT device**
 - ▶ Works “well” in a client-server model
 - ▶ Do not allow for P2P or similar applications
 - ▶ Do not allow innovation on the Internet
 - ▶ Makes software development more expensive
 - ▶ Management and security gets harder

IPv4 addresses (IV)

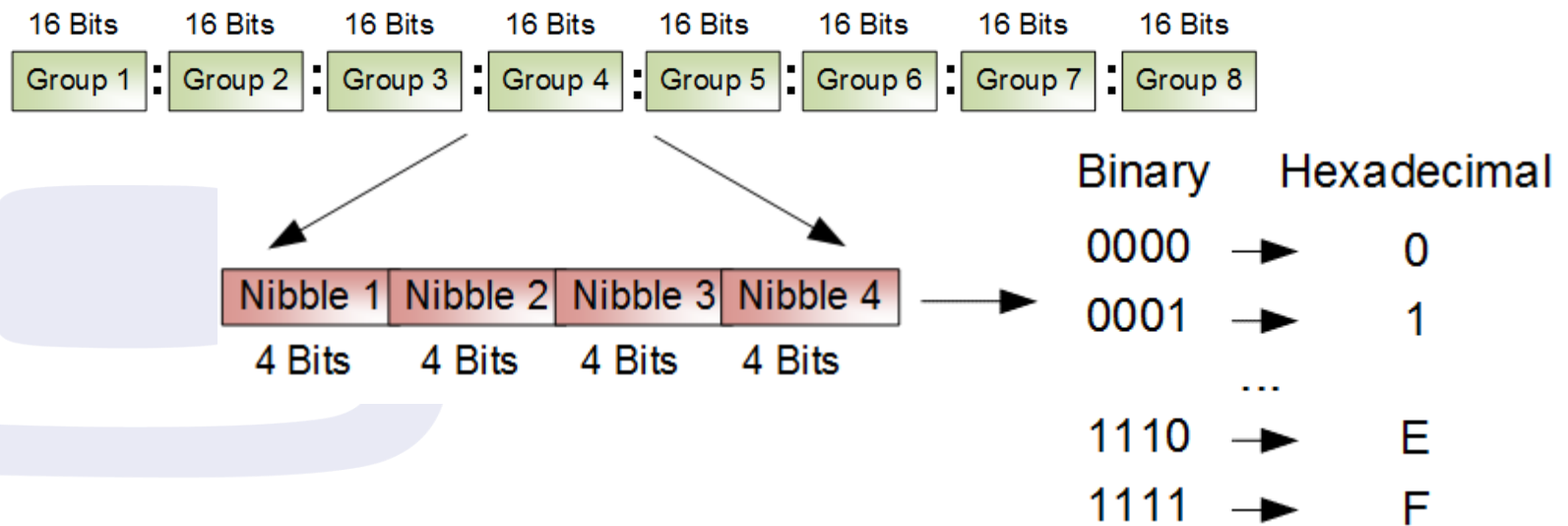
- ▶ NAT issues examples: private not reachable, several levels of NAT



IPv6 addresses: Types

- ▶ Unicast (one-to-one)
 - ▶ Link-local
 - ▶ Unique Local (ULA)
 - ▶ IPv4-mapped
 - ▶ Global (GUA)
 - ▶ Site-local (deprecated)
 - ▶ IPv4-compatible (deprecated)
- ▶ Multicast (one-to-many)
 - ▶ Anycast (one-to-nearest) (taken from unicast space)
 - ▶ Reserved (Trans. Mechs, documentation, loopback, etc.)
 - ▶ There are no BROADCAST addresses -> well-known multicast

IPv6 addresses: Notation (I)



IPv6 addresses: Notation (II)

▶ IPv6 address notation rules:

- ▶ 8 Groups of 16 bits separated by “:”
- ▶ Hexadecimal notation of each nibble (4 bits) ->
- ▶ No case sensitive

Binary	-	Hex.
0000	->	0
0001	->	1
0010	->	2
...		
1110	->	E
1111	->	F

▶ Compression rules:

- ▶ Leftmost zeroes within each group could be eliminated
- ▶ One or more groups of all zeroes could be changed by “::”.
Only once!

▶ Use “[]” to specify port -> `http://[2001:db8::10]:8080`

▶ Examples:

- ▶ `2001:0db8:0102:0DA0:0000:0000:0000:1000` ->
`2001:db8:102:DA0::1000`
- ▶ `2001:db8:0000:0000:0020:0000:0000:0abc` -> ?

IP addresses (v4/v6)(VII)

- ▶ Network prefixes follow CIDR notation
- ▶ Compression rules could be applied
- ▶ Examples:

- ▶ 2001:db8::**/32** -> **2001:0db8:**0000:0000:0000:0000:0000:0000

- ▶ 2001:db8:1200::**/40** -> **2001:0db8:12**00:0000:0000:0000:0000:0000

- ▶ 2001:db8:abcd::**/48** -> **2001:0db8:abcd:**0000:0000:0000:0000:0000

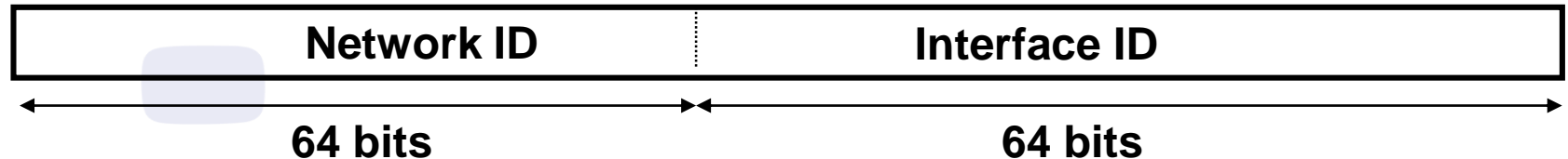
- ▶ Non-prefix bits (rightmost) used for subnetting

- ▶ Example: I'll take the first two /52 prefixes out of 2001:db8:abcd::

- ▶ 2001:0db8:abcd:**0**000:0000:0000:0000:0000 -> **2001:db8:abcd:0000::/52**

- ▶ 2001:0db8:abcd:**1**000:0000:0000:0000:0000 -> **2001:db8:abcd:1000::/52**

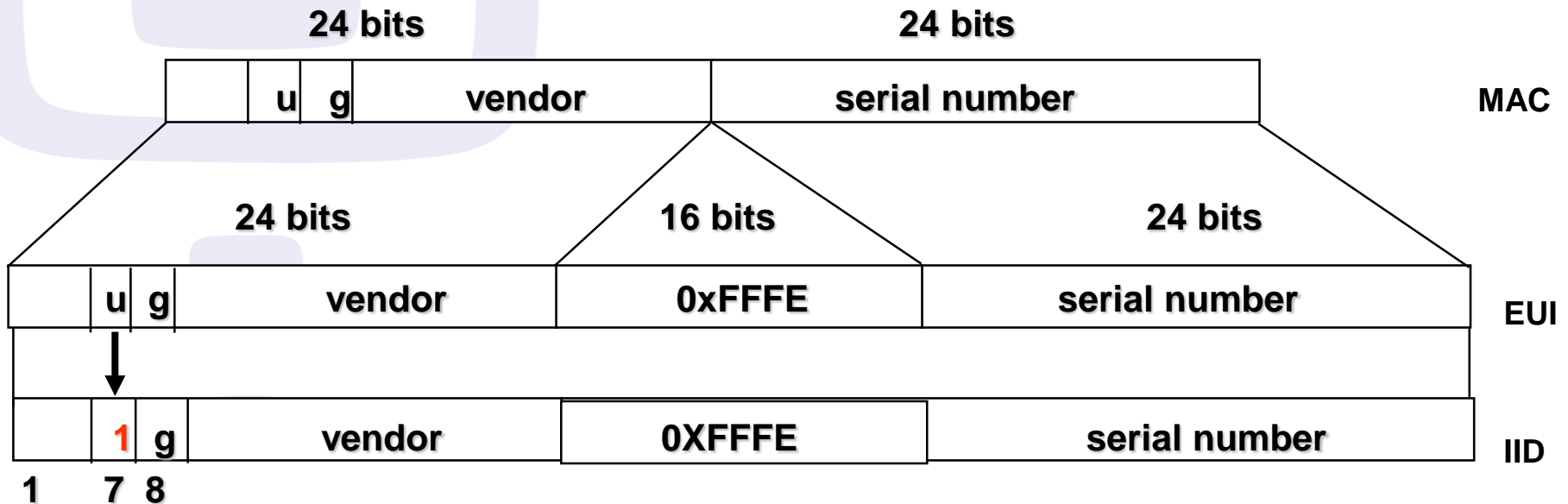
IPv6 addresses: IID (I)



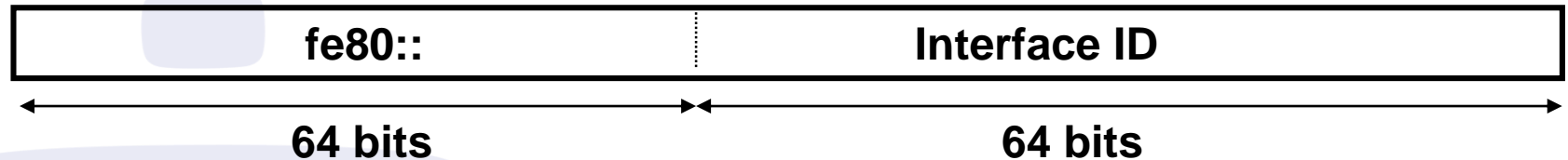
- ▶ Network prefix in a LAN will be /64
- ▶ Interface ID: 64 bits available to identify hosts in the LAN
- ▶ They could be created in many different ways
 - ▶ From MAC addresses (EUI-64)
 - ▶ Automatically using some kind of algorithm (randomly)
 - ▶ Manually
 - ▶ DHCPv6
- ▶ **TWO IDEAS HERE:**
 - ▶ /64 prefix for a LAN -> this is the minimum unit you will manage on your addressing plan
 - ▶ Interface identifier are generated locally on the host (except DHCP)

IPv6 addresses: IID (II)

- ▶ IEEE defines a mechanism to create an **EUI-64** from an IEEE 802 MAC address (Ethernet, FDDI)
- ▶ You get the IID modifying the EUI-64's u bit (Universal). Set to 1 to indicate universal scope and 0 to indicate local scope

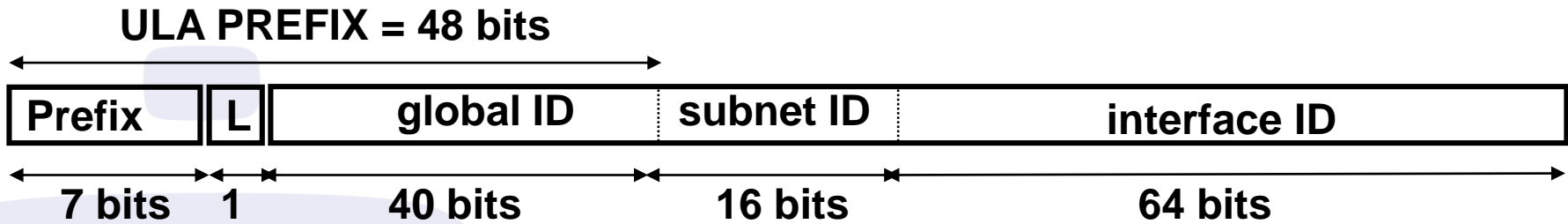


IPv6 addresses: Link-local



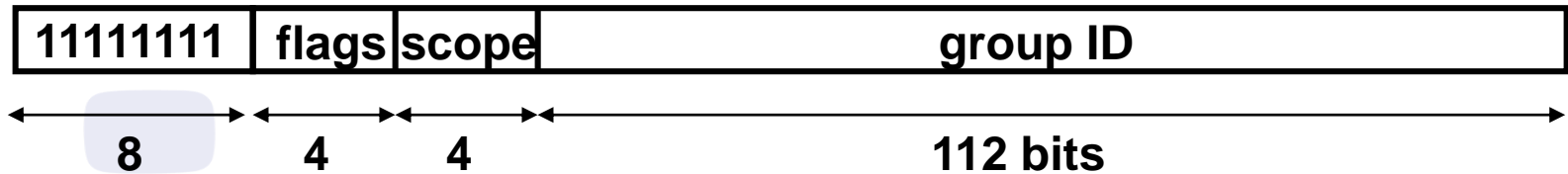
- ▶ Valid only in a link
- ▶ Always present in any IPv6-enabled interface
- ▶ Prefix `fe80::/10` -> In practice `fe80::/64` is used
- ▶ Interface ID is generated locally on the host: based on MAC, randomly or anyhow

IPv6 addresses: ULA



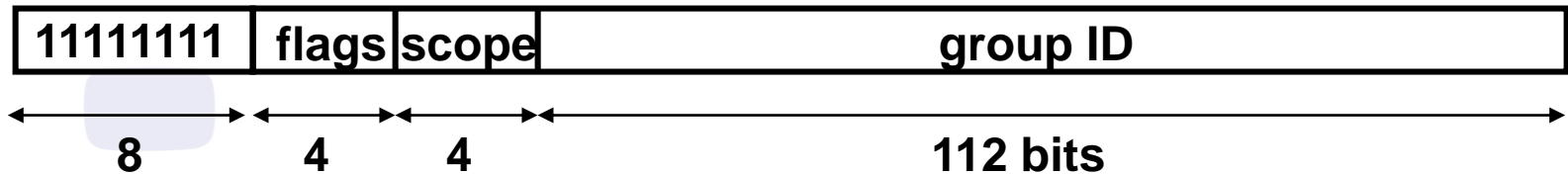
- ▶ ULA Addresses: FC00::- ▶ L = 1 if the prefix is locally assigned
- ▶ L = 0 may be defined in the future (RFC4193) (in practice used for centrally assigned prefixes)
- ▶ global ID: pseudo-randomly generated
- ▶ You'll create a /48 prefix, usually starting with FD00::

Direccionamiento IPv6: Multicast



- ▶ Prefijo FF00::/8
- ▶ Flags: usados para routing y servicios multicast
- ▶ Scope (ámbito): parte de la red donde dirección es válida
 - ▶ 1 - Interface-Local
 - ▶ 2 - link-local
 - ▶ 4 - admin-local
 - ▶ 5 - site-local
 - ▶ 8 - organization-local
 - ▶ E - global
- ▶ Group ID: Identifica el grupo multicast
- ▶ Sustituto de las Broadcast de IPv4

IPv6 addresses: Multicast



- ▶ Multicast Addresses: Prefix FF00::/8
- ▶ Flags: used for multicast routing and services
- ▶ Scope: part of network where address is valid
 - ▶ 1 - Interface-Local
 - ▶ 2 - link-local
 - ▶ 4 - admin-local
 - ▶ 5 - site-local
 - ▶ 8 - organization-local
 - ▶ E - global
- ▶ Group ID: Identifies the multicast group
- ▶ Substitute of IPv4 Broadcast addresses:
 - ▶ Well-known: FF02::1 (all nodes), FF02::2 (all routers), FF02::1:2 (all DHCP-agents)

IPv6 Multicast Addresses to Remember

- ▶ FF01::1, FF02::1 All-nodes
- ▶ FF01::2, FF02::2, FF05::2 All-routers
- ▶ SN (Solicited Node) multicast address, created from unicast
 - ▶ Last 24 bits of unicast address: “XY:ZTUV”
 - ▶ Associated SN is: FF02::1:FFXY:ZTUV
 - ▶ SN Prefix: FF02::1:FF00:0/96
- ▶ Each IPv6 node should join the SN multicast address associated with all its unicast and anycast IPv6 addresses

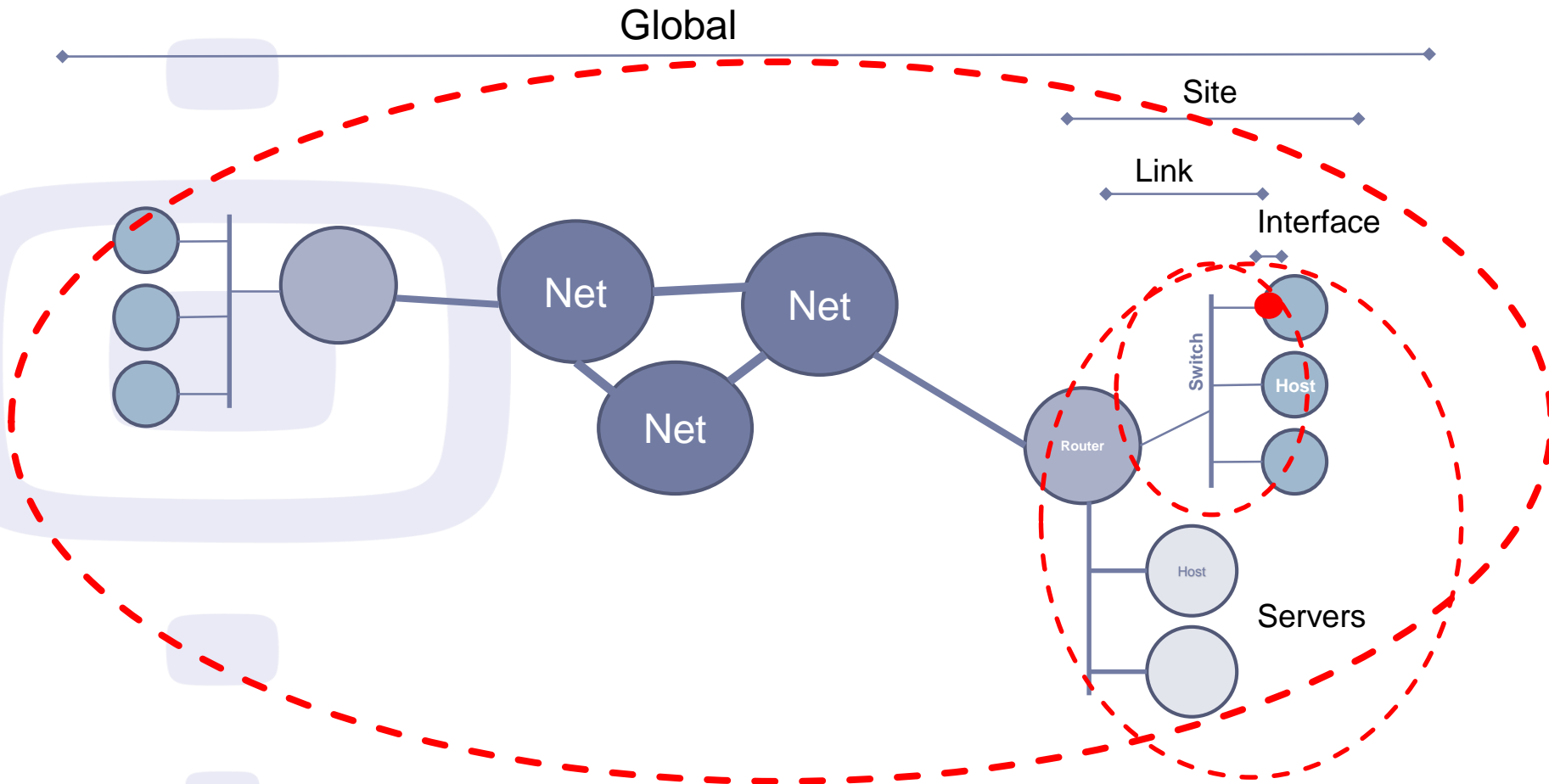
IPv6 addresses: Types & Prefixes

- ▶ Unicast (one-to-one)
 - ▶ Link-local (FE80:: - ▶ Unique Local (ULA) (FC00:: - ▶ IPv4-mapped (::FFFF:IPv4/128)
 - ▶ Global (GUA) (2000::0010)
 - ▶ Site-local (**deprecated**) (FEC0:: - ▶ IPv4-compatible (**deprecated**)(::IPv4/128)
- ▶ Multicast (one-to-many) (FF00::- ▶ Anycast (one-to-nearest) (taken from unicast space)
- ▶ Reserved (Trans. Mechs, documentation (2001:db8::- ▶ There are no BROADCAST addresses -> well-known multicast (FF02::1, FF02::2)

IPv6 Address Scope (I)

- ▶ **Address Scope:** Defines where the IPv6 address is valid as a unique identifier for one or more interfaces.
- ▶ Unspecified address (::/128): only one without scope
- ▶ Scope defined in two ways:
 - ▶ **By definition: Unicast Addresses.** Belongs to a prefix/range with an associated scope. For example:
 - ▶ **Loopback:** (::1/128) link-local scope in an imaginary link to which the virtual loopback interface is connected
 - ▶ **Link-local [RFC4291]:** (fe80::/10) Valid on a local link o layer two domain
 - ▶ **ULA:** (fc00::/7) Scope of a site (site-local)
 - ▶ **Global:** By exclusion, global scope for all the addresses not belonging to previous categories.
 - ▶ **By means of a specific field: Multicast addresses.**
 - ▶ Different values: 1 (Interface-Local) , 2 (Link-Local), 4 (Admin-Local), 5 (Site-Local), 8 (Organization-Local) y E (Global). Other values are reserved or not assigned.

IPv6 Address Scope (II)

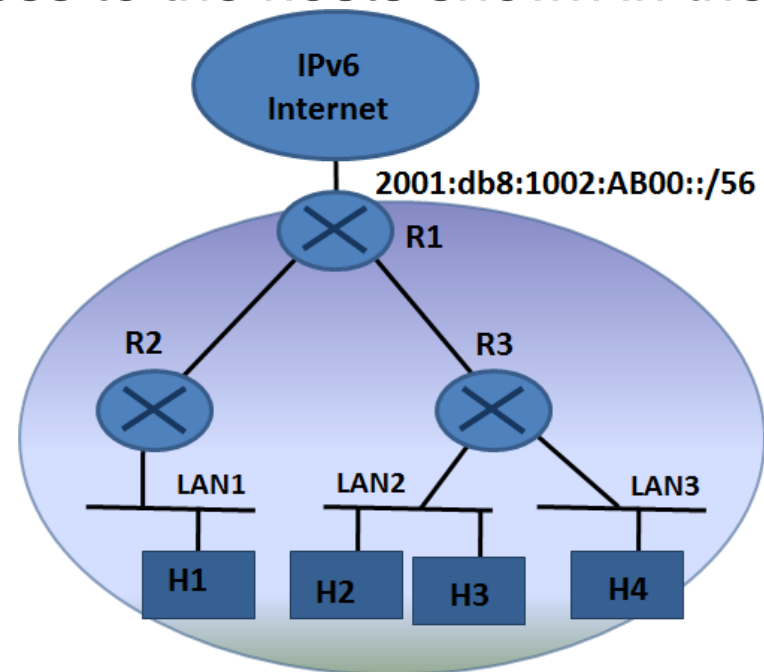


IPv6 addresses: Types (II)

- ▶ Which IPv6 addresses will you use?
- ▶ For sure:
 - ▶ Link-local
 - ▶ Multicast (link-local scope, including SN, ff02::1)
 - ▶ Loopback & Unspecified
- ▶ Probably (or you should)
 - ▶ GUA
- ▶ Maybe
 - ▶ ULA
 - ▶ Multicast (other scopes)
 - ▶ IPv4-mapped (transition mechs.)
 - ▶ Reserved (transition mechs., documentation for tests, etc.)

IPv6 addresses: Exercise (I)

- ▶ **Short Exercise with IPv6 Addresses:**
 - ▶ You have the IPv6 prefix `2001:0db8:1002:AB00::/56`
 - ▶ Take three /64 prefixes from it to assign to three different LANs: LAN1, LAN 2 and LAN3
 - ▶ Give a complete IPv6 address to the hosts shown in the figure: H1, H2, H3, and H4



IPv6 addresses: Exercise (II)

- ▶ Fill the table:

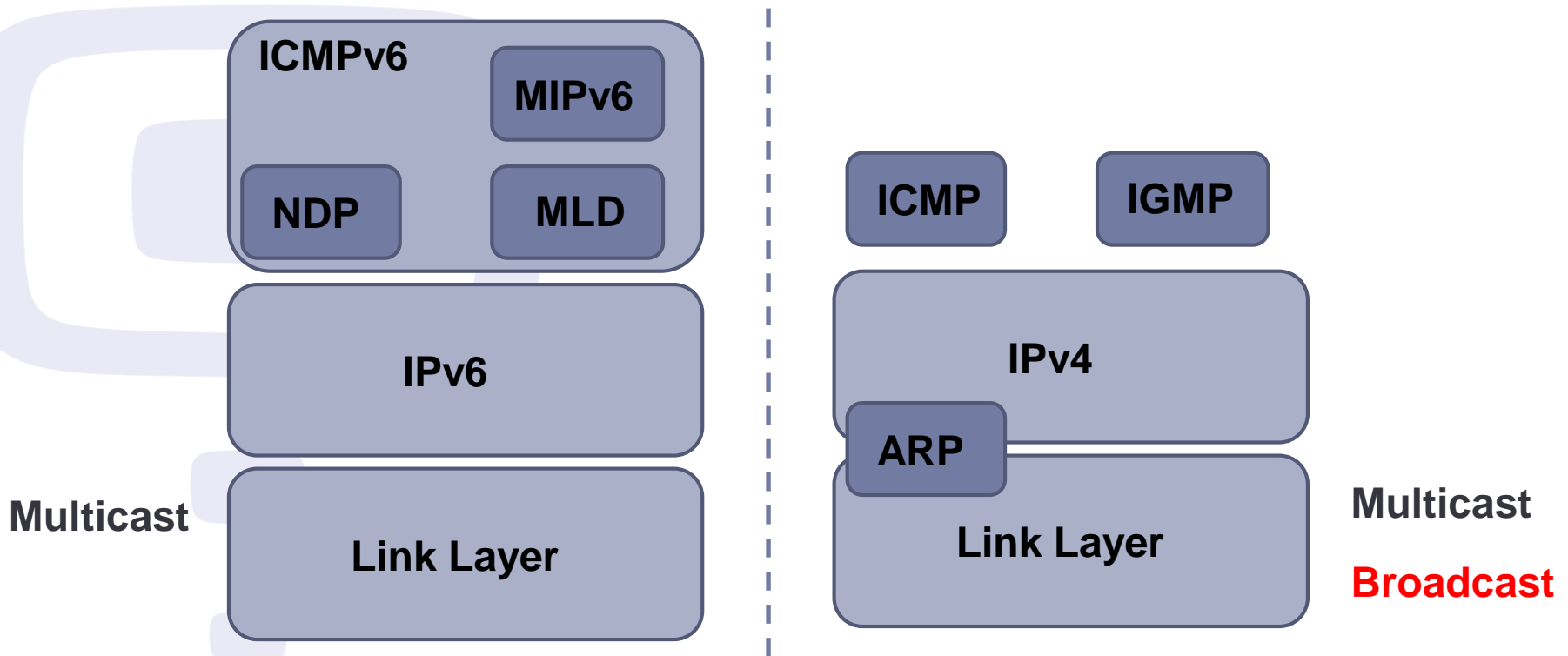
Description	Prefix/Address
LAN1	/64
LAN2	/64
LAN3	/64
H1	
H2	
H3	
H4	

IPv6 addresses: Exercise (III)

- ▶ Start with /56 prefix -> you have to divide into /64s
 - ▶ 2001:db8:1002:AB00::/56 ->
2001:0db8:1002:AB00:0000:0000:0000:0000
 - ▶ : 1010 1011 0000 0000 : (binary) :AB00:
 - ▶ : 1010 1011 0000 0001 : (binary) :AB01:
 - ▶ : 1010 1011 0000 0010 : (binary) :AB02:
 - ▶ : . . . : ...
 - ▶ : 1010 1011 0000 1111 : (binary) :AB0F:
 - ▶ : . . . : ...
 - ▶ : 1010 1011 1111 1111 : (binary) :ABFF:
- ▶ I've got $2^8 = 256$ /64 prefixes: 2001:db8:1002:ab00::/64, 2001:db8:1002:ab01::/64, ... 2001:db8:1002:abFF::/64

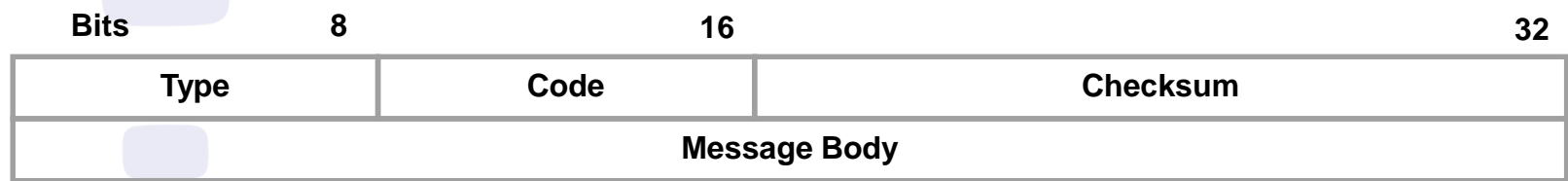
IPv6 Protocols and Autoconfiguration (I)

- ▶ ICMPv6 fundamental part of IPv6



IPv6 Protocols and Autoconfiguration (II)

- ▶ It's used for several things, both:
 - ▶ Locally on the LAN: **NDP, MLD**
 - ▶ On the Internet: Fragmentation, detect other errors
- ▶ You should be careful when filtering
- ▶ Two type of messages:
 - ▶ Error: Destination unreachable, packet too big, time exceeded, parameter problem (type = 0 ... 127)
 - ▶ Informative: echo request, echo reply (type = 128 ... 255)



IPv6 Protocols and Autoconfiguration (III)

ICMPv6 Error Messages

- ▶ Destination Unreachable (type = 1, parameter = 0)
 - ▶ No route to destination (code = 0)
 - ▶ Communication with destination administratively prohibited (code = 1)
 - ▶ Beyond scope of source address (code = 2)
 - ▶ Address Unreachable (code = 3)
 - ▶ Port Unreachable (code = 4)
 - ▶ Source address failed ingress/egress policy (code = 5)
 - ▶ Reject route to destination (code = 6)
- ▶ Packet Too Big (type = 2, code = 0, parameter = next hop MTU)
- ▶ Time Exceeded (type = 3, parameter = 0)
 - ▶ Hop Limit Exceeded in Transit (code = 0)
 - ▶ Fragment Reassembly Time Exceeded (code = 1)
- ▶ Parameter Problem (type = 4, parameter = offset to error)
 - ▶ Erroneous Header Field (code = 0)
 - ▶ Unrecognized Next Header Type (code = 1)
 - ▶ Unrecognized IPv6 Option (code = 2)

Path MTU Discovery (I)

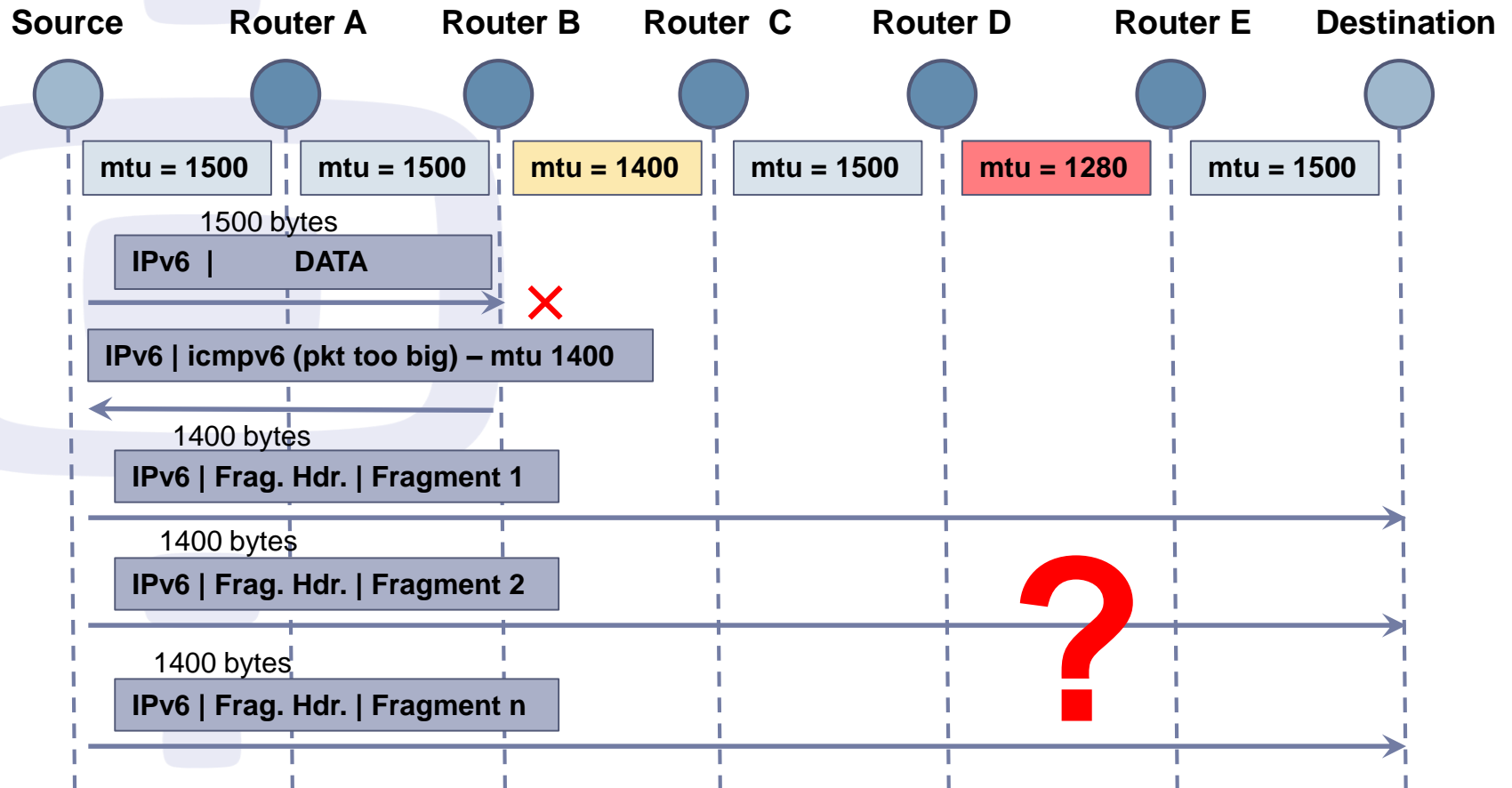
- ▶ **MTU: Maximum Transmission Units**
 - ▶ Link MTU: maximum number of bytes of IP packet
 - ▶ Path MTU: minimum link MTU from source to destination
- ▶ In IPv6 the minimum link MTU is 1280 bytes (v4 68 bytes)
- ▶ In IPv6 this is important because:
 - ▶ Fragmentation process changes: extension header
 - ▶ Encapsulation frequently used: *overhead* reduces available MTU

Path MTU Discovery (II)

- ▶ Implementations do Path MTU Discovery sending packets bigger than 1280 bytes.
 - ▶ For each destination, starts assuming the MTU of first hop
 - ▶ If the packet reaches a link with MTU smaller than its size, an ICMPv6 “packet too big” packet is sent to the source, with information about the MTU of that link. That MTU value is saved for that specific destination
 - ▶ Eventually, saved MTU values are discarded to detect possible changes on the MTU values for different destinations
- ▶ In constrained implementations, PMTU-D could be omitted, if it’s detected that 1280 bytes packets could reach a destination

Path MTU Discovery (III)

- ▶ In IPv6 fragmentation is done in source node



IPv6 Protocols and Autoconfiguration (IV)

ICMPv6 Informative Messages

- ▶ Echo Request (type =128, code = 0)
- ▶ Echo Reply (type =129, code = 0)
- ▶ MLD (Multicast Listener Discovery) Messages:
 - ▶ Query, Report, Done (Like IGMP for IPv4)
- ▶ NDP Messages:
 - ▶ NS (Neighbor Solicitation)
 - ▶ NA (Neighbor Advertisement)
 - ▶ RS (Router Solicitation)
 - ▶ RA (Router Advertisement)
 - ▶ Redirect

IPv6 Protocols and Autoconfiguration (V)

- ▶ **NDP: Neighbor Discovery Protocol**
- ▶ Used for hosts-hosts and routers-hosts communication
- ▶ It offers several services on a LAN:
 - ▶ Discovery of routers, network prefixes, network parameters
 - ▶ Autoconfiguration
 - ▶ Address Resolution
 - ▶ DAD (Duplicate Address Detection)
 - ▶ NUD (Neighbor Unreachability Detection)
- ▶ It only uses 5 type of ICMPv6 packets:
 - ▶ **RA:** Router Advertisement
 - ▶ **RS:** Router Solicitation
 - ▶ **NA:** Neighbor Advertisement
 - ▶ **NS:** Neighbor Solicitation
 - ▶ Redirect

NS/NA

- ▶ A Host will send NS:

1. To determine the MAC address associated with an IPv6 address: Dest. Addr. Multicast Solicited Node (Address Resolution = ARP IPv4)
2. To check reachability: Dest. Addr. Unicast

- ▶ A Host will send NA:

1. Answer to NS
2. To quickly send new information (Unsolicited)

Neighbor Solicitation Format

- ▶ NSs to determine MAC. Own MAC address is sent

Bits	8	16	32
Type = 135	Code = 0	Checksum	
Reserved = 0			
Target Address			
Options ...			

- ▶ Target Address: IPv6 address that generated the request. Could not be a multicast address.
- ▶ Possible Options: Source Link-Layer Address

Neighbor Advertisement Format

Bits	8	16	32
Type = 136		Code = 0	Checksum
R	S	O	Reserved = 0
Target Address			
Options ...			

- ▶ **Flags:**
 - ▶ **R: Router Flag**=1 sending node is a router
 - ▶ **S: Solicited Flag**=1 sent as an answer to a NS
 - ▶ **O: Override Flag**=1 indicating caches should be updated
- ▶ **Target Address (can't be a multicast address):**
 - ▶ Solicited NAs = "Target Address" of NS
 - ▶ Unsolicited NA: IP address which MAC address has changed
- ▶ **Possible Options: Target Link-Layer Address (MAC of Tx).**

RS/RA

- ▶ A Host will send RS
 1. When bring up an interface: Dest. Addr = Well known multicast address of all routers
- ▶ A Router will send RA:
 1. As an answer to RS
 2. Periodically to inform about network parameters

Router Solicitation Format

Bits	8	16	32
Type = 133	Code = 0	Checksum	
Reserved = 0			
Options ...			

- ▶ Possible Options: Source Link-Layer Address.

Router Advertisement Format (I)

Bits	8		16		32
Type = 134	Code = 0			Checksum	
Cur Hop Limit	M	O	Reserved = 0	Router Lifetime	
Reachable Time					
Retrans Timer					
Options ...					

- ▶ Cur Hop Limit: default value to be used as Hop Limit in IPv6 header for packets sent
- ▶ M: 1-bit "Managed address configuration" flag
- ▶ O: 1-bit "Other configuration" flag
- ▶ Router Lifetime: time the router could be used as default router
- ▶ Reachable Time: time node assumes a neighbor is reachable after having received a reachability confirmation (used in NUD)
- ▶ Retrans Timer: time (ms) between retransmitted NS (used in NUD, AR)
- ▶ Possible Options: Source LinkLayer Address, MTU, Prefix Information, RDNSS, Flags Expansion (RFC5175)

Router Advertisement Format (II)

- ▶ Options: TLV (Type-Length-Value)
- ▶ Example: Prefix Information
 - ▶ **L**(1bit): **on-link flag**=1 indicates if prefix could be used for “on-link determination”
 - ▶ **A**(1bit): **autonomous address-configuration flag**=1 indicates if prefix could be used for stateless address autoconfiguration.
 - ▶ **Valid Lifetime**: Time in secs. Prefix is valid for on-link determination. Used for stateless address autoconfiguration as well.
 - ▶ **Preferred Lifetime**: Time in secs. that addresses generated with this prefix using SLAAC are in preferred state
 - ▶ **Prefix** (128 bits): IPv6 Address or prefix.

Bits	8	16	24	32		
	Type = 3	Length = 4	Prefix Length	L	A	Reserved1 = 0
Valid Lifetime						
Preferred Lifetime						
Reserved2 = 0						
Prefix						

IPv6 Protocols and Autoconfiguration (VI)

- ▶ Autoconfiguration in general is about automatically configure network parameters, not manually
- ▶ In IPv4 we only have DHCP
- ▶ In IPv6 there are more options
- ▶ Two scenarios: router or non-router
- ▶ **Router:**
 - ▶ Sends RAs -> M and O Flags -> four combinations
 - ▶ Hosts should look at M and O flags and then start to autoconfigure
 - ▶ M is about IPv6 address, O is about other parameters (DNS, etc.)
 - ▶ We have two “tools” SLAAC (0) and DHCPv6 (1)

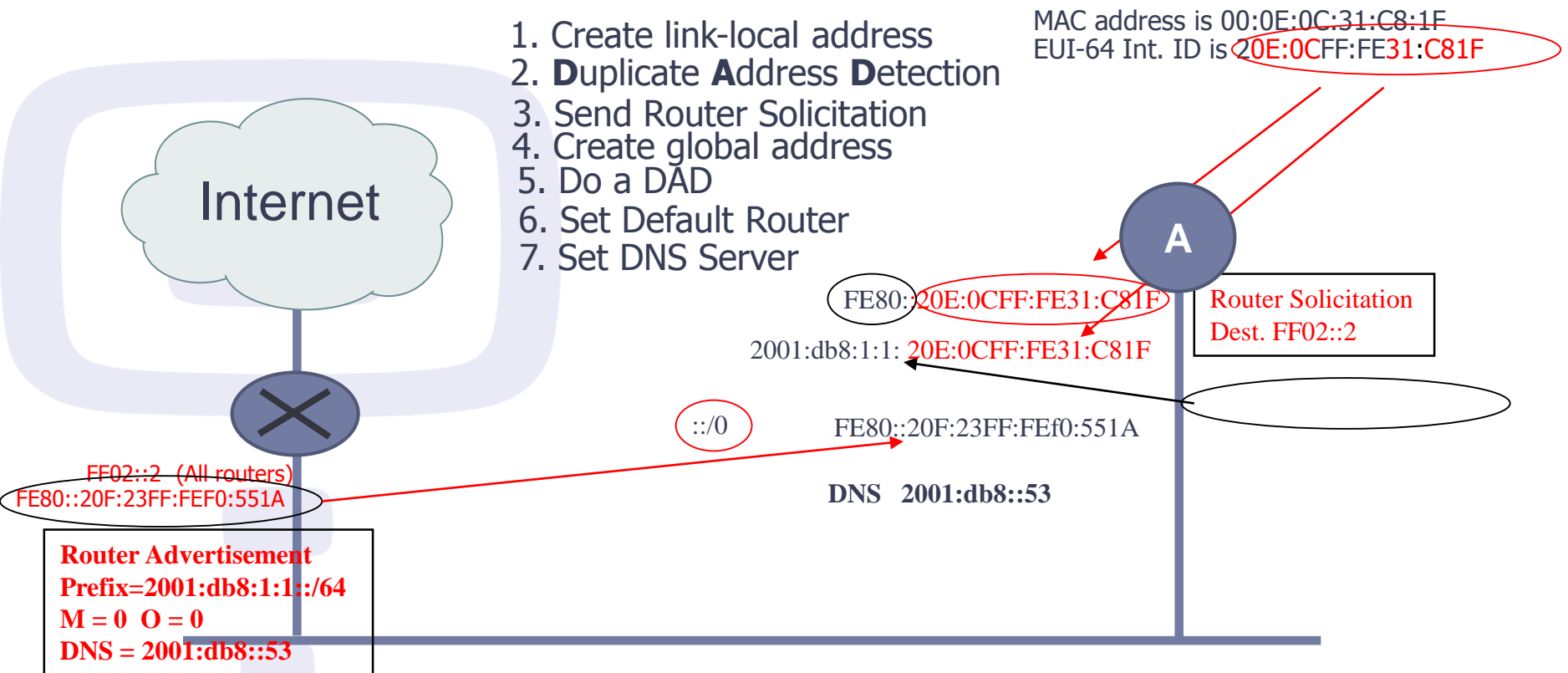
IPv6 Protocols and Autoconfiguration (VII)

- ▶ SLAAC vs. DHCPv6
- ▶ NOTE: Default gateway is learnt from the RA(s) (or manually)

IP / Other	M	O	Comments
SLAAC / SLAAC	0	0	If dual-stack, could use IPv4 for DNS
SLAAC / DHCPv6	0	1	DHCPv6 Stateless
DHCPv6 / SLAAC	1	0	If dual-stack, could use IPv4 for DNS
DHCPv6 / DHCPv6	1	1	Gateway is learnt from RA

IPv6 Protocols and Autoconfiguration (VIII)

▶ Host A attaches to a network with a Router

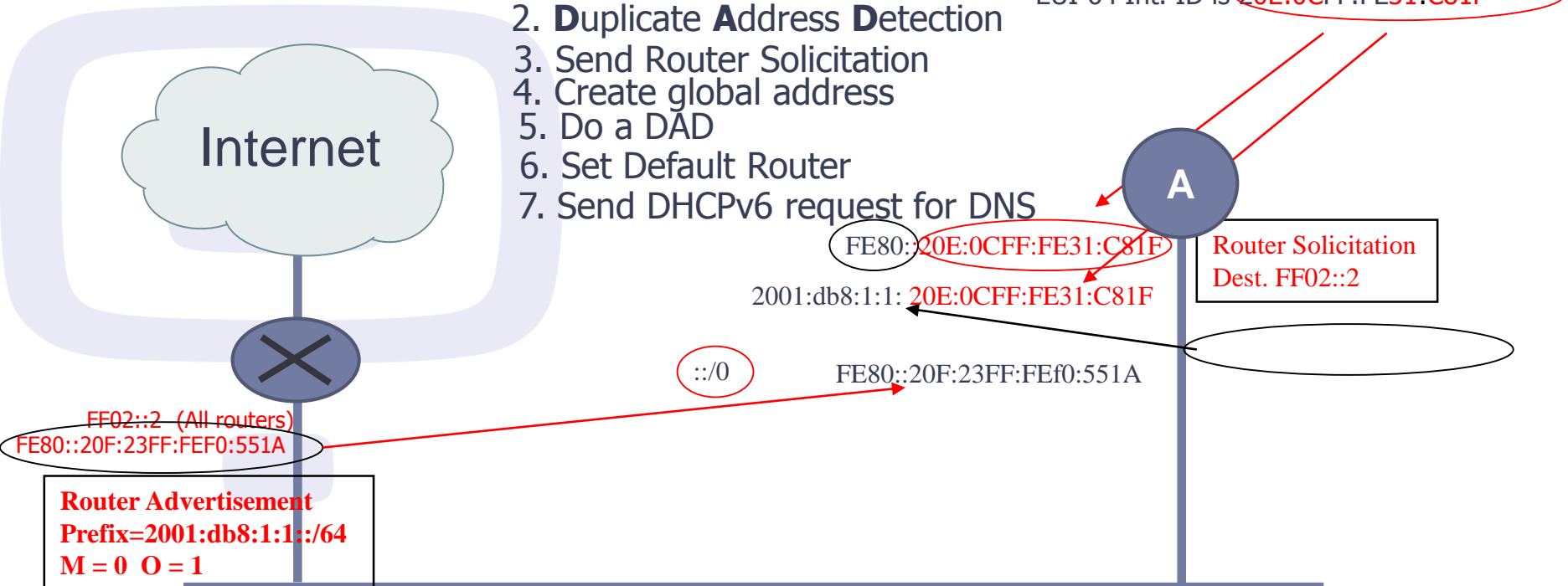


IPv6 Protocols and Autoconfiguration (IX)

- ▶ In practice SLAAC for DNS is not yet available. Use IPv4 for DNS resolution (dual-stack) or DHCPv6 (O = 1)

1. Create link-local address
2. Duplicate Address Detection
3. Send Router Solicitation
4. Create global address
5. Do a DAD
6. Set Default Router
7. Send DHCPv6 request for DNS

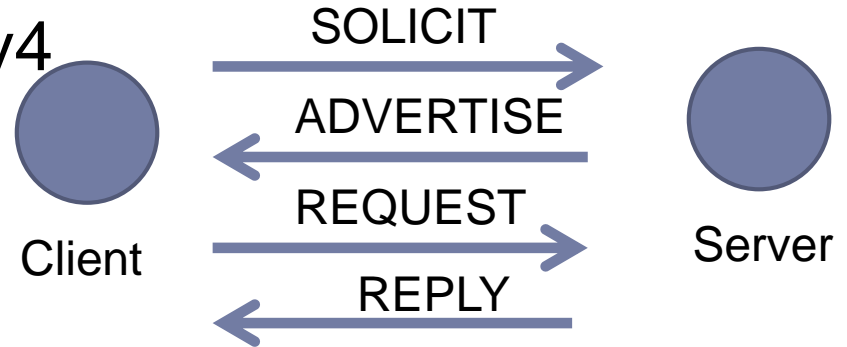
MAC address is 00:0E:0C:31:C8:1F
EUI-64 Int. ID is 20E:0CFF:FE31:C81F



DHCPv6 (I)

- ▶ DHCPv6 works as DHCPv4

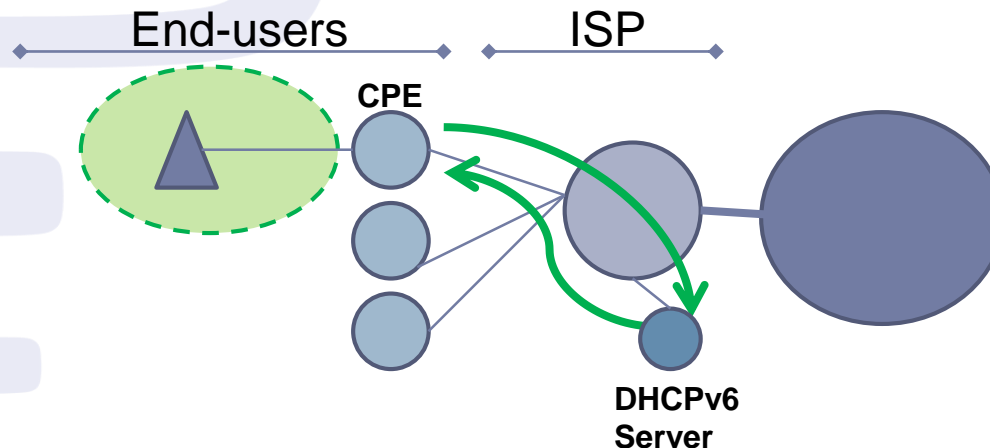
- ▶ Client-server
- ▶ UDP
- ▶ Use of relay



- ▶ DIFFERENCE: Does not provide default gateway
- ▶ Messages names change: SOLICIT, ADVERTISE, REQUEST, REPLY
- ▶ Servers/Relays listen on well-known multicast addresses (FF02::1:2)
- ▶ DHCPv6 stateless: only provides “other” info, not IP

DHCPv6 (II)

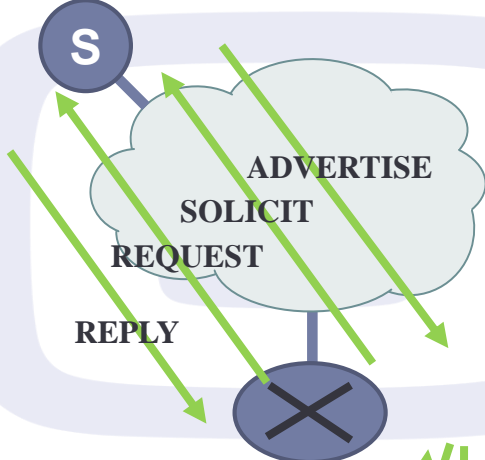
- ▶ DHCPv6-PD (Prefix Delegation)
- ▶ In IPv6 no private IP + NAT. A GUA prefix is needed
- ▶ DHCPv6-PD allows scalable configuration of IPv6 prefixes in routers
- ▶ Same as for IP addresses: client-server, etc.
- ▶ Only changes the requested object: a prefix (IA-PD)
- ▶ Example: CPE connected to an ISP



DHCPv6 (II)

- ▶ Host A connected to network with Router and DHCPv6 relay
- ▶ $M = O = 1$

DHCPv6



1. Creates link-local address
2. Duplicate Address Detection
3. Sends RS
4. SOLICIT/ADVERTISE/REQUEST/REPLY

MAC address is 00:0E:0C:31:C8:1F
 EUI-64 Int. ID is **20E:0CFF:FE31:C81F**

SOLICIT
 Dest. FF02::1:2

Router Solicitation
 Dest. FF02::2

FE80:**20E:0CFF:FE31:C81F**

IP: 2001:db8:1:1::A:B:1002

DNS: 2001:db8:FF::5:3

::/0 FE80::20F:23FF:FEf0:551A

FF02::2 (All routers)
 FF02::1:2 (DHCPv6 Relay)
 FE80::20F:23FF:FEf0:551A

Router Advertisement
M = 1 O = 1

REPLY




REQUEST

ADVERTISE

¡Thanks!

Questions?



- ▶ **Contact:** info@nodo6.com / training@nodo6.com
- ▶  <http://www.nodo6.com>
- ▶  <https://www.linkedin.com/company/nodo6>
- ▶  https://twitter.com/NODO6_RRSS