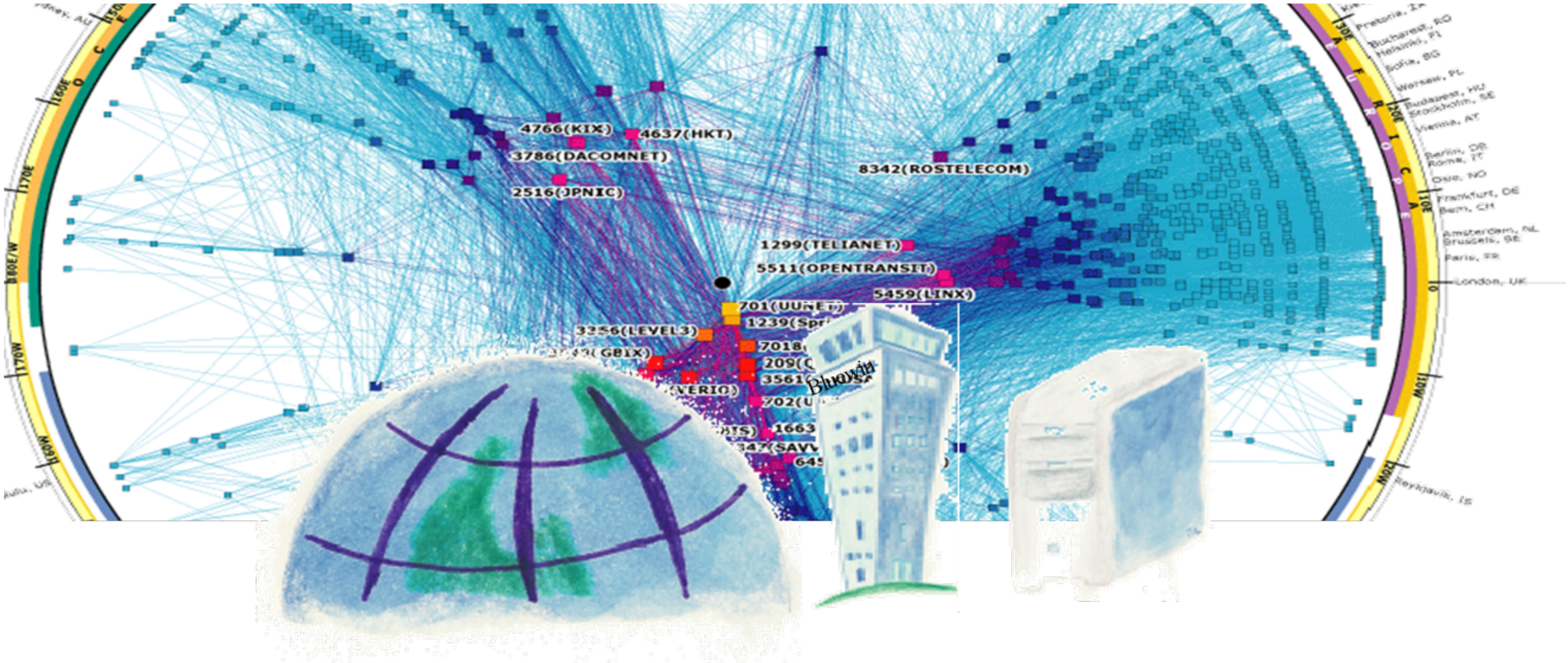


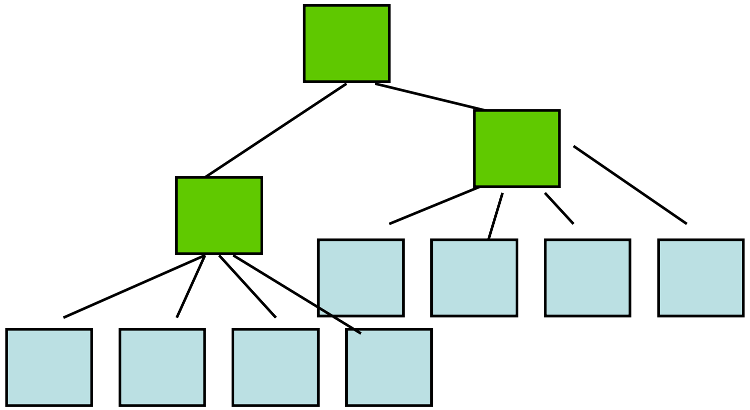
Network Layer
IPv4 and IPv6

2025

EPFL

Recap

communication	Application
end-host connectivity	Transport
all-host, across-LANs interconnection	Network
within-LAN interconnection	MAC
point-to-point transmission of bits	Physical



Recap: most important protocol = IP (Internet Protocol)

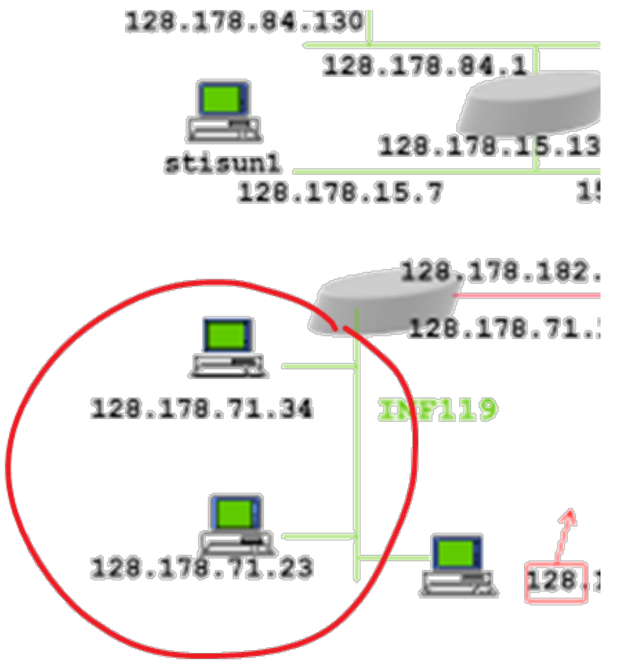
- interconnects multiple *local area networks* (LANs)
- uses packet switching
- delivers packets from a source to a destination via a series of routers
- forwards packets from router to router based on *IP addresses*
- offers *no reliable*-delivery guarantees (*best-effort* approach)
    - packets are briefly stored in routers' buffers
    - packets of the same source-destination flow may follow different routes/paths
    - so, packets may be *dropped*, *delayed* or *reordered*

Recall: IP prefixes and subnet masks

- *subnet* <— a LAN, i.e. a set of devices:
  - connected at the Data-link layer
  - sharing the same IP-address ***prefix***, e.g.: 128.178.71.X

- The prefix is specified using a ***subnet mask***
  (= sequence of bits, where 1s indicate fixed positions of the prefix)
  - e.g. for an EPFL IPv4 LAN, the subnet mask is
    1111 1111 1111 1111 1111 1111 0000 0000
  - The size (in bits) of the prefix is not always the same, e.g.:
    ETHZ IPv4 LANs = 26 bits
    EPFL IPv6 LANs = 64 bits

- Various notations for the subnet mask:
  - dotted, decimal: e.g., address =128.178.71.34, mask = 255.255.255.0
  - "/" (slash): e.g. 128.178.71.34/24
    or 2001:620:618:1a6:0a00:20ff:fe78:30f9/64

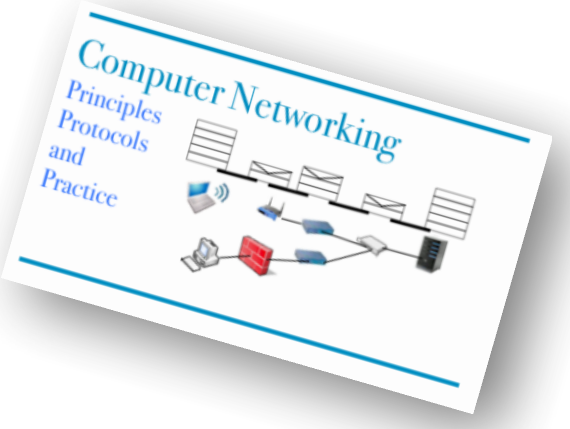


Contents - Internet Protocol (IP)

1. The 2 main rules of IP Unicast
2. IPv4 addresses
3. IPv6 addresses
4. NATs
5. Host configuration
6. Hop Limit and TTL
7. ARP (connection with MAC layer)

Textbook

Chapter 5: The Network Layer



IP Rule #1: Forward pkts according to dest IP address

Why? Goal of IP = interconnect all systems in the world using *IP addresses*

➡ Each IP address should *uniquely* identify a system

What?

1. Assign structured addresses:
    - every network interface has an IP address with *prefix* + *suffix*: e.g. 128.178.71.202
    - interfaces within a *subnet* have the *same prefix* => *same subnet mask*
    - but *different* suffix (a.k.a. host part)

2. Forward packets according to *longest prefix match* principle:
    - every packet contains the destination IP address in its header
    - every system (i.e. **host** = end-system or **router** = intermediate system)
        - has a **forwarding table** (= routing table) and
        - forwards each packet based on the *closest* table entry to the destination IP address

Longest prefix match (= *closest* matching table entry)

R1's forwarding table

to…	outgoing interface
B.*	2
A.*	0

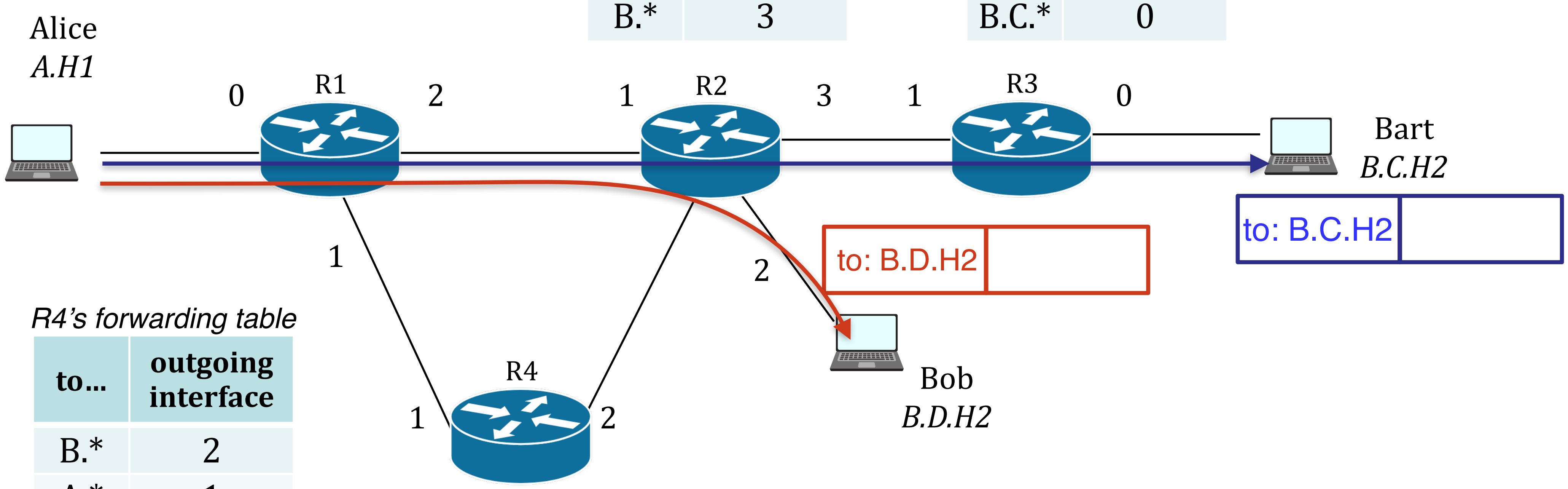
R2's forwarding table

to…	outgoing interface
A.*	1
B.D.*	2
B.*	3

R3's forwarding table

to…	outgoing interface
A.*	1
B.D.*	1
B.C.*	0

Letters in addresses denote prefixes; e.g.:
A = 2001:0620:0008::/48
B.C = 2001:0620:0618:01a5::/64
B.D = 2001:0620:0618:01a6::/64



R4's forwarding table

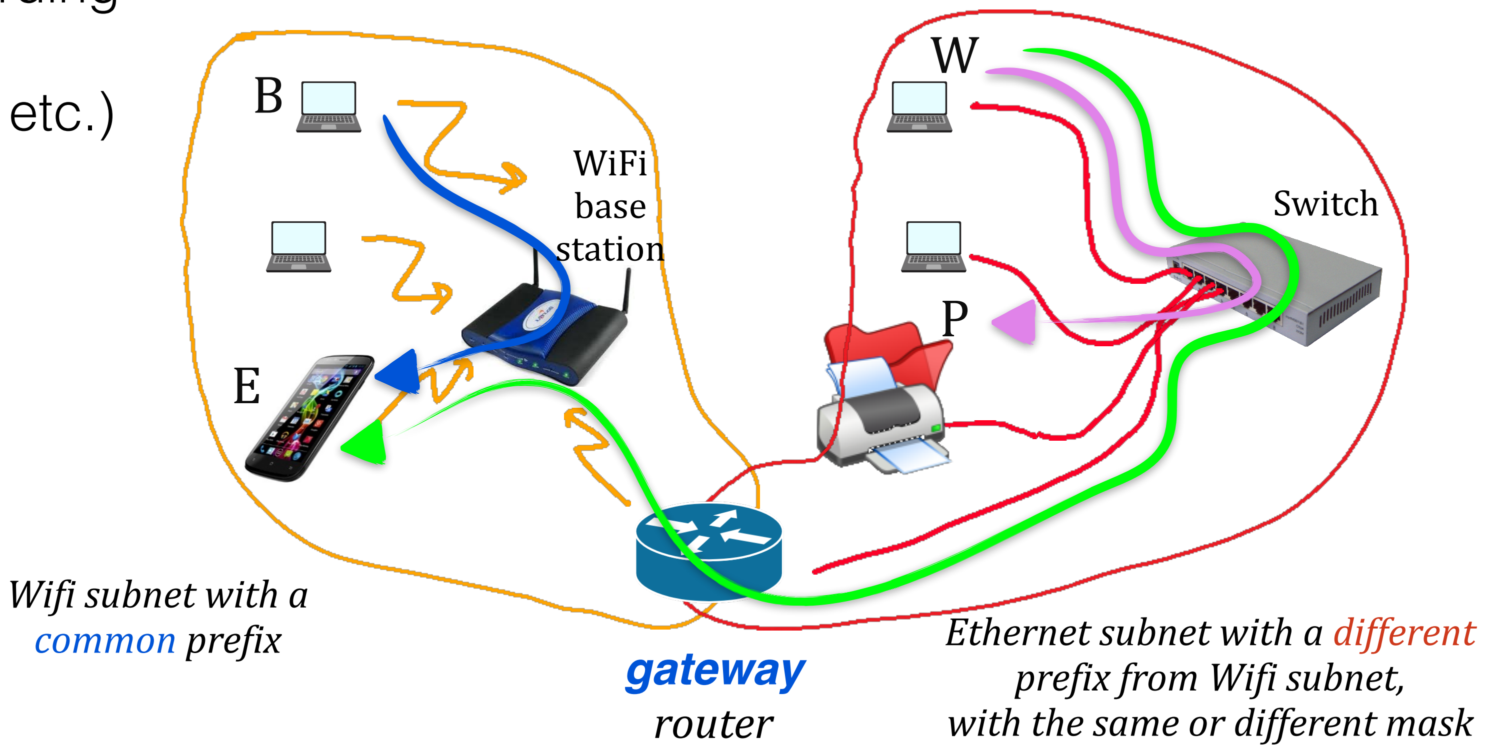
to…	outgoing interface
B.*	2
A.*	1

- Benefit: addresses can be *aggregated*, tables can be *compressed*

IP Rule #2: Only routers interconnect different LANs/subnets

- ➡ *Between* LANs/subnets, use *routers*
- ➡ *within* each subnet, do *not*

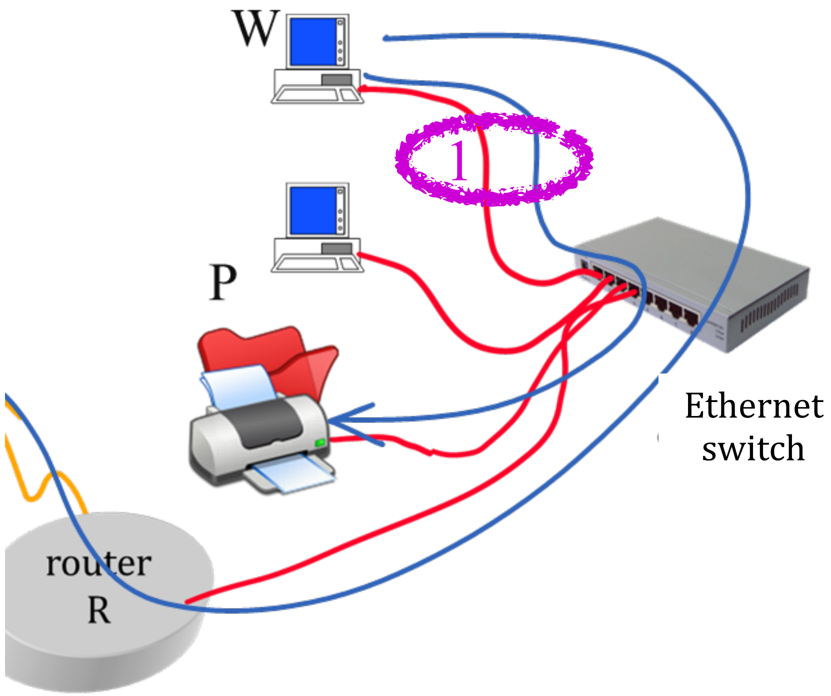
  instead, use data-link-layer forwarding devices within subnet
  (e.g. switches, wifi base stations, etc.)



E.g.:

- Traffic B ↔ E and W ↔ P does not go through router
- Traffic W ↔ E goes through router

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 switch

C. There is no destination IP address in the packet since communication is inside the subnet and does not go through a router

Go to web.speakup.info or download speakup app

Join room 87072

Solution

Answer A

The IP address is *always present* in the IP header, even if communication is inside the same LAN.

2. IPv4

- Address format: 32 bits, usually written in dotted decimal notation

binary: 32 bits

example 1: **b**1000 0000 1011 1111 1001 0111 0000 0001

example 2: **b**1000 0001 1100 0000 1100 1000 0000 0010

dotted decimal: 4 integers (one integer = 8 bits/ binary digits)

example 1: 128.191.151.1

example 2: 129.192.200.2

hexadecimal: 8 hex digits (one hex digit = 4 bits/ binary digits)

example 1: **x**80 bf 97 01

example 2: **x**81 c0 c8 02

Review: Binary, Decimal and Hexadecimal

Given an integer B (the basis) any integer can be represented as a string in an alphabet of B symbols, starting from 0.

	Basis	*Alphabet*	*Example*
Binary	II	$\{0,1\}$	1100 1000
Decimal	X	$\{0,1,2,3,4,5,6,7,8,9\}$	200
Hexadecimal	XVI	$\{0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f\}$	$c8$

Binary <—> hex is easy: one hex digit (= nibble) is 4 binary digits

$c_{hex} = 1100_{bin}$ $8_{hex} = 1000_{bin}$ $c8_{hex} = 1100\ 1000_{bin}$

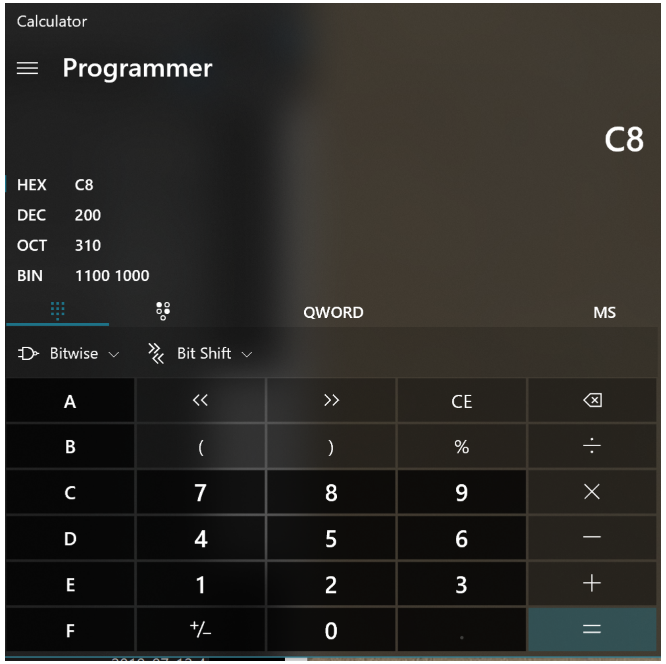
Binary/hex <—> decimal is best done by a calculator

$1100\ 1000_{bin}$ = 2^7 + 2^6 + 2^3 = 128 + 64 + 8 = 200

Special Cases to remember

$f_{hex} = 1111_{bin} = 15_{dec}$

$ff_{hex} = 1111\ 1111_{bin} = 255_{dec}$



The mask 255.255.254.0 means that the subnet is made of the first …

Go to web.speakup.info or download speakup app

Join room 87072

A. 16 bits
B. 18 bits
C. 22 bits
D. 23 bits
E. 24 bits

Solution

Answer D

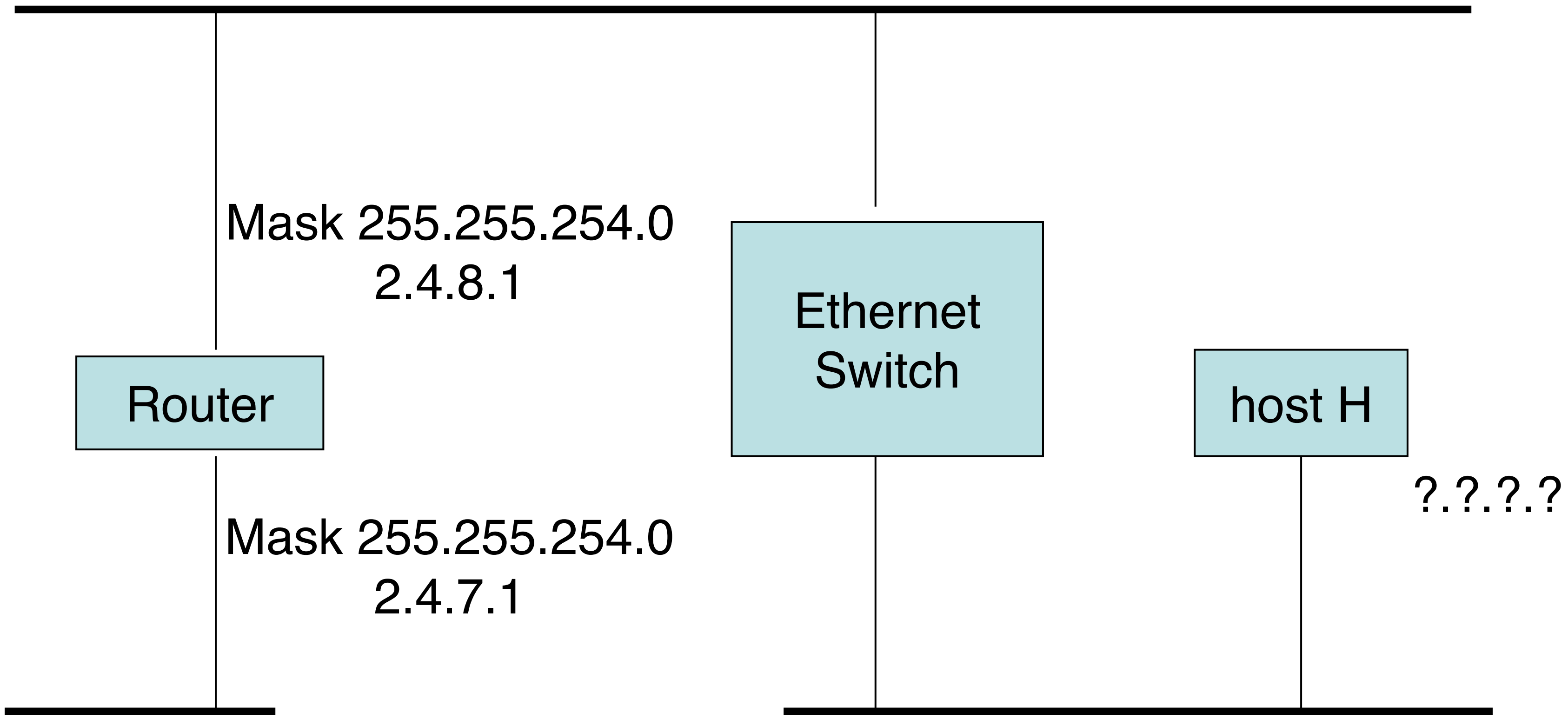
254 is 1111 1110 in binary form.

I.e., 255.255.254.0 = 1111 1111 1111 1111 1111 1110 0000 0000 in binary form.

So, the mask has 23 contiguous bits equal to 1 starting from the beginning.

Which address is a valid choice for H ?

A. only 2.4.8.2
B. only 2.4.9.1
C. Both A and B
D. None



Solution

Answer C

Router's north interface and H are in the same subnet

So H must have a subnet prefix of 23 bits.

We have:

- Router north's subnet prefix: 2.4.8 / 23 = 0000 0010 0000 0100 0000 100
- So A is correct because: 2.4.8.2 = 0000 0010 0000 0100 0000 1000 0000 0010

- B is also correct because: 2.4.9.1 = 0000 0010 0000 0100 0000 1001 0000 0001

and: 2.4.9 / 23 = 0000 0010 0000 0100 0000 100

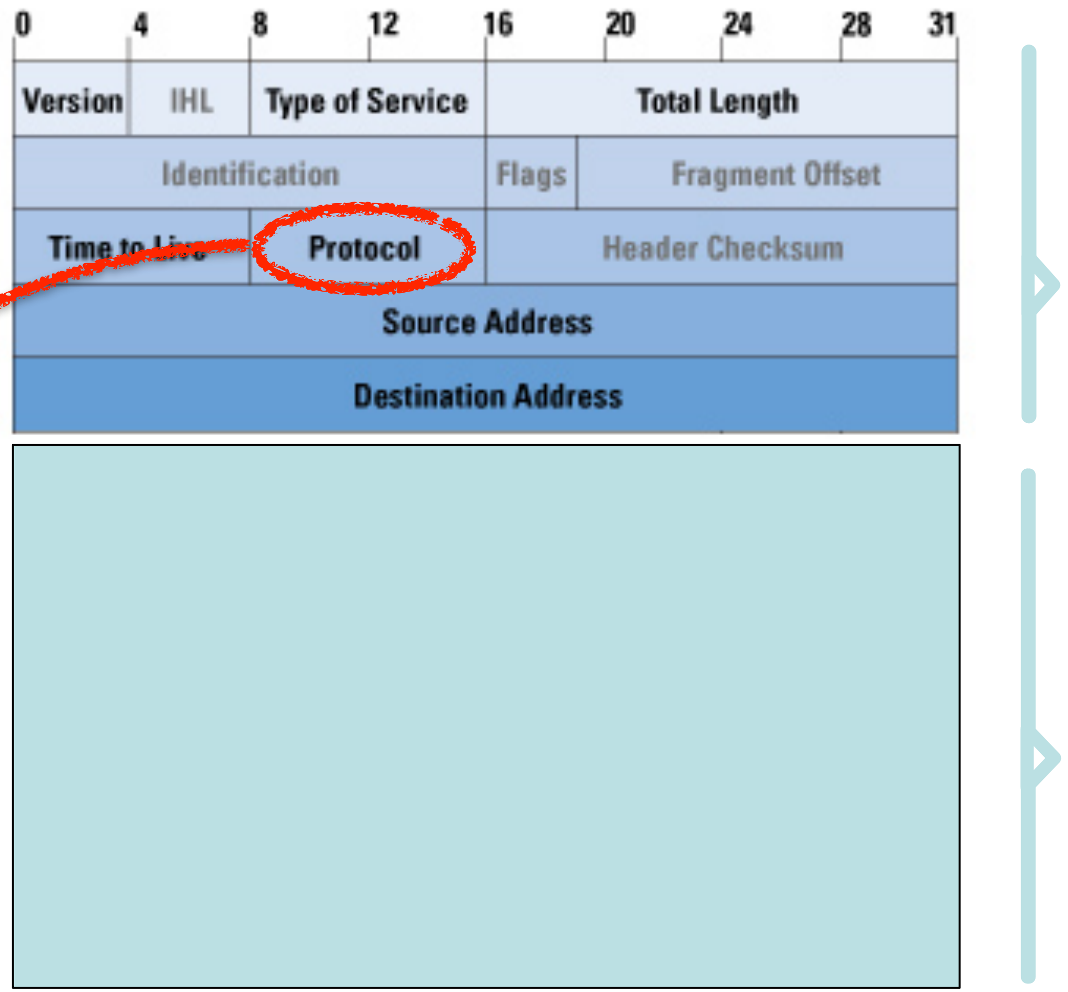
I.e.: the two prefixes are the *same*: 2.4.9/23 = 2.4.8/23 !

Reserved IPv4 address blocks

0.0.0.0	**absence** of address, or **any** address
127/8	**loopback** addresses (this host, e.g. 127.0.0.1)
10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16	**private** addresses (e.g. at home): used by *anyone*, but *not* in the public Internet (internet routers drop packets destined to such addresses)
100.64/10	private addresses used only by Internet Service Providers (ISPs)— **Carrier Grade NAT addresses**
192.88.99/24	**IPv6-to-IPv4** relay routers
169.254.0.0/16	**link local** addresses (used by systems in the same LAN, assigned automatically with *stateless autoconfiguration*, never routed)
224/4	**multicast**
240/4	reserved "for experimental/future use" until recently
255.255.255.255/32	link local (LAN) **broadcast**

✴ Every other address is *global/public* = *uniquely* identifies a network interface in the public internet

IPv4 Packet Format



Higher-layer protocol [1 = ICMP*, 6 = TCP, 17 = UDP]

Header 20 bytes (+ options, if any)

protocol overhead

payload

useful bits (higher-layer data)

* (ICMP is used to carry error messages at the network layer)

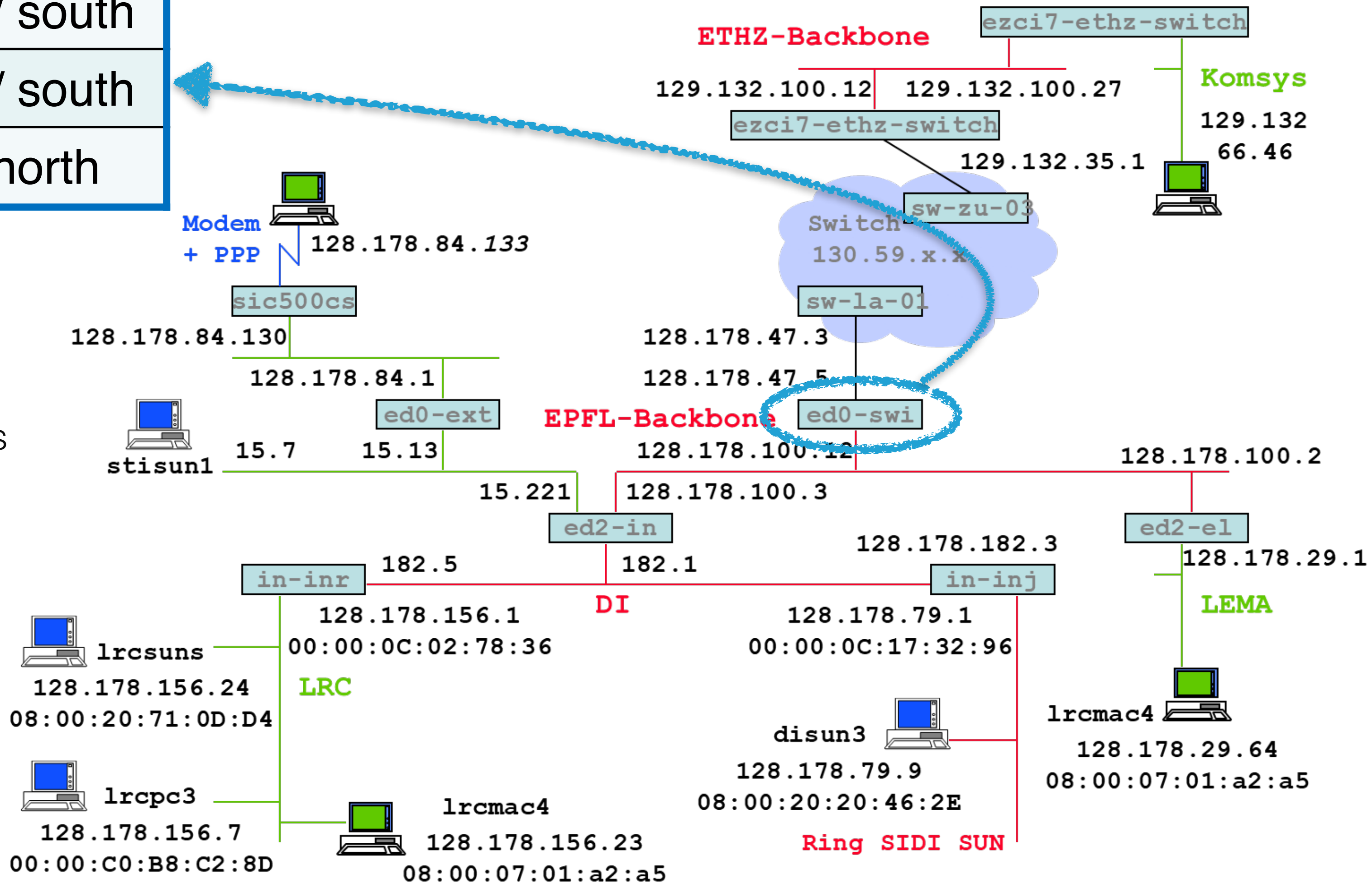
Forwarding table and longest prefix match: example from EPFL

Destination	Next-Hop / Interface
128.178.29/24	128.178.100.2 / south
128.178/16	128.178.100.3 / south
0/0	128.178.47.3 / north

- destination entries in *aggregated/subnet* form
- next hops identified by both “IP” and “interface”
- 0/0 (empty string) = default route for any address

- Longest prefix match means:

```
if packet -> 128.178.*
    if packet -> 128.178.29*
        forward to ed2-el
    else
        forward to ed2-in
else
    forward to sw-la-01
```



3. IPv6

Why a new version ?

IPv4 address space is too small (32 bits $\rightarrow \approx 4 \cdot 10^9$ unique addresses)

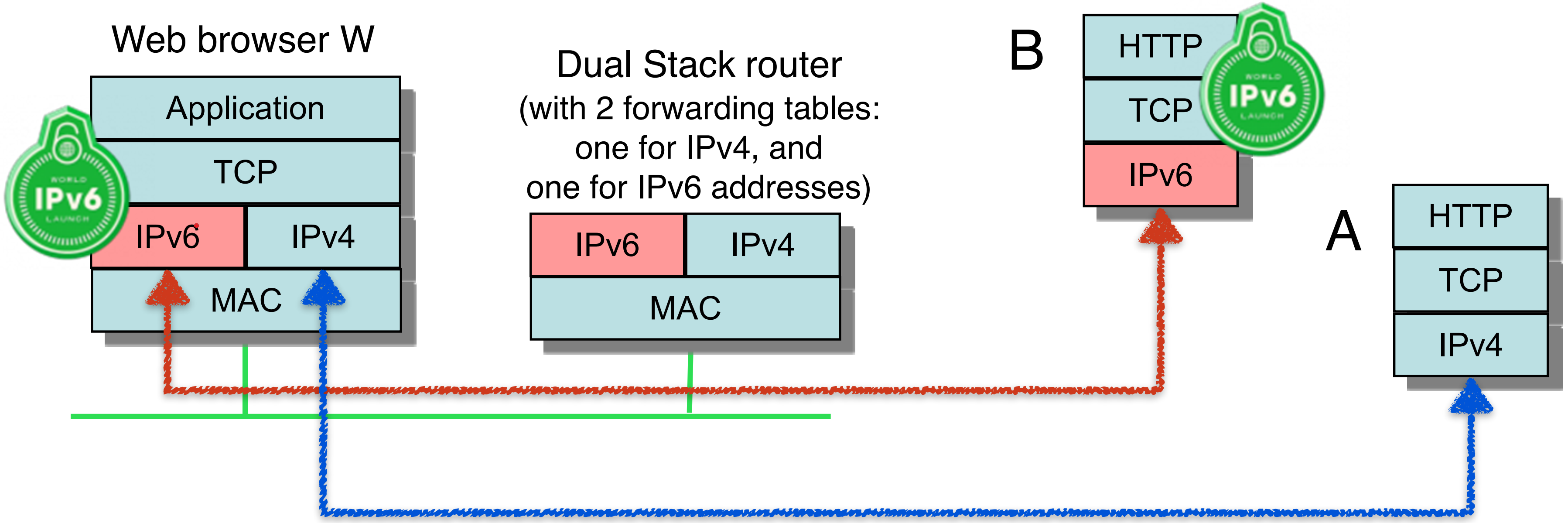
What does IPv6 do ?

Redefines packet format with larger addresses of: 128 bits ($\approx 3 \cdot 10^{38}$ unique addresses)

Otherwise, it offers essentially the same services as IPv4

But IPv6 is *incompatible* with IPv4; routers and hosts must handle them separately

A can talk to W, B can talk to W, A and B cannot communicate at the network layer



IPv6 Address format

- Hex notation with lowercase letters
  - 8 hextets separated by ":"
  - *hextet* = [0-4] hex digits = 16 bits
- Compression Rules
  - leading 0s of a hextet can be *omitted*
  - :: replaces any number of hextets of all 0s; but it is used *at most once* to avoid ambiguity

uncompressed form	compressed form
2002:0000:0000:0000:0000:ffff:80b2:0c26	2002::ffff:80b2:c26
2001:0000:0000:01a6:0000:20ff:fe78:30f9	2001::1a6:0:20ff:fe78:30f9

Reserved address blocks

::/128	**absence** of address or **any** address
::1/128	**loopback** address (this host)
fc00::/7 (i.e. fcxx: and fdxx:) for example: fd24:ec43:12ca:1a6:a00:20ff:fe78:30f9	unique local addresses = **private** networks (e.g. in EPFL): *not* to be used in the public Internet
fe80::/10	**link local** addresses (used only by systems in same LAN)
ff00::/8	**multicast**
ff02::1:ff00:0/104	**solicited node multicast** (see NDP later)
ff02::1/128	**link local broadcast**
ff02::2/128	**multicast to all link-local routers** (in same LAN)

EPFL Private

IPv6 private prefixes are also allocated, not reused

an EPFL public address:

2001:620:618:1a6:a00:20ff:fe78:30f9

an EPFL private address:

fd24:ec43:12ca:1a6:a00:20ff:fe78:30f9

This is a private address

EPFL's private prefix,
not to be re-used by anyone else
(*truly private*)

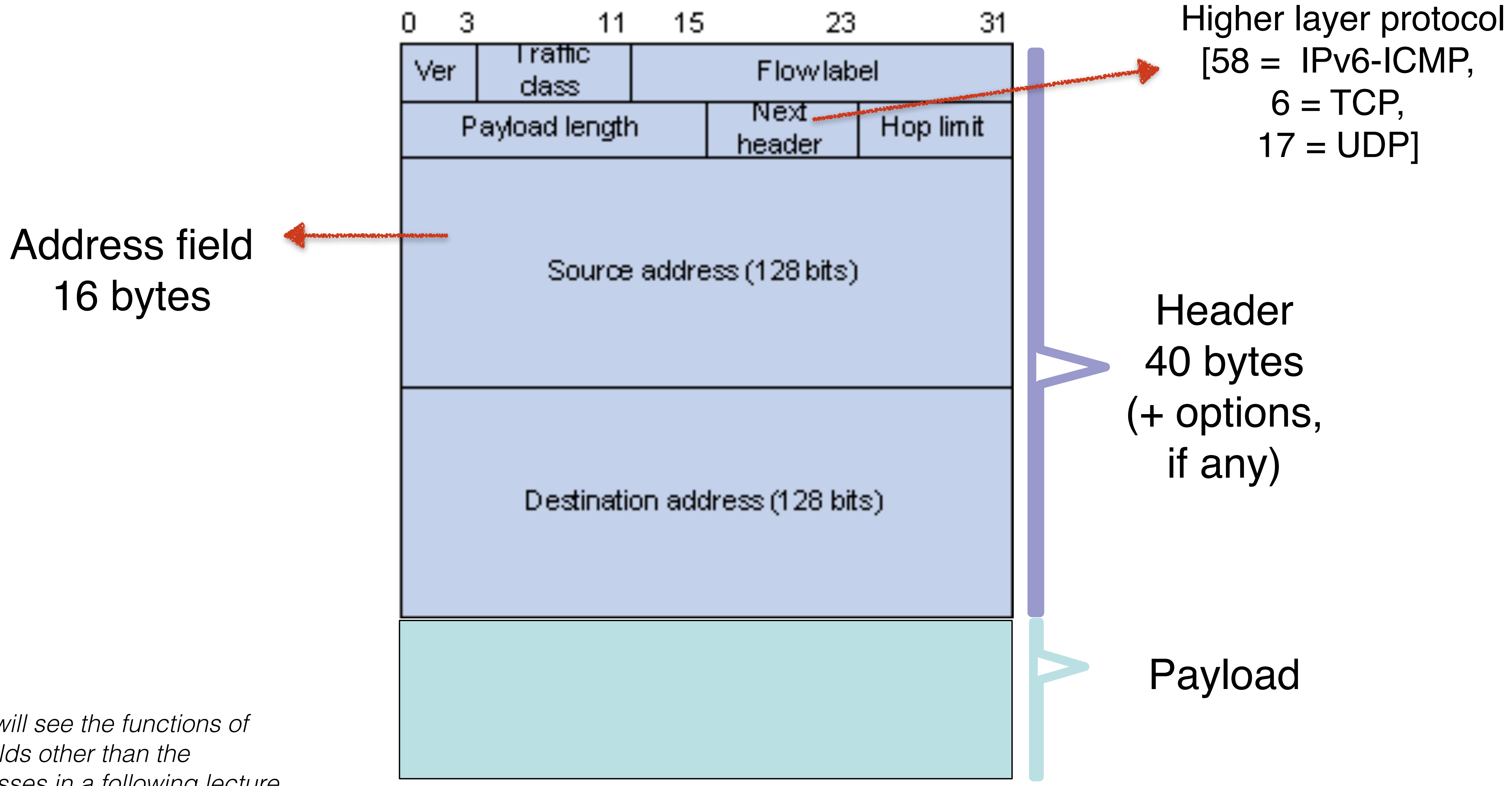
A few IPv6 global unicast addresses

- Currently, only the block 2000/3 (i.e. 2xxx and 3xxx) is for global/public unicast addresses

2001:620::/32	Switch
2001:620:618::/48	EPFL
2001:620:8::/48	ETHZ
2a02:1200::/27	Swisscom
2001:678::/29	provider independent address
2001::/32	Teredo (tunnels IPv6 in IPv4)
2002::/16	6to4 (tunnels IPv6 in IPv4)

- *Hierarchical* allocation:
  Networks served by a provider use blocks that are ***subsets*** of the provider's address block, (e.g. check EPFL, ETHZ and SWITCH)

IPv6 Packet Format



***We will see the functions of the fields other than the addresses in a following lecture*

IPv6 Forwarding table: same example from EPFL at ed0-swi

IPv6 forwarding table

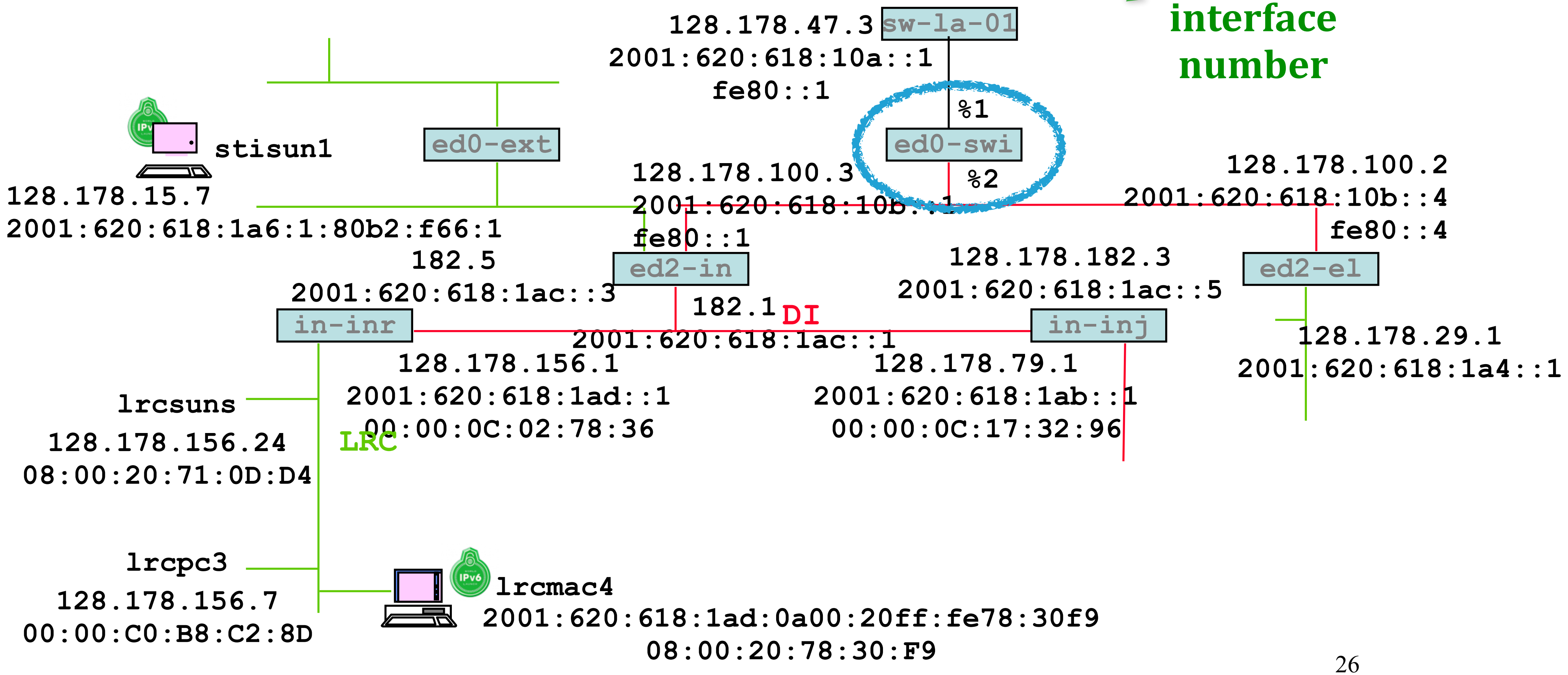
Destination	Next-Hop-IP%Interface
2001:620:618:1a4/64	fe80::4%2
2001:620:618/48	fe80::1%2
::/0	fe80::1%1

IPv4 forwarding table

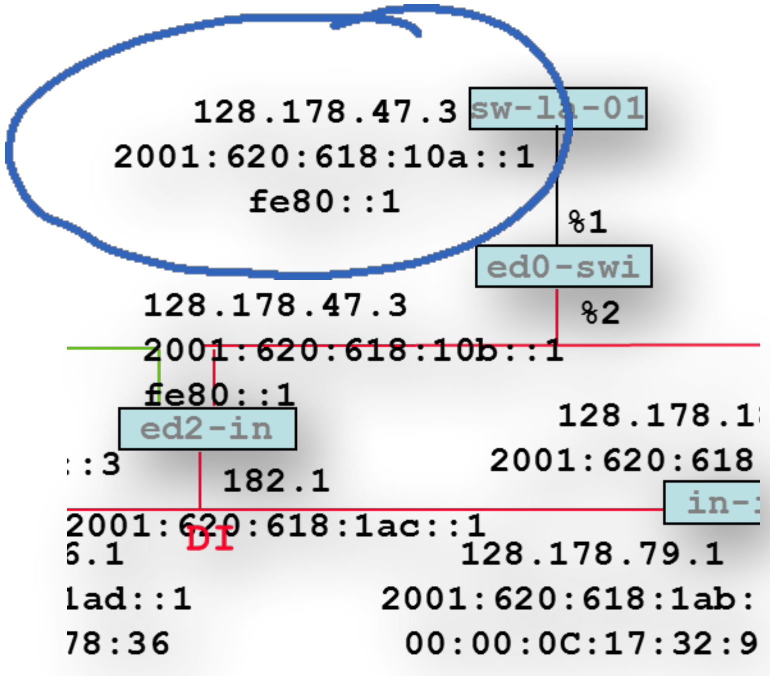
Destination	Next-Hop / Interface
128.178.29/24	128.178.100.2 / south
128.178/16	128.178.100.3 / south
0/0	128.178.47.3 / north

IP address of next hop

interface number



Multiple Addresses per Interface are the Rule with IPv6



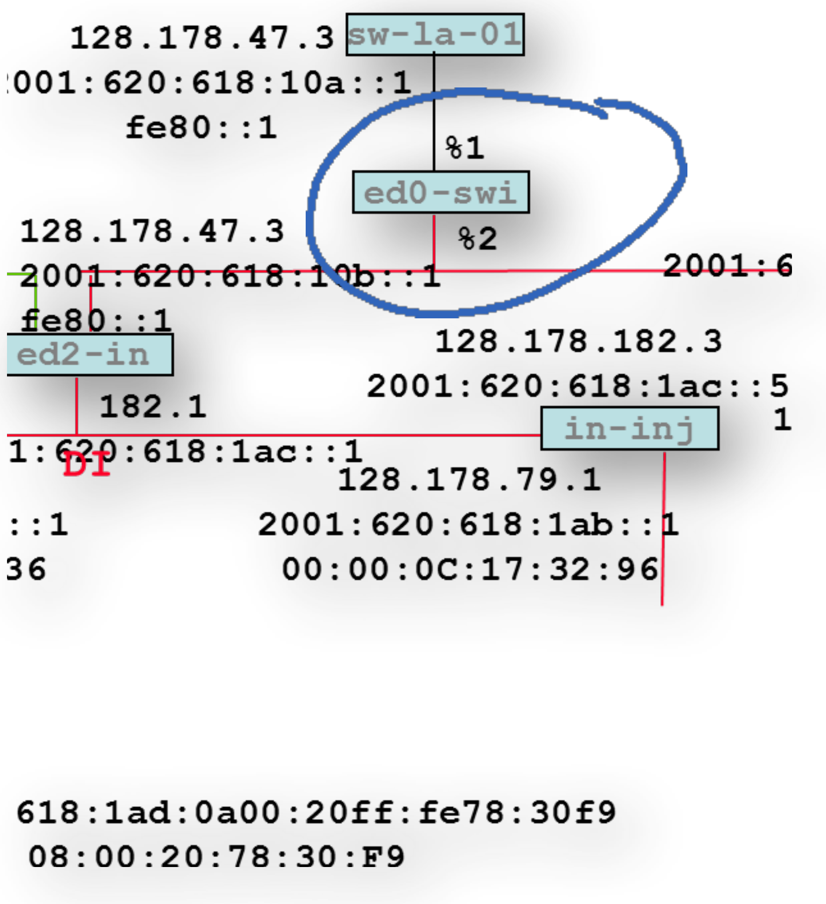
A host interface typically has

- one or several *link local* addresses
- plus one or several global *unicast* addresses (secure (CGA) address, temporary addresses)

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

In contrast, in IPv4:
there is usually only one IP address per interface

Zone Index



Identifies an interface inside one machine that has several interfaces (e.g. a router)

- 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

The dotted decimal notation for $0102:ffff$ is …

A. 1.2.255.255
B. 16.32.255.255
C. 228.393.255.255

Go to web.speakup.info or download speakup app

Join room 87072

Solution

Answer A

Recall the mapping (hex) ff → (decimal) 255

In full, the hexadecimal notation «2001::ada:bada» means…

A. 2001:0ada:bada
B. 2001:0000:0000:0000:0000:0000:0ada:bada
C. 2001:0000:0ada:bada
D. 2001:0000:ada:bada
E. None of the above

Go to web.speakup.info or download speakup app

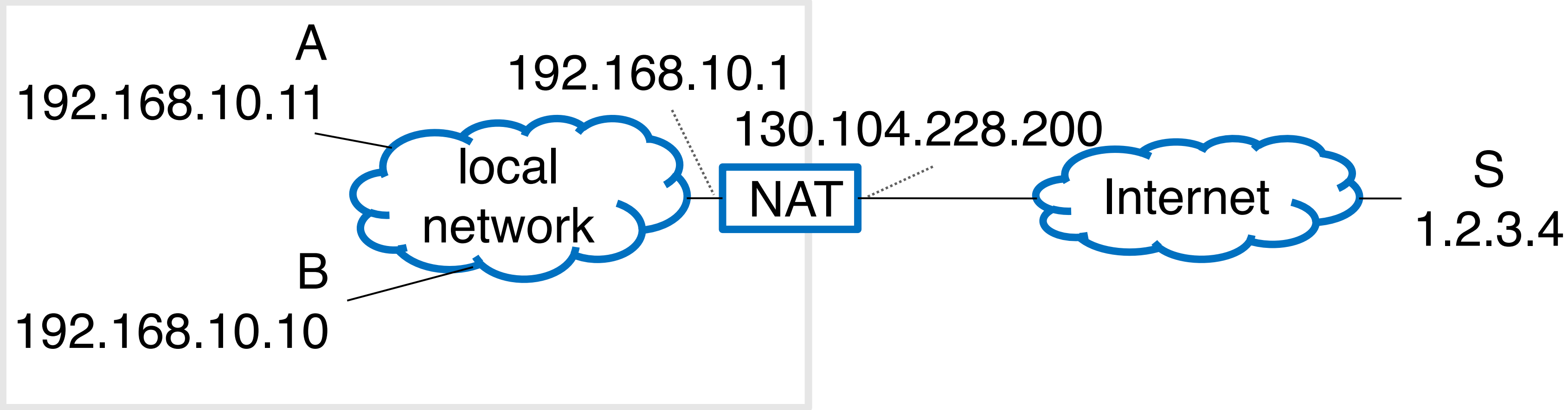
Join room 87072

Solution

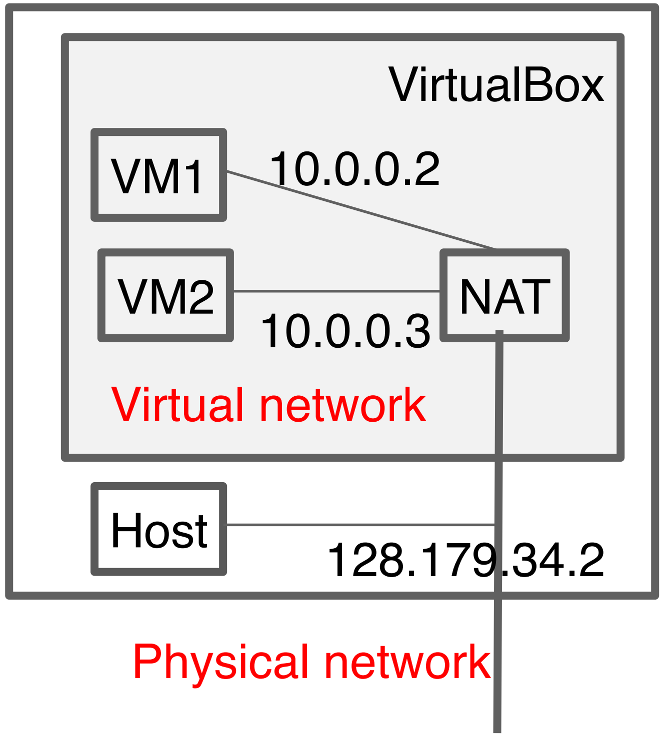
Answer B. The convention :: means as many 0's as required to make the string 128 bits.

Leading zeros are omitted, so that :bad: means :0bad:

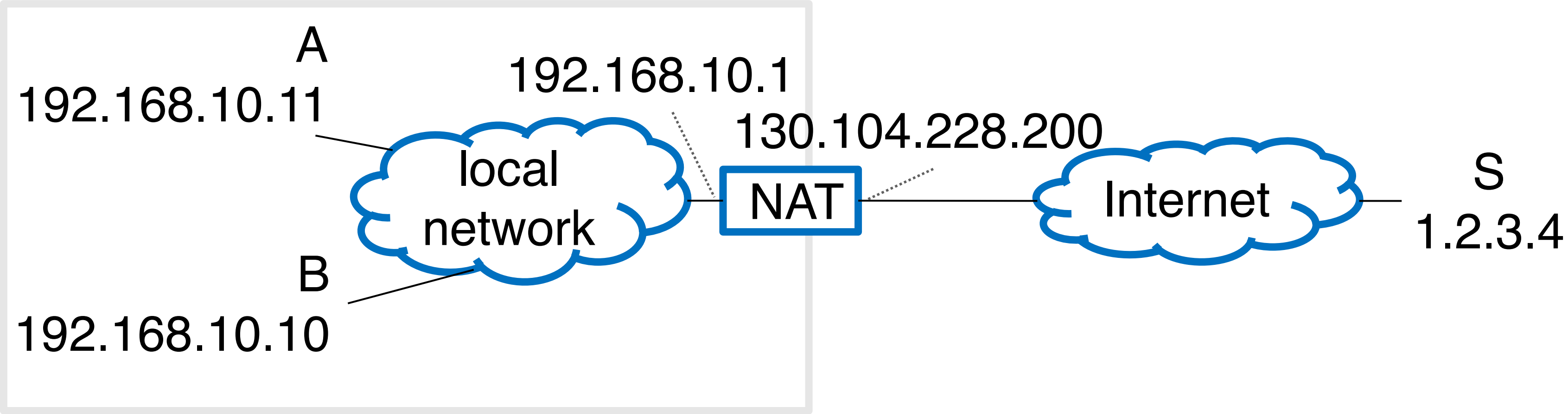
4. NAT (Network Address Translation) box

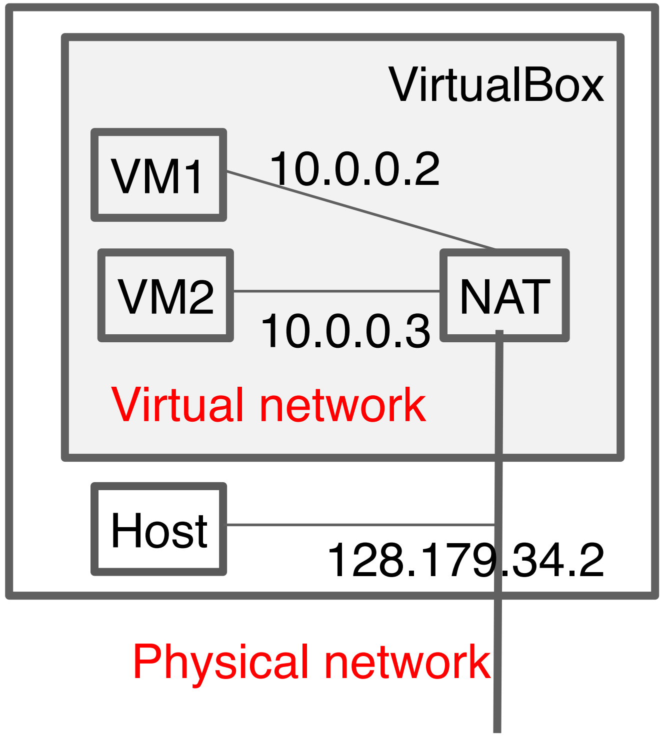


- *Why* invented? To allow $n > 1$ devices to ***share*** a ***single*** public IP address; e.g.:
  - Internet service provider gives you a single IPv4 address, but you have *n* devices at home and need more addresses.
  - A virtualization platform offers *n* guest VMs in one host and allows them to communicate with outside, using the single IP address of the physical machine.
- *What* does it do?
  - NAT translates (= *masquerades*) an *internal IP* address and *internal port* number into *NAT IP* address and *NAT port*
  - Internal addresses are typically ***private***, ports are either UDP or TCP
  - From outside, one sees only the (***public***) NAT IP address and a NAT port



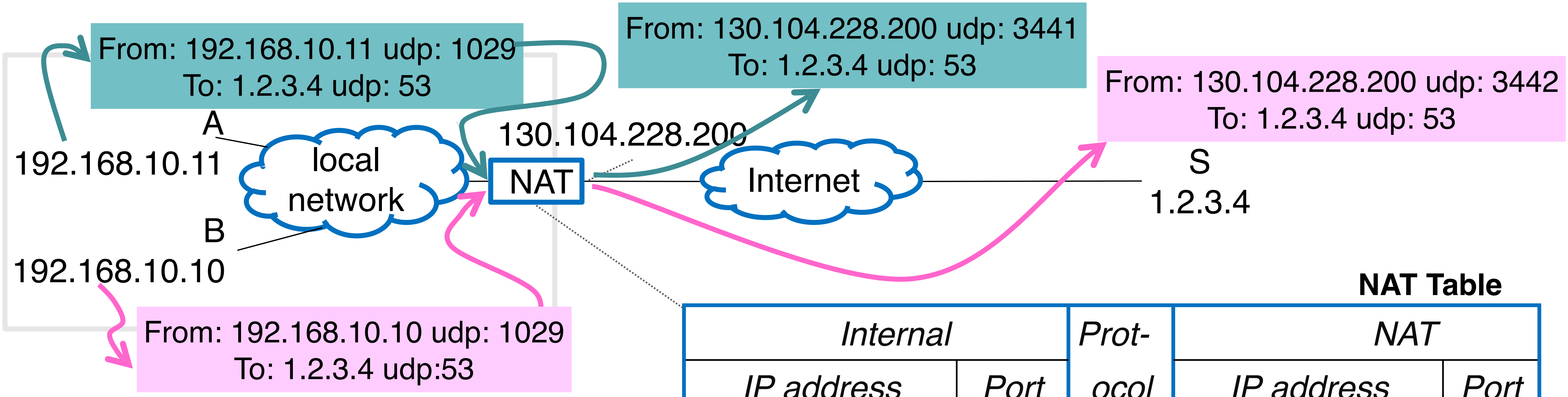
4. NAT (Network Address Translation) box





- NAT is a ***network-layer*** middle box, but ***violates***:
  - layering —> manipulates 2 layers to work
  - IP rule that public addresses should identify hosts *uniquely*

How does NAT treat *outbound* traffic?

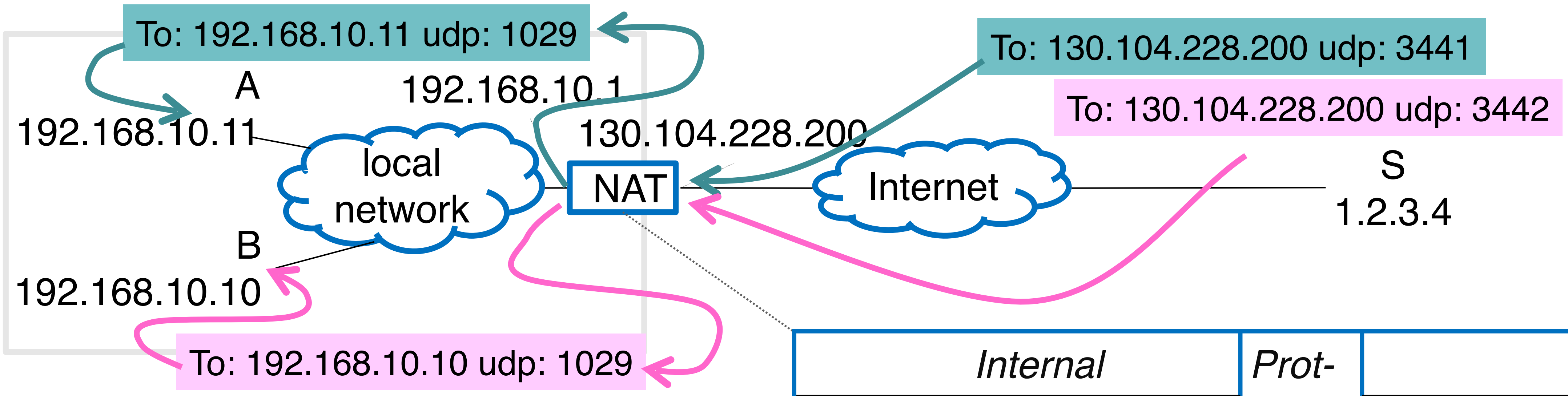


NAT Table

Internal		*Prot-*	*NAT*	
IP address	*Port*	*ocol*	*IP address*	*Port*
192.168.10.11	1029	udp	130.104.228.200	3441
192.168.10.10	1029	udp	130.104.228.200	3442

- When a packet goes from internal network (a.k.a. LAN) to the external network (a.k.a. WAN), NAT:
  - translates *source IP address + port* by *changing* the IP and UDP/TCP headers
  - stores this mapping in the *NAT table*
  - finally forwards the packet

How does NAT treat *inbound* traffic?



NAT Table

Internal		*Prot-ocol*	*NAT*	
IP address	*Port*		*IP address*	*Port*
192.168.10.11	1029	udp	130.104.228.200	3441
192.168.10.10	1029	udp	130.104.228.200	3442

- When packets come from external network to internal network, NAT translates *destination IP address + port*
- IP forwarding is based on *exact matching* in the NAT table
  - ➡ If no matching entry exists, packet is dropped