RIPE NCC
RIPE NETWORK COORDINATION CENTRE

Introduction to IPv6 - I

Basics of IPv6

Alvaro Vives | 26 June 2017 | Workshop on Open Source Solutions for the IoT

Contents

- Digital Data Transmission
- Packet-Switched Networks
- Layered Model
- IPv4 and IPv6 basics
  - IPv4 Header
  - IPv6 Header
  - Differences
- IPv6 addresses
  - Types of IPv6 addresses
  - IPv6 Notation
  - Interface Identifier (IID)
  - IPv6 Addresses Exercise

Digital Data Transmission (1)

- Objective: send information from one place/device to another
- Different type of info, through different transport networks
- Need 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 by ASCII characters => 256 (2^8)

Digital Data Transmission (2)

- 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

- 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

Packet-Switched Networks (1)

- Digital pieces of information put in packets

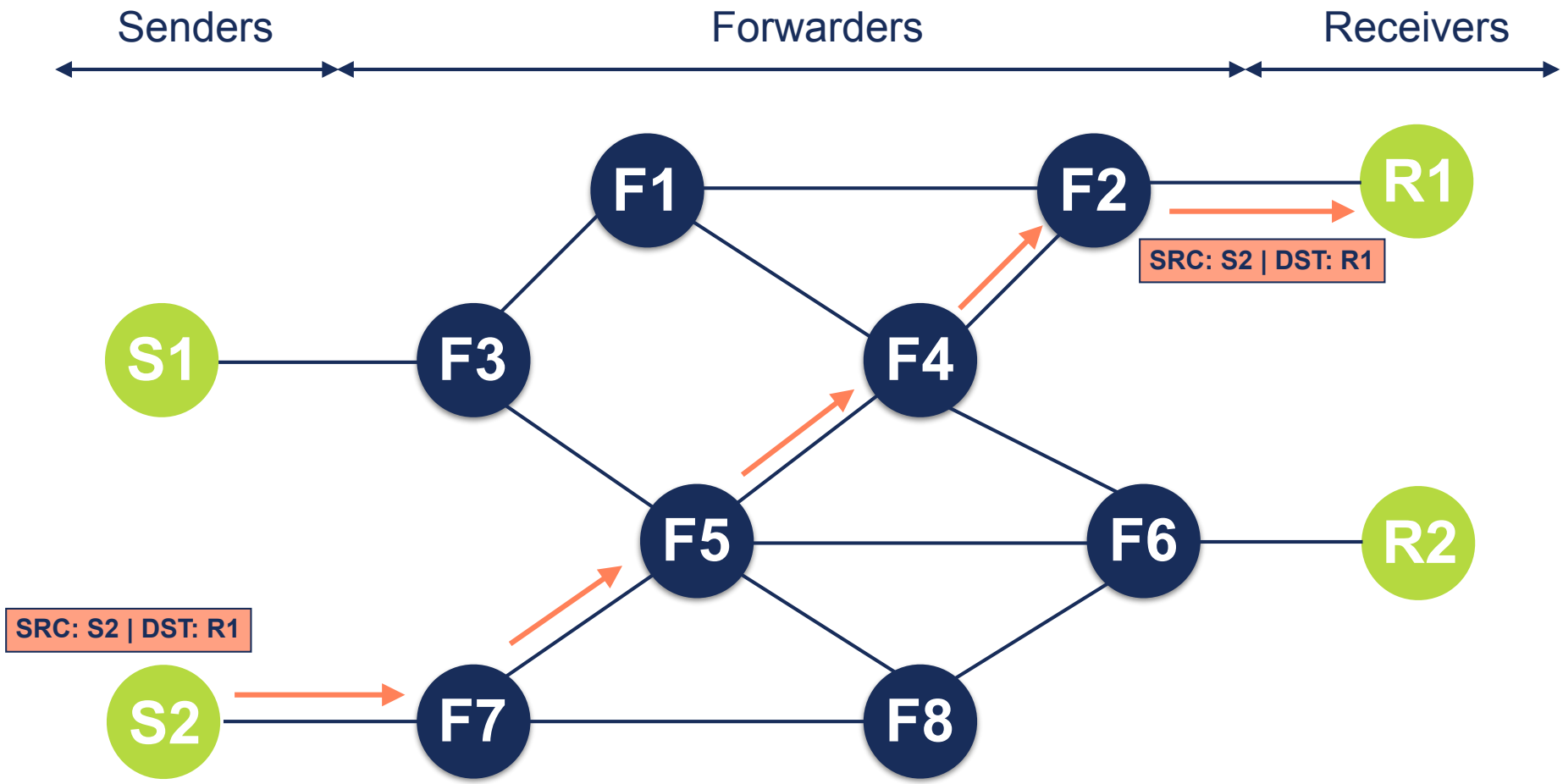
- Packets sent over packet-switched networks:
  - Paths can vary
  - Shared resources (best effort)
  - Communication can start at any moment (example: postal mail, Internet)

Packet-Switched Networks (2)

Basic elements on a switched network:

- **Sender**: Generates the info to be sent to a receiver. Should codify the message.
- **Receiver**: Is the destination of the information sent by the sender. Should decode the message.
- **Forwarder**: Nor the origin or the destination of the information. Just receive and forward the information in its path to the destination
- **Identification**: Each element in the switched network should be uniquely identified

Packet-Switched Networks (3)



Layered Model (1)

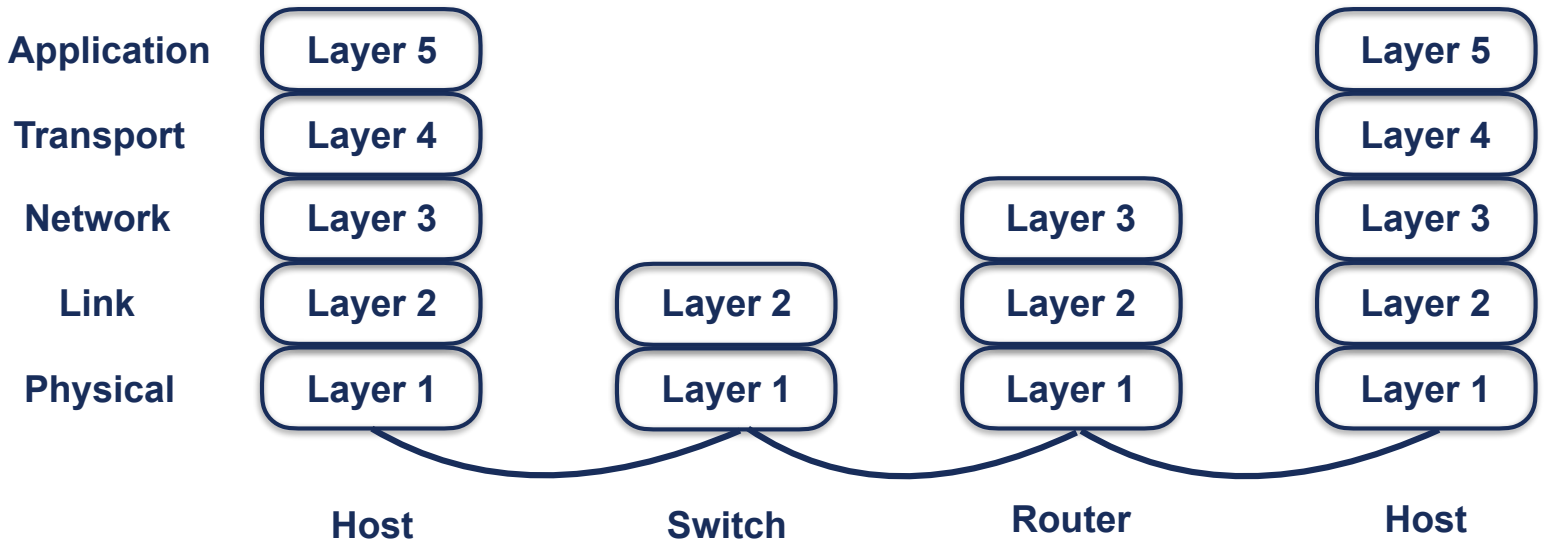
Let's define things:

- **Layered model**: physical, link, network, etc. each one is in charge of different things/services
- **Network elements**: Node, host, router, server
- **Addresses**: link layer, network layer

- **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

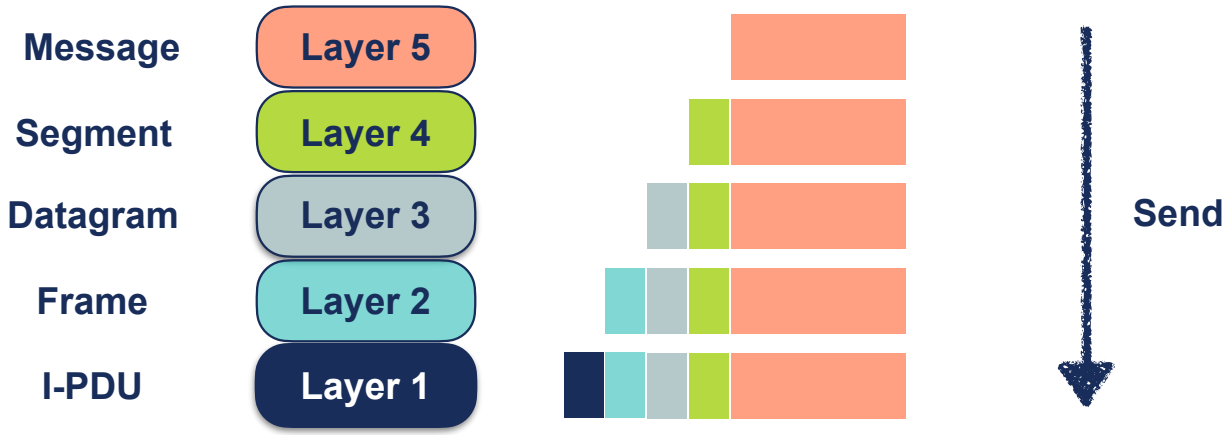
Layered Model (2)

- TCP/IP layered model -> Used in Internet



Layered Model (3)

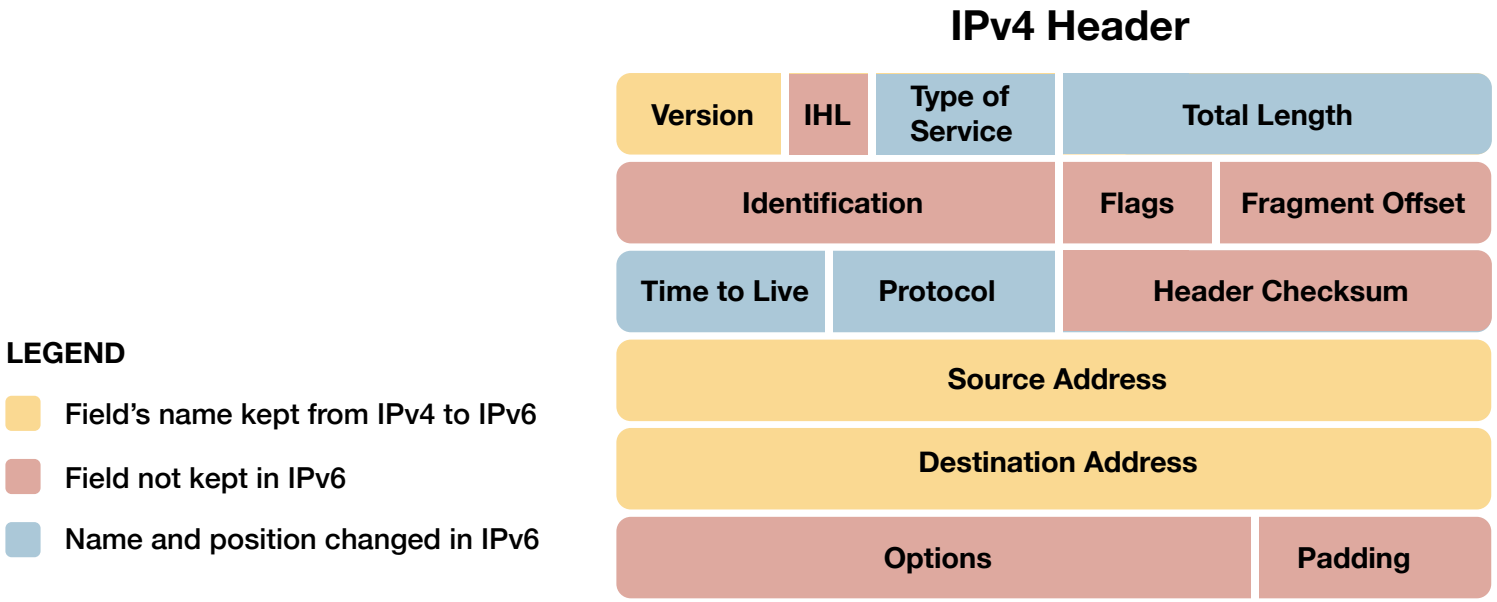
- PDU: Protocol Data Unit



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

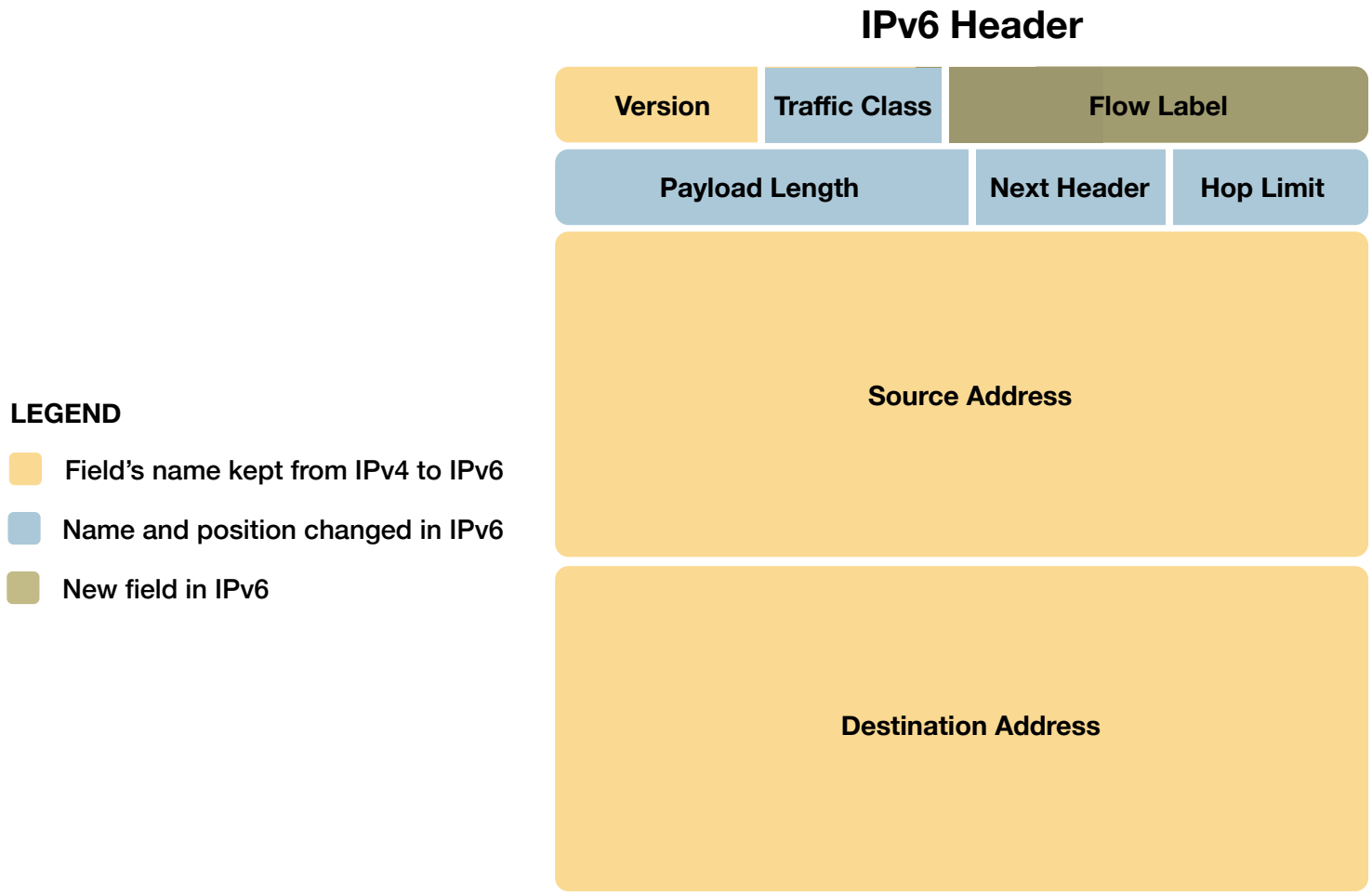
IPv4 and IPv6 basics (1)

- IPv6 is an evolution of IPv4



IPv4 and IPv6 basics (2)

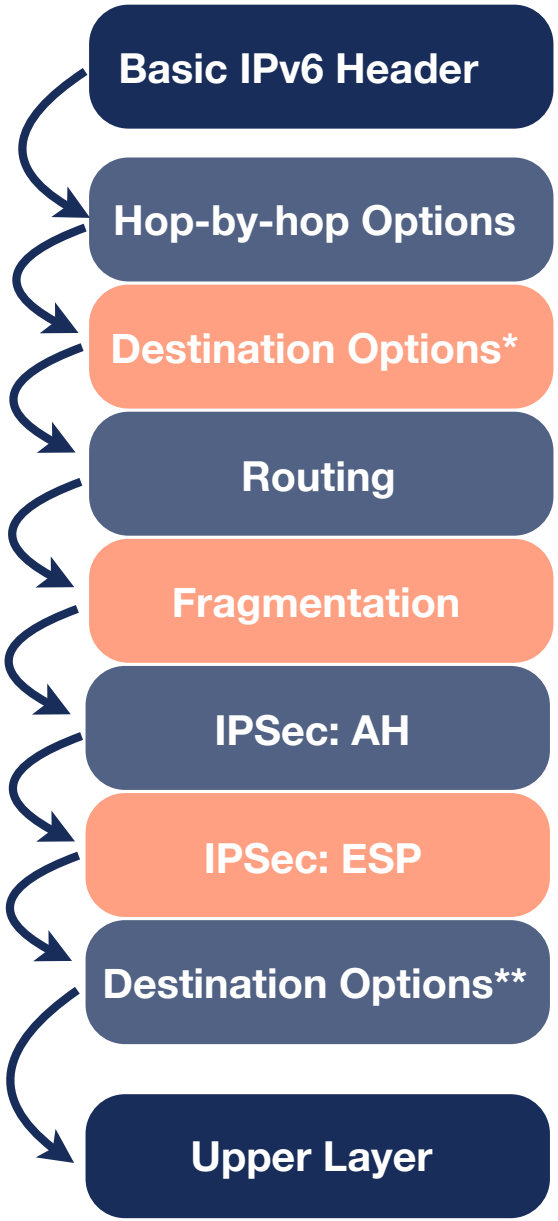
- Simplified, fixed-length, 64 bits aligned -> complexity from core to border



IPv4 and IPv6 basics (3)

- 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 (4)



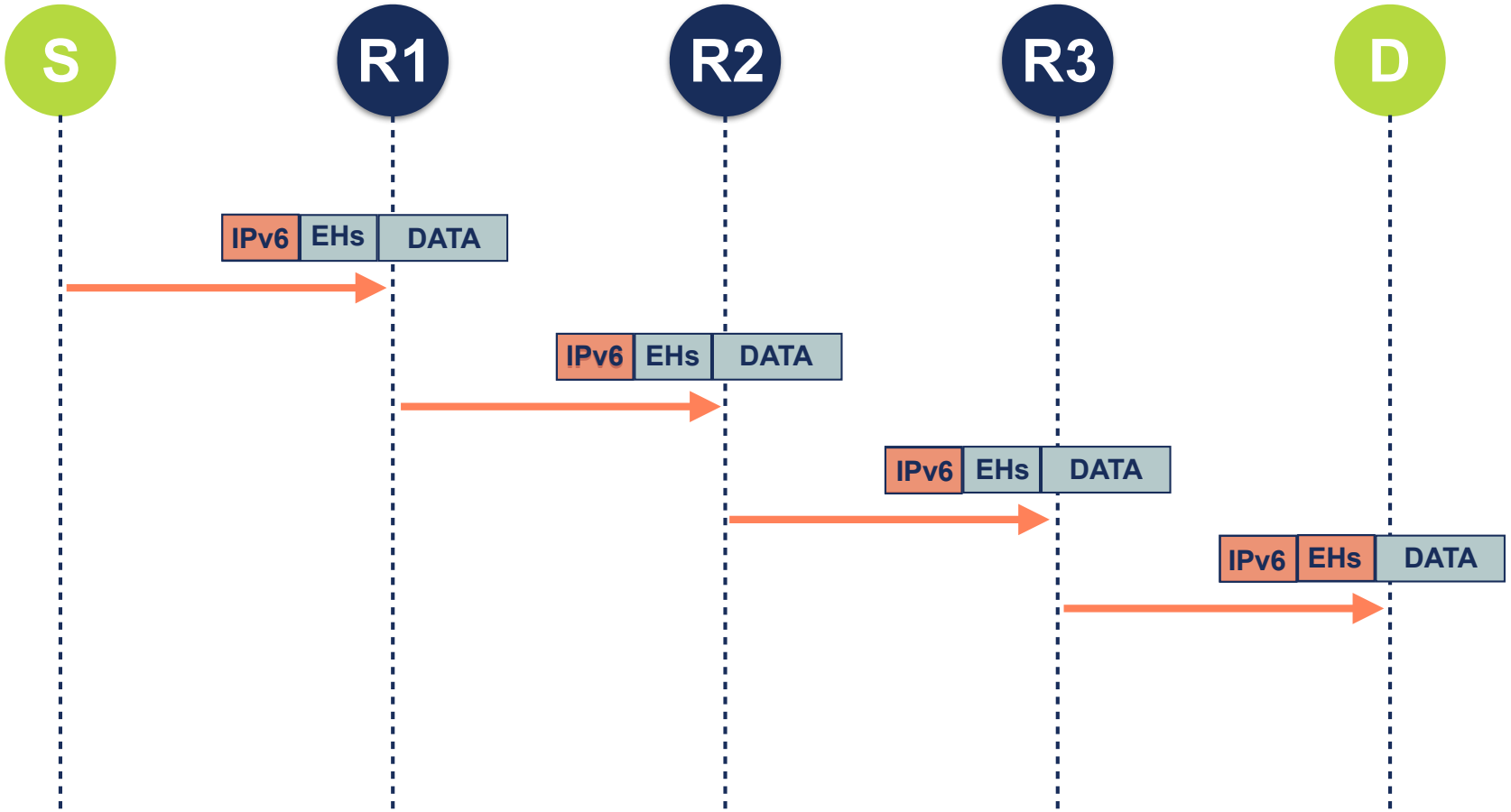
- Fixed: Types and order
- Flexible use
- Processed only at endpoints
  - Exceptions: Hop-by-hop (and Routing)
- Only appear once
  - Exception: Destination Options

* Options for IPs in routing header

** Options for destination IP

IPv4 and IPv6 basics (5)

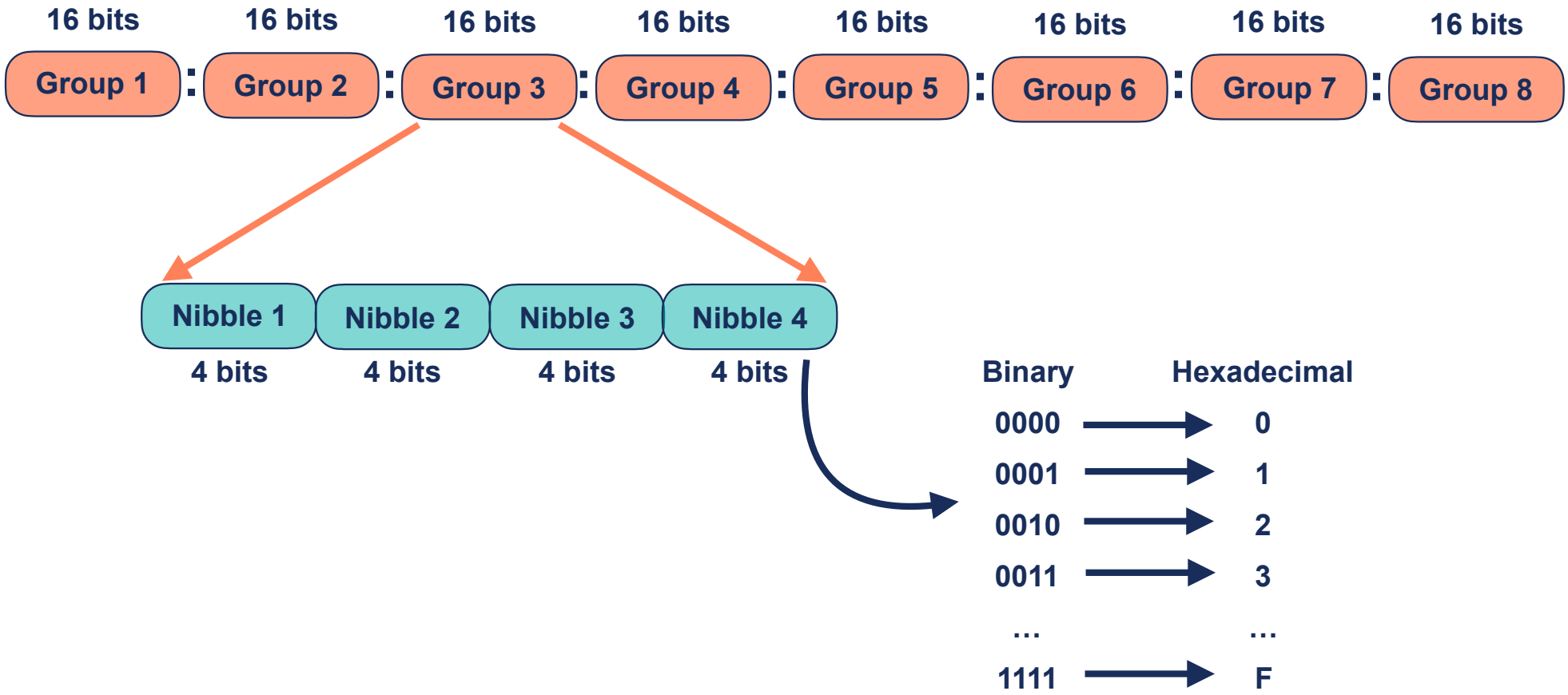
- Basic IPv6 header is processed in all hops
- Extension headers processed at destination (exception Hop-by-hop)



Types of IPv6 Addresses

- 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 (1)



IPv6 addresses: Notation (2)

- IPv6 address notation rules:
  - 8 Groups of 16 bits separated by “:”
  - Hexadecimal notation of each nibble (4 bits) ->
  - No case sensitive

Binary	Hexadecimal
0000	0
0001	1
0010	2
0011	3
…	…
1111	F

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

IPv6 addresses: Notation (3)

- 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 -> ?

IPv6 addresses: Notation (4)

- 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

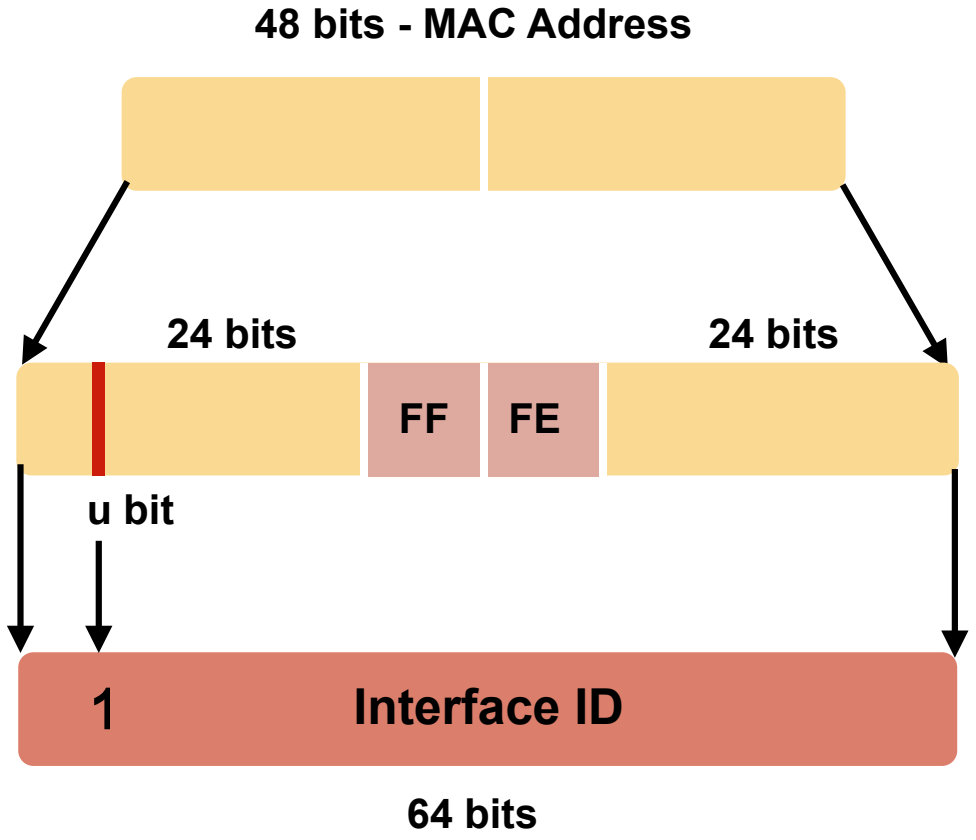
- Non-prefix bits (rightmost) used for subneting
  - Example: I'll take the first two /52 prefixes out of 2001:db8:abcd::/48
  - 2001:0db8:abcd:**0**000:0000:0000:0000:0000 -> **2001:db8:abcd:0**000::/52
  - 2001:0db8:abcd:**1**000:0000:0000:0000:0000 -> **2001:db8:abcd:1**000::/52

IPv6 addresses: IID (1)

- IID could be created in many different ways
  - Automatically from MAC addresses (EUI-64)
  - Automatically using some kind of algorithm (randomly)
  - Manually
  - DHCPv6
- /64 prefix for a LAN
- IIDs generated locally in the host (except DHCP)

IPv6 addresses: IID (2)

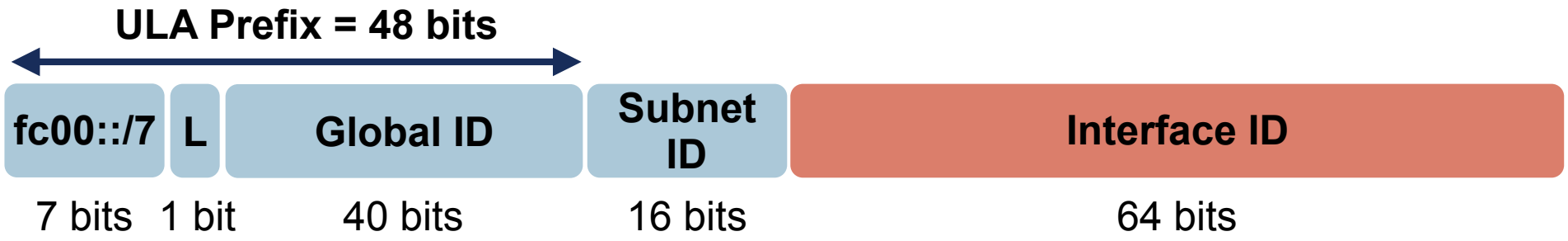
- EUI-64: IID generated from MAC address



IPv6 addresses: Link-local

- Valid only in a link
- Always present in IPv6-enabled interface
- Prefix fe80::/10 (In practice fe80::/64)
- IID generated locally in the host: based on MAC, randomly or anyhow

IPv6 addresses: ULA



- ULA addresses: FC00::/7 Prefix
- L = 1 prefix is locally assigned
- L = 0 may be defined in the future (defined for centrally assigned prefixes)
- Global ID: pseudo-randomly generated
- You'll create a /48 prefix, starting with FD00::/8

IPv6 addresses: Multicast (1)

- 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

IPv6 addresses: Multicast (2)

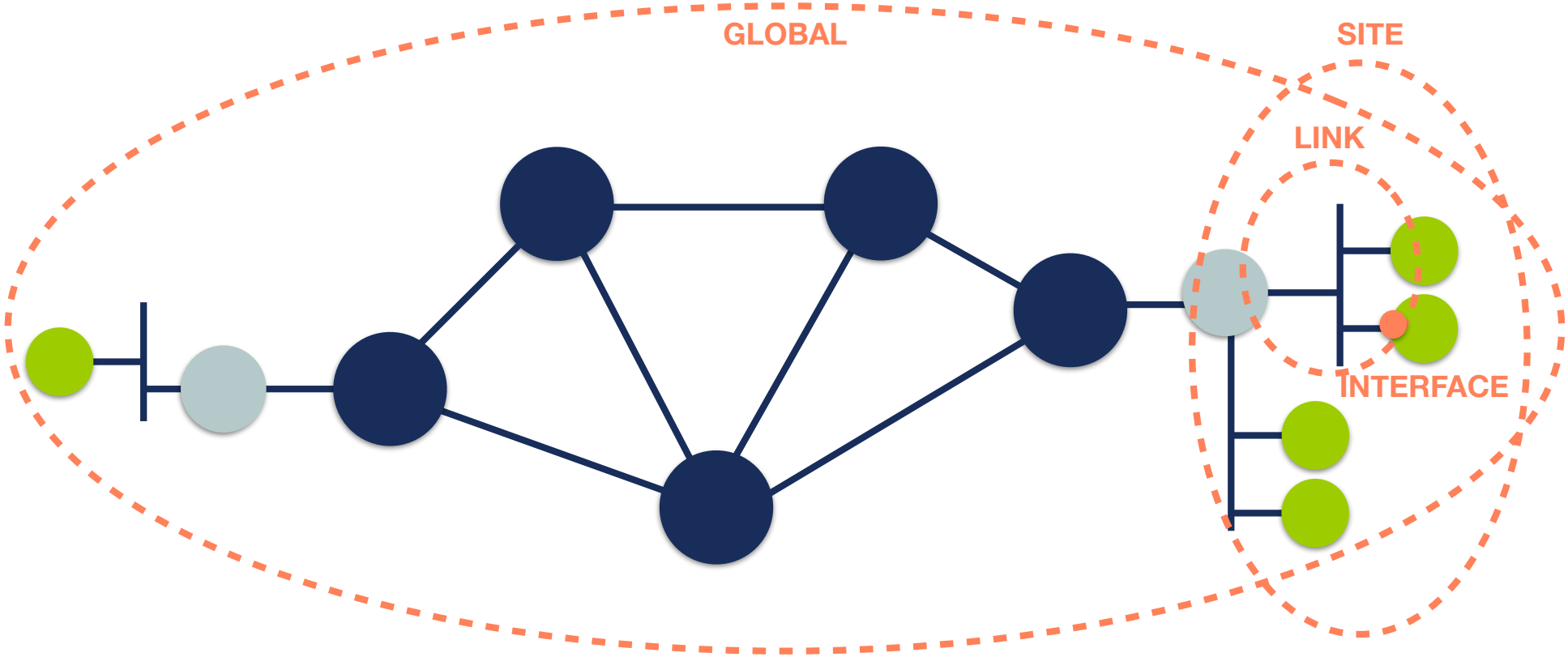
- Substitute of IPv4 Broadcast addresses
- Well-known IPv6 multicast addresses:
  - FF02::1 (all nodes)
  - FF02::2 (all routers)
  - FF02::1:2 (all DHCP-agents)

IPv6 addresses: Types and Prefixes

- Unicast (one-to-one)
  - Link-local (FE80::/10)
  - Unique Local (ULA) (FC00::/7)
  - IPv4-mapped (::FFFF:IPv4/128)
  - Global (GUA) (2000::/3) (binary: 0010)
  - Site-local (deprecated) (FEC0::/10)
  - IPv4-compatible (deprecated) (::IPv4/128)
- Multicast (one-to-many) (FF00::/8)
- Anycast (one-to-nearest) (taken from unicast space)
- Reserved (Trans. Mechs, documentation (2001:db8::/32), loopback (::1/128), unspecified (::/128), etc.)
- No BROADCAST addresses -> well-known multicast (FF02::1, FF02::2)

IPv6 addresses: Scope

- Where the address is valid



fe80::A:b:100

ff01::2

2001:67c:2e:1::c1

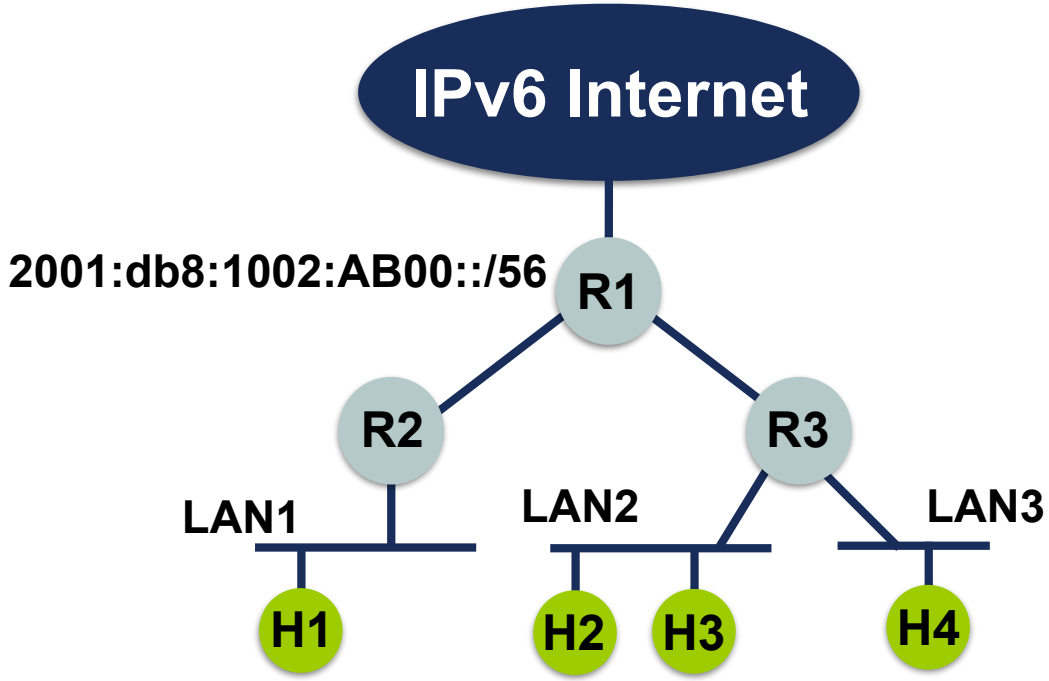
FD00:A:B::100

FF05::1:3

ff02::1

IPv6 addresses: Exercise (1)

- Short Exercise with IPv6 Addresses:
  - Your 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 IPv6 address to hosts: H1, H2, H3, and H4



IPv6 addresses: Exercise (2)

- Complete the table:

Description	Prefix / Address
LAN 1	/64
LAN 2	/64
LAN 3	/64
H1	
H2	
H3	
H4	

IPv6 addresses: Exercise (3)

- Start with /56 prefix -> you have to divide into /64s

2001:0db8:1002:AB00:0000:0000:0000:0000

: **1010 1011** 0000 0000 : (binary) :**AB**00: (hex.)

: **1010 1011** 0000 0001 : (binary) :**AB**01: (hex.)

. . . …

: **1010 1011** 1111 1111 : (binary) :**AB**FF: (hex.)

- In total 2^8 = 256 /64 prefixes:
  2001:db8:1002:ab00::/64, 2001:db8:1002:ab01::/64,
  … 2001:db8:1002:abFF::/64

Questions

avives@ripe.net
@TrainingRIPENCC