X_405082
Advanced Computer Networks

Forwarding and Routing

Lin Wang (lin.wang@vu.nl)

Period 2, Fall 2020

VRIJE UNIVERSITEIT AMSTERDAM

Part of the content is adapted from P. Brighten Godfrey

Course outline

Warm-up

- Fundamentals
- **Forwarding and routing** 👈
- Network transport

Data centers

- Data center networking
- Data center transport

Programmability

- Software defined networking
- Programmable forwarding

Video

- Video streaming
- Video stream analytics

Networking and ML

- Networking for ML
- ML for networking

Mobile computing

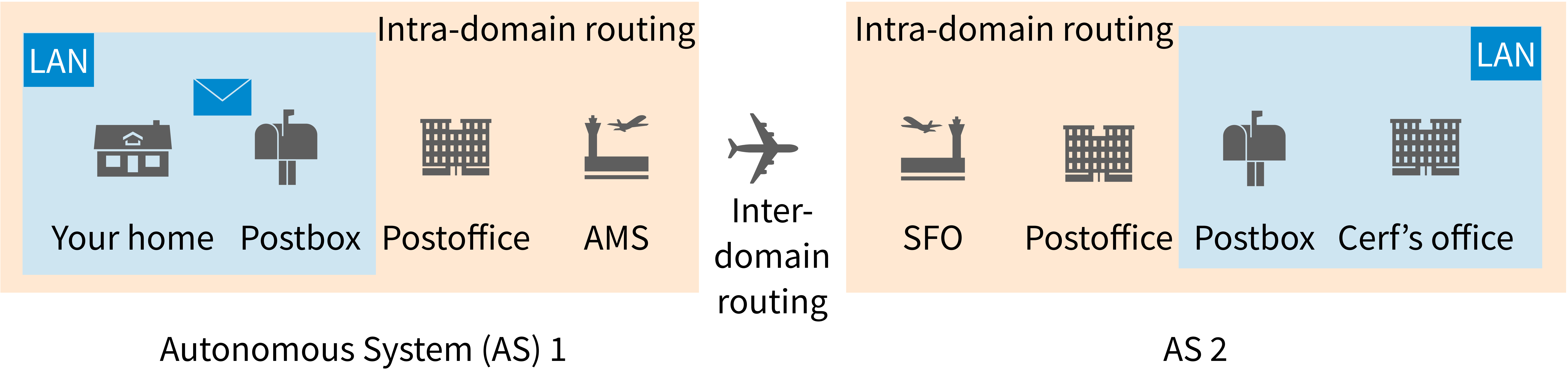
- Wireless and mobile

Learning objectives

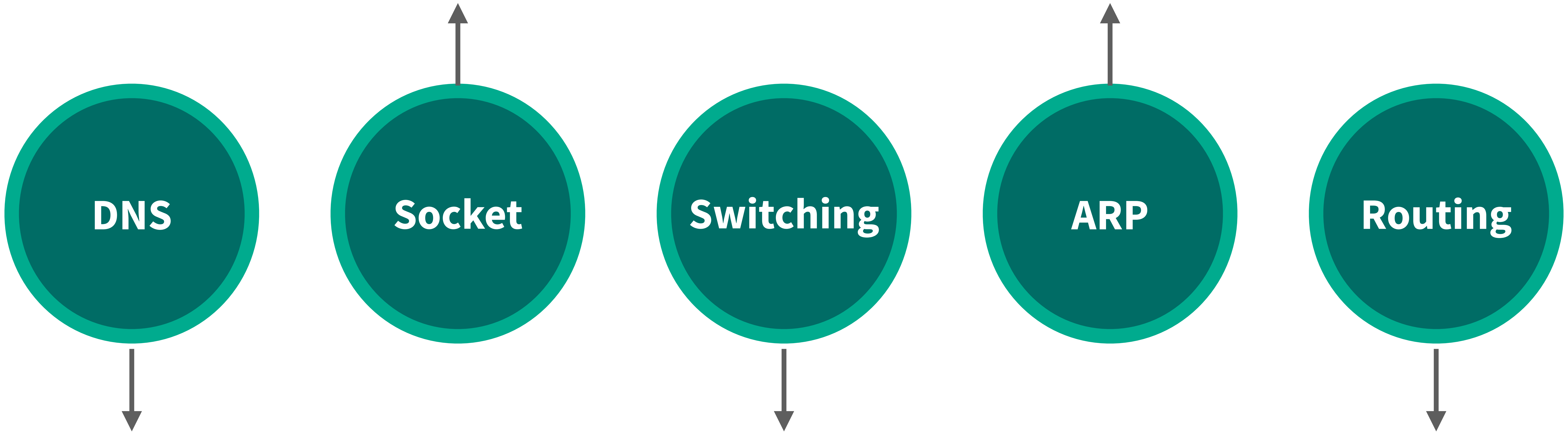
Answer the question **what happens when you visit the link** “https://www.google.com”?

An analogue

Suppose you want to send a letter to Internet pioneer Vint Cerf @Google



Key networking concepts



DNS

Translates the human-friendly URL to machine-friendly IP address

Socket

The endpoints (OS interfaces) for sending/receiving data across the network

Switching

Forwards packets within a local area network (i.e., Ethernet)

ARP

Resolves IP address to the Ethernet MAC address

Routing

Routes a network packet across multiples (sub)networks

Domain Name System (DNS)

If you want to mail someone

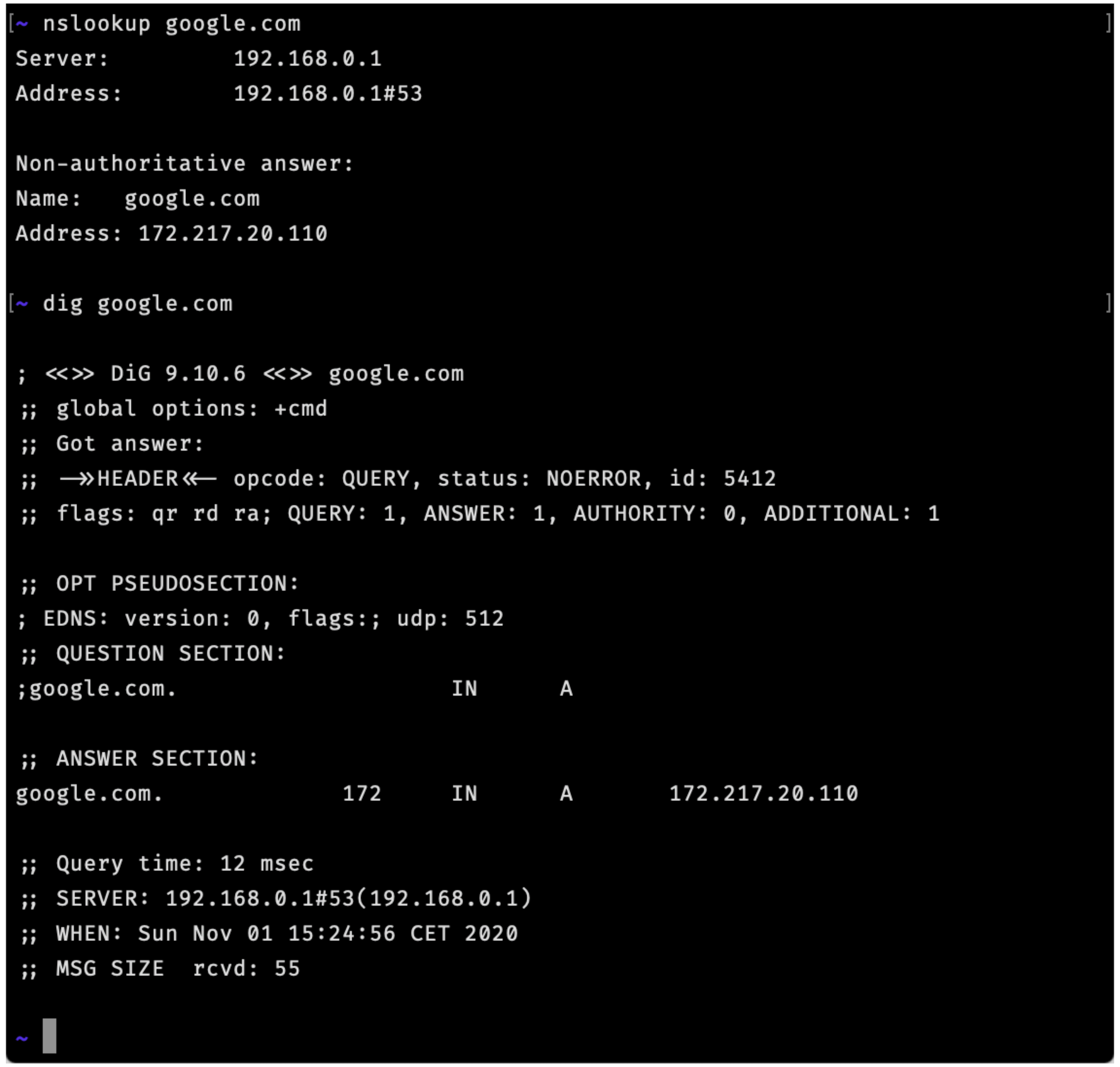
- You need to get their address first (recall the analogue)

What about the Internet?

- If you need to reach Google, you need their IP
- Does anyone know Google's IP?

Problem

- People cannot remember IP addresses
- Need human readable names that map to IPs

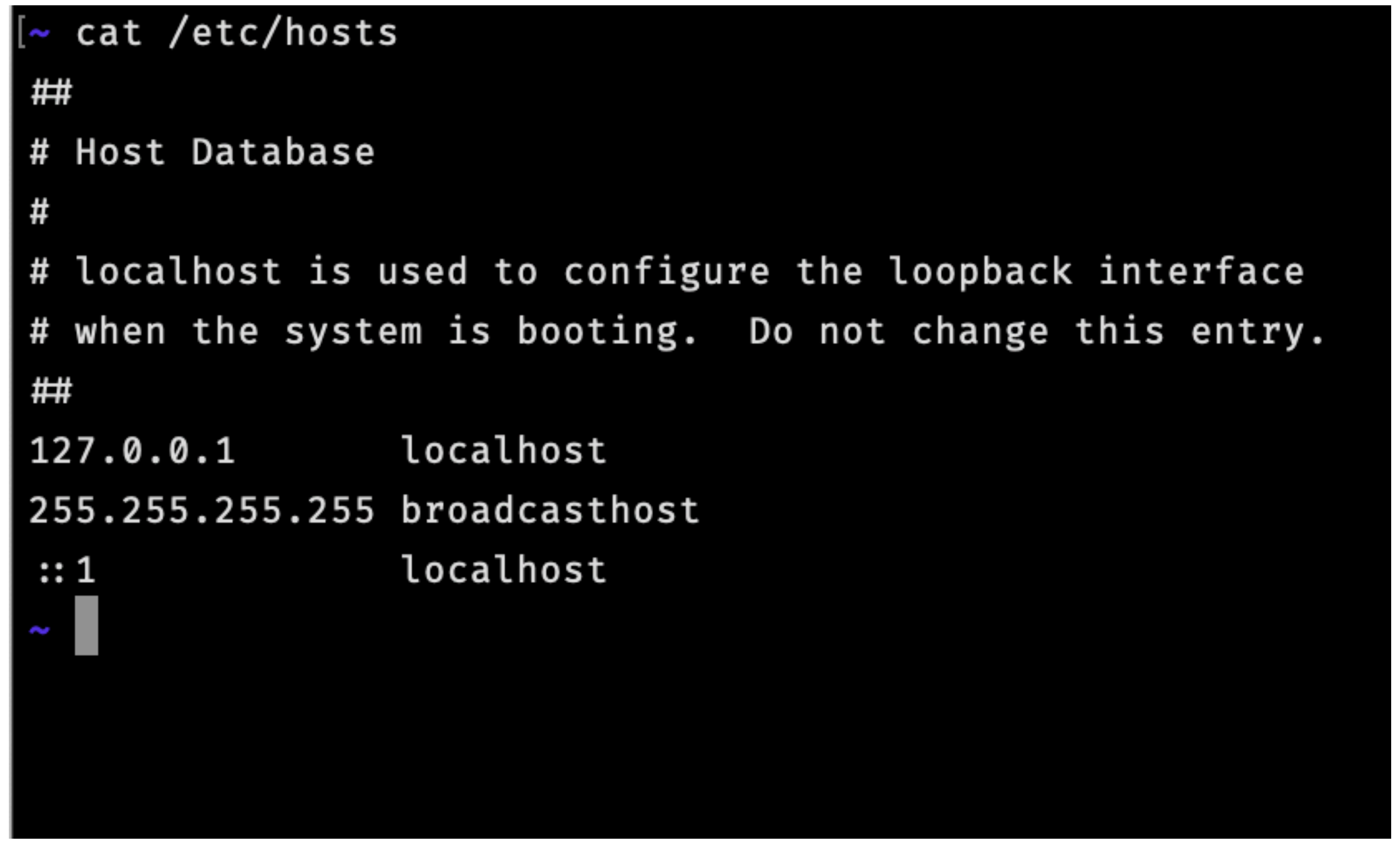


```
[~ nslookup google.com
Server:         192.168.0.1
Address:        192.168.0.1#53

Non-authoritative answer:
Name:   google.com
Address: 172.217.20.110

[~ dig google.com

; <<>> DiG 9.10.6 <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 5412
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;google.com.                    IN      A

;; ANSWER SECTION:
google.com.             172     IN      A       172.217.20.110

;; Query time: 12 msec
;; SERVER: 192.168.0.1#53(192.168.0.1)
;; WHEN: Sun Nov 01 15:24:56 CET 2020
;; MSG SIZE  rcvd: 55

~
```

DNS history

Before 1983 (the advent of DNS), all mappings were in a single file

- `/etc/hosts on Linux`
- `C:\\Windows\System32\drivers\etc\hosts` on Windows



```
[~ cat /etc/hosts
##
# Host Database
#
# localhost is used to configure the loopback interface
# when the system is booting.  Do not change this entry.
##
127.0.0.1       localhost
255.255.255.255 broadcasthost
::1             localhost
~
```

Centralized, manual system

- Changes were submitted to SRI (Stanford Research Institute) via email
- End hosts periodically FTP new copies of the `hosts` file
- Administrators could pick names at their discretion
- Any name was allowed: `alices_server_at_vrije_universiteit_amsterdam`

Not scalable

Hard to enforce uniqueness

Consistency issue

DNS overview

Distributed database

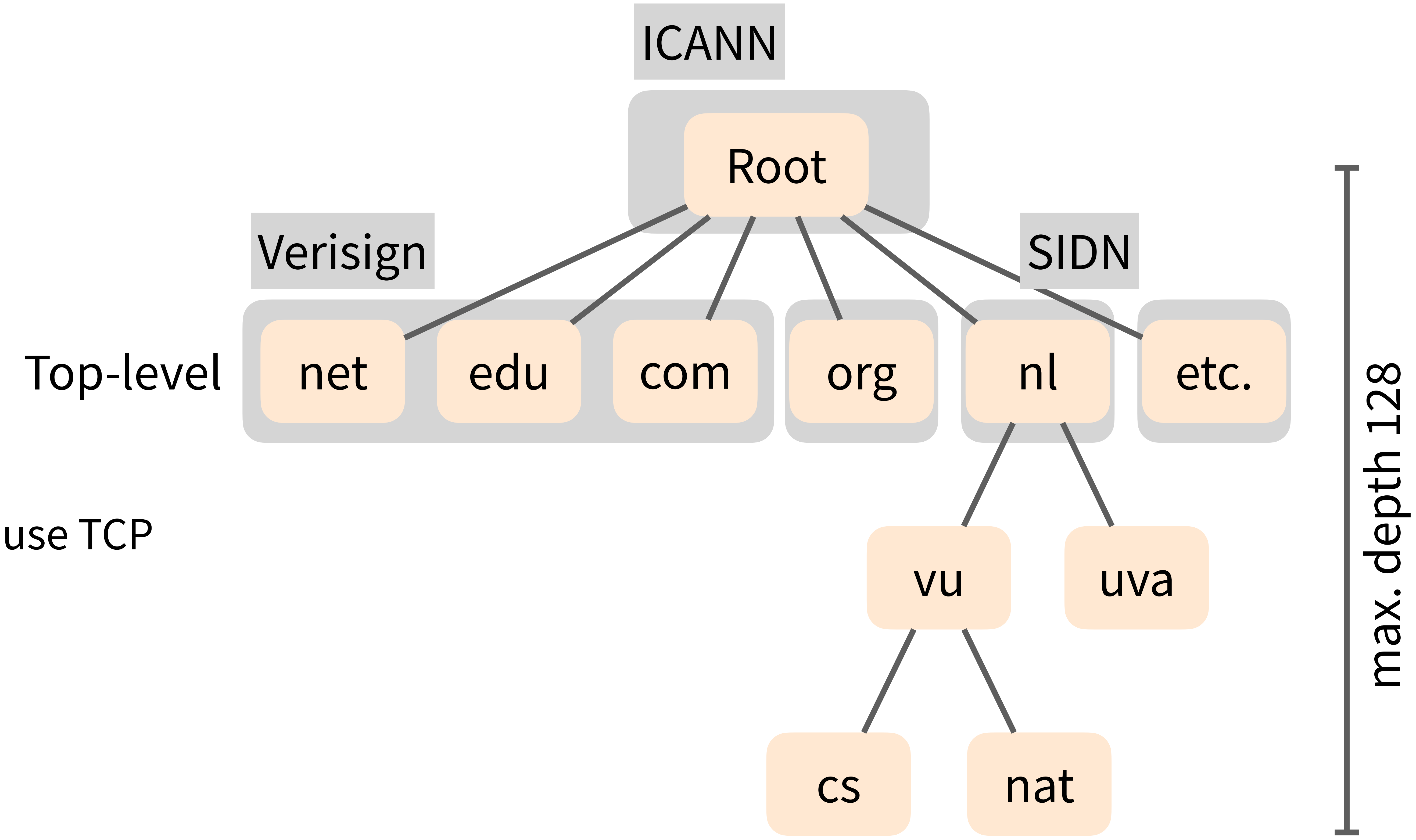
- No centralization → scalability

Simple client/server architecture

- UDP port 53, some implementations also use TCP

Hierarchical namespace

- As opposed to original, flat namespace
- E.g., `.com` → `google.com` → `mail.google.com`



Tree is divided into zones and each zone has an administrator, with a DNS server (maybe replicated)

Root name server

Responsible for the root zone file

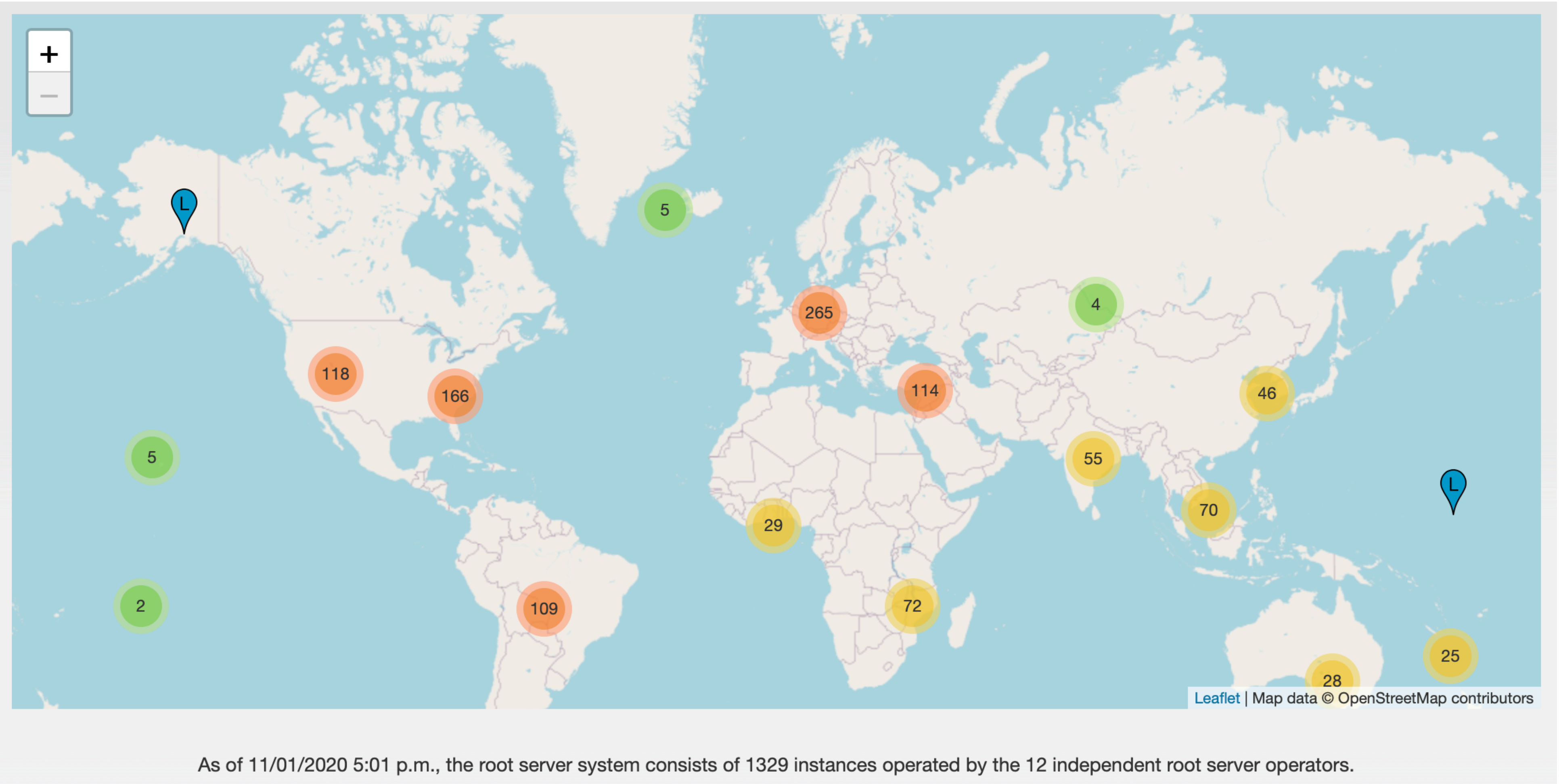
- Lists the top-level domains (TLDs) and who controls them
- ~ 2MB file size

Administrated by International Corporation for Assigned Names and Numbers (ICANN)

- 13 root servers, labeled A → M
- All are anycasted, i.e., they are globally replicated

Contacted when names cannot be resolved locally

- In practice, most systems cache this information

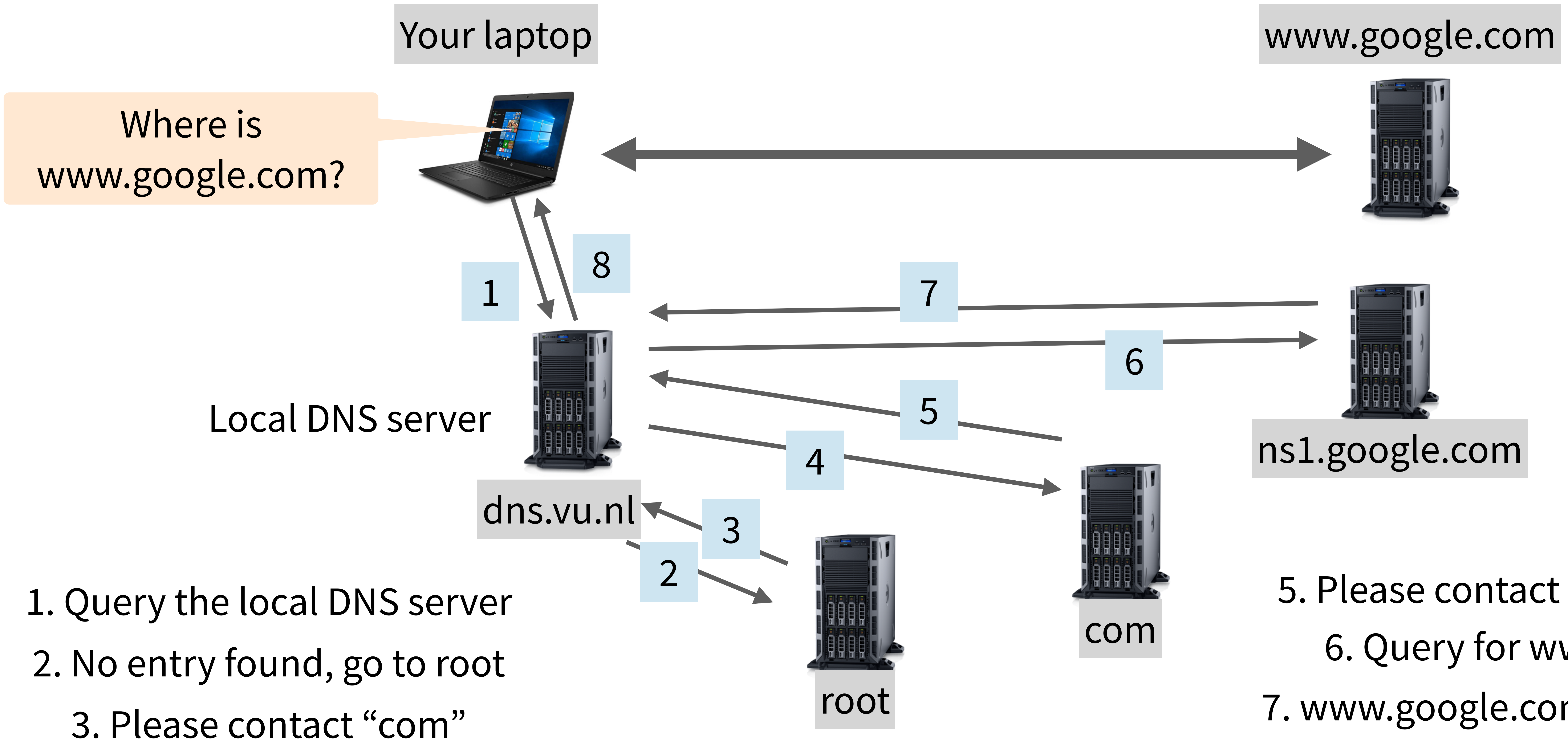


As of 11/01/2020 5:01 p.m., the root server system consists of 1329 instances operated by the 12 independent root server operators.

https://root-servers.org

Recursive DNS query

Each layer may apply caching (1-72 hours) to improve efficiency



1. Query the local DNS server
2. No entry found, go to root
3. Please contact “com”
4. Query for www.google.com
5. Please contact “ns1.google.com”
6. Query for www.google.com
7. www.google.com → 172.217.20.110
8. 172.217.20.110 is the answer

DNS types

Query
Name: www.cs.vu.nl
Type: A (or AAAA)

Resp.
Name: www.cs.vu.nl
Value: 130.37.164.171

DNS resolution (AAAA for IPv6)

Query
Name: foo.cs.vu.nl
Type: CNAME

Resp.
Name: foo.cs.vu.nl
Value: bar.cs.vu.nl

Look for alias (canonical hostname)

Query
Name: cs.vu.nl
Type: NS

Resp.
Name: cs.vu.nl
Value: 130.37.164.1

Query for DNS server
responsible for the partial name

Query
Name: cs.vu.nl
Type: MX

Resp.
Name: cs.vu.nl
Value: mail.cs.vu.nl

Look for the mail server

The importance of DNS

Without DNS…

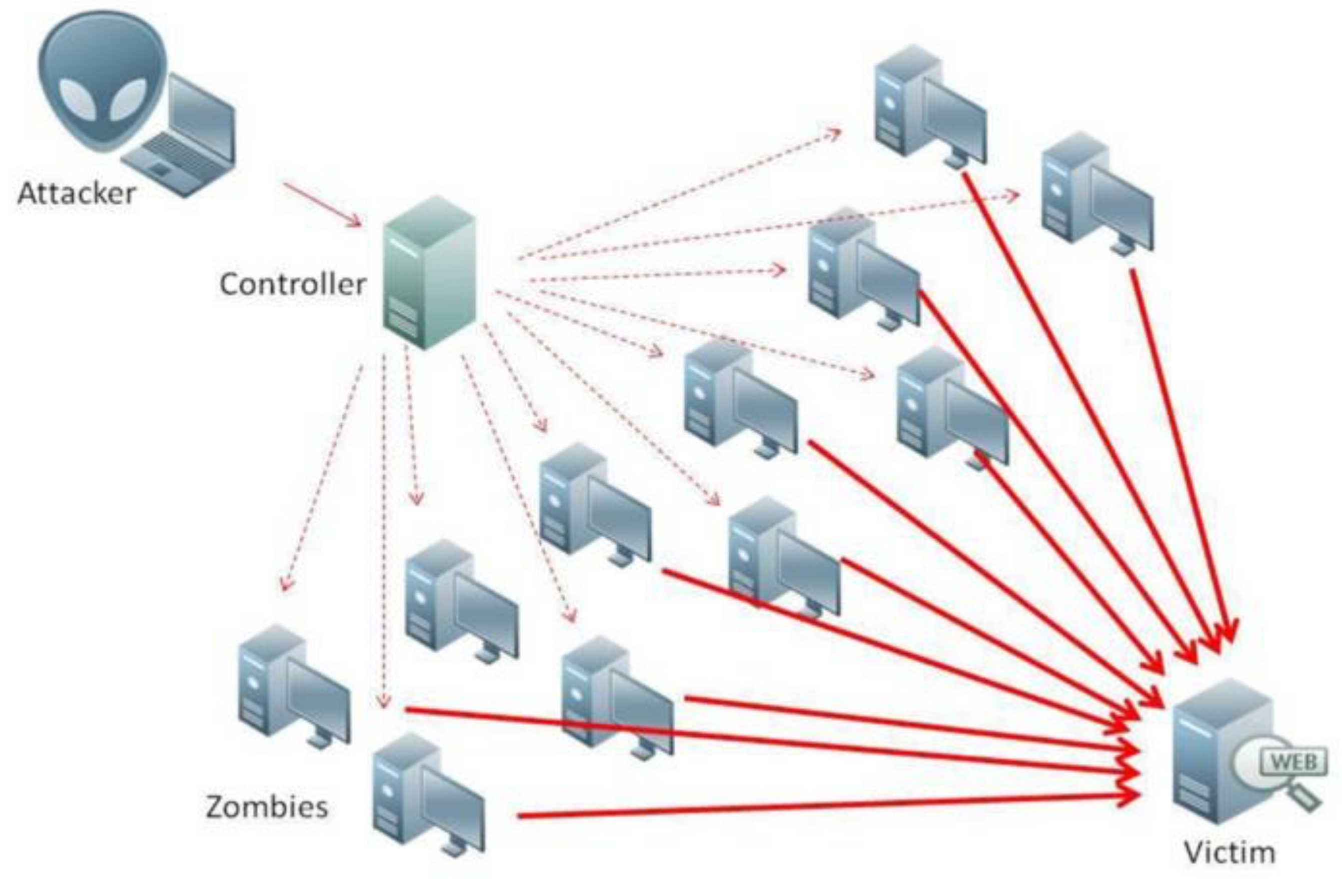
- How could you get to any websites?

You are your mail server

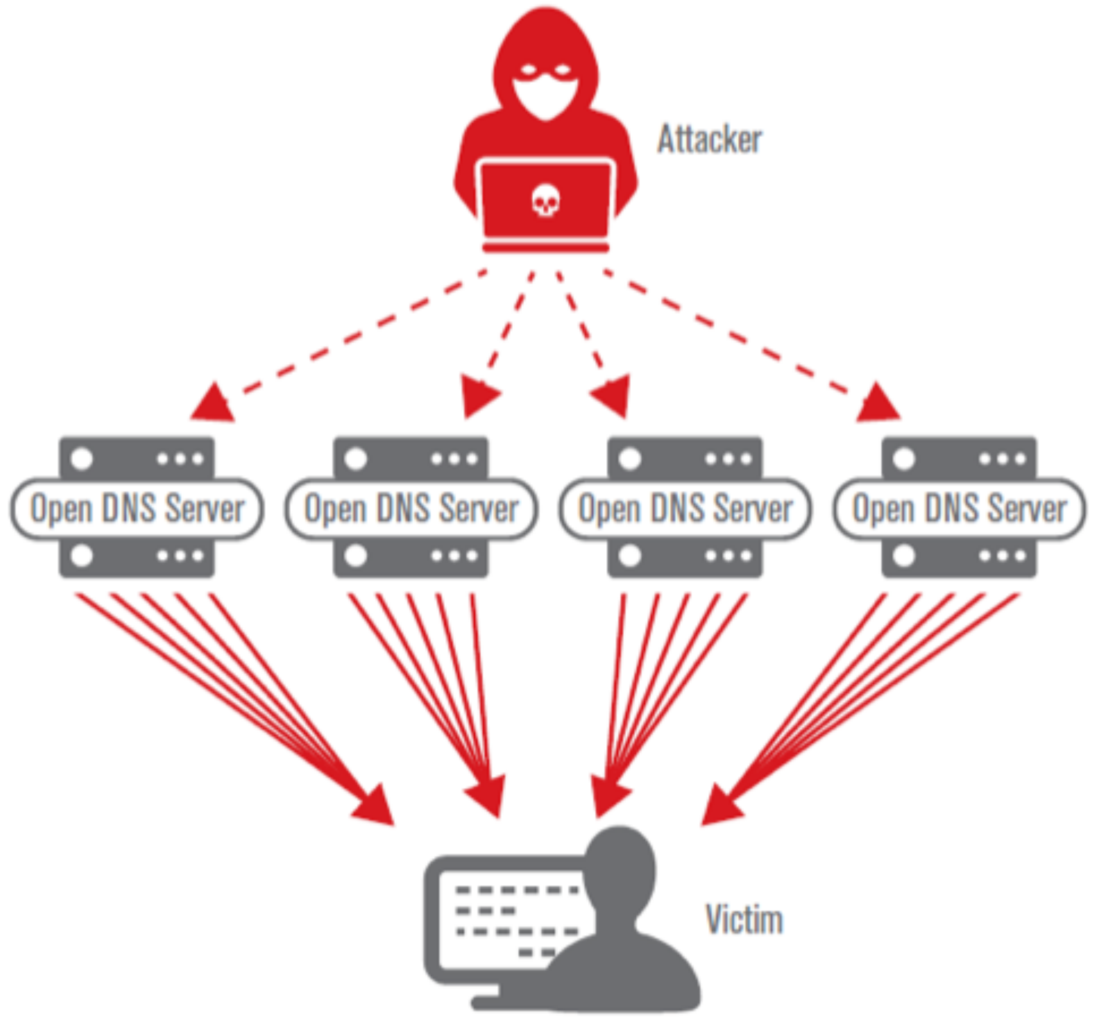
- When you sign up for websites, you use your email address
- What if someone hijacks the DNS for your mail server?

DNS is the root of trust for the web

- When a user types www.ing.nl, they expect to be taken to their bank's website
- What if the DNS record is compromised?

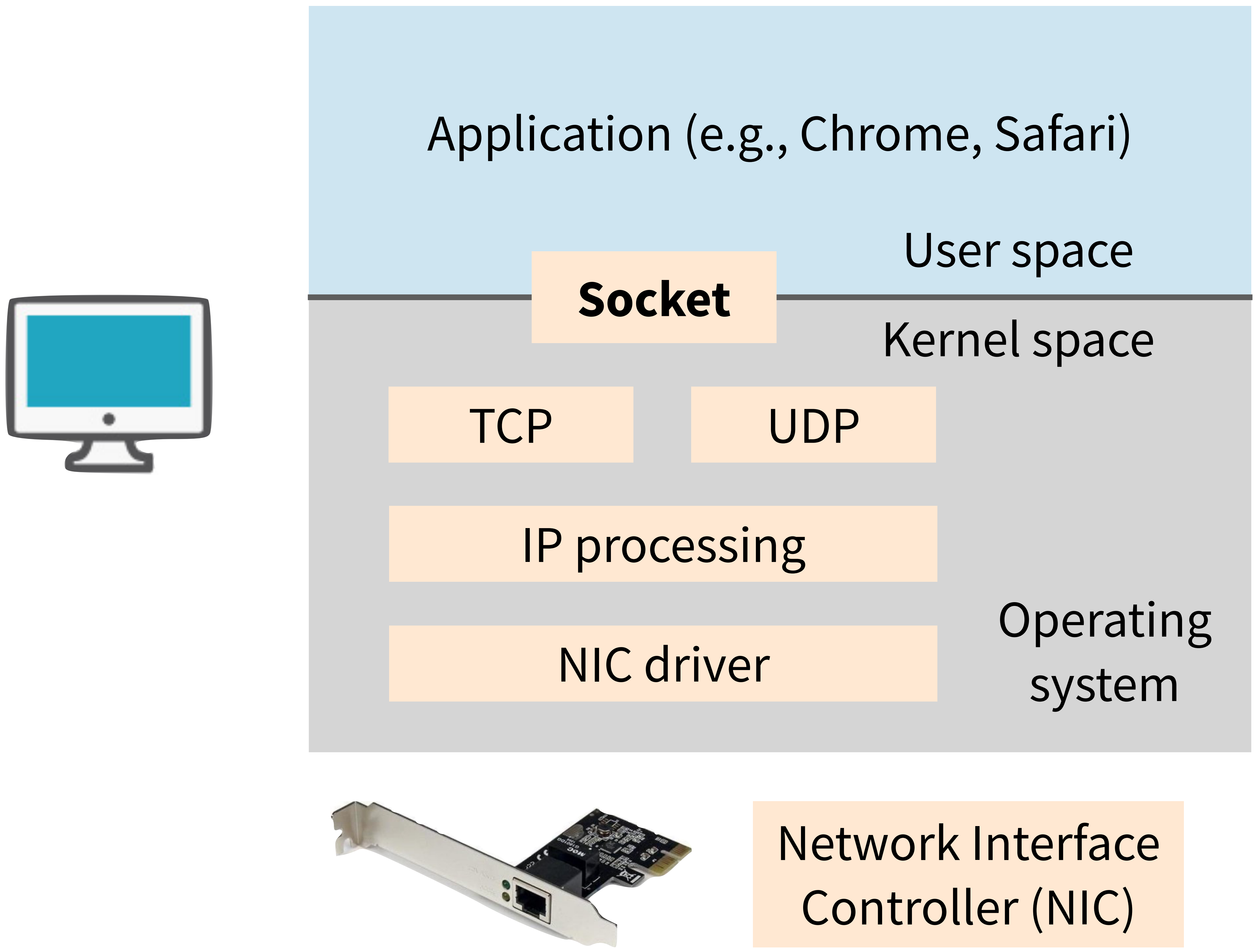


Distributed Denial of Service (DDoS)



DNS amplification attack

Socket



Application (e.g., Chrome, Safari)

User space

Socket

Kernel space

TCP

UDP

IP processing

NIC driver

Operating system

Network Interface Controller (NIC)

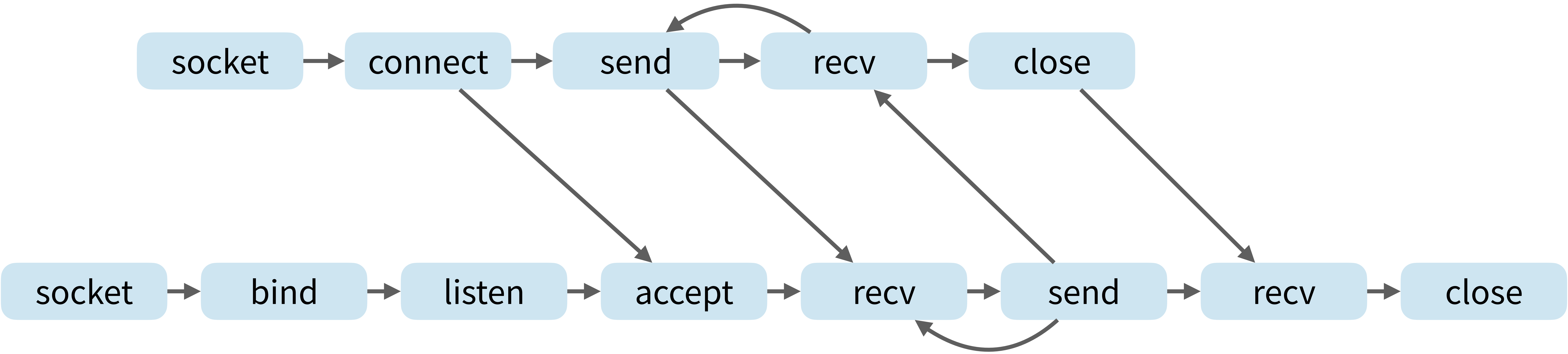
Socket represents the **communication endpoint**. It is an abstraction for user applications to access network functionalities implemented in the OS kernel.

Berkeley sockets

The de-facto socket implementation in Unix-like systems

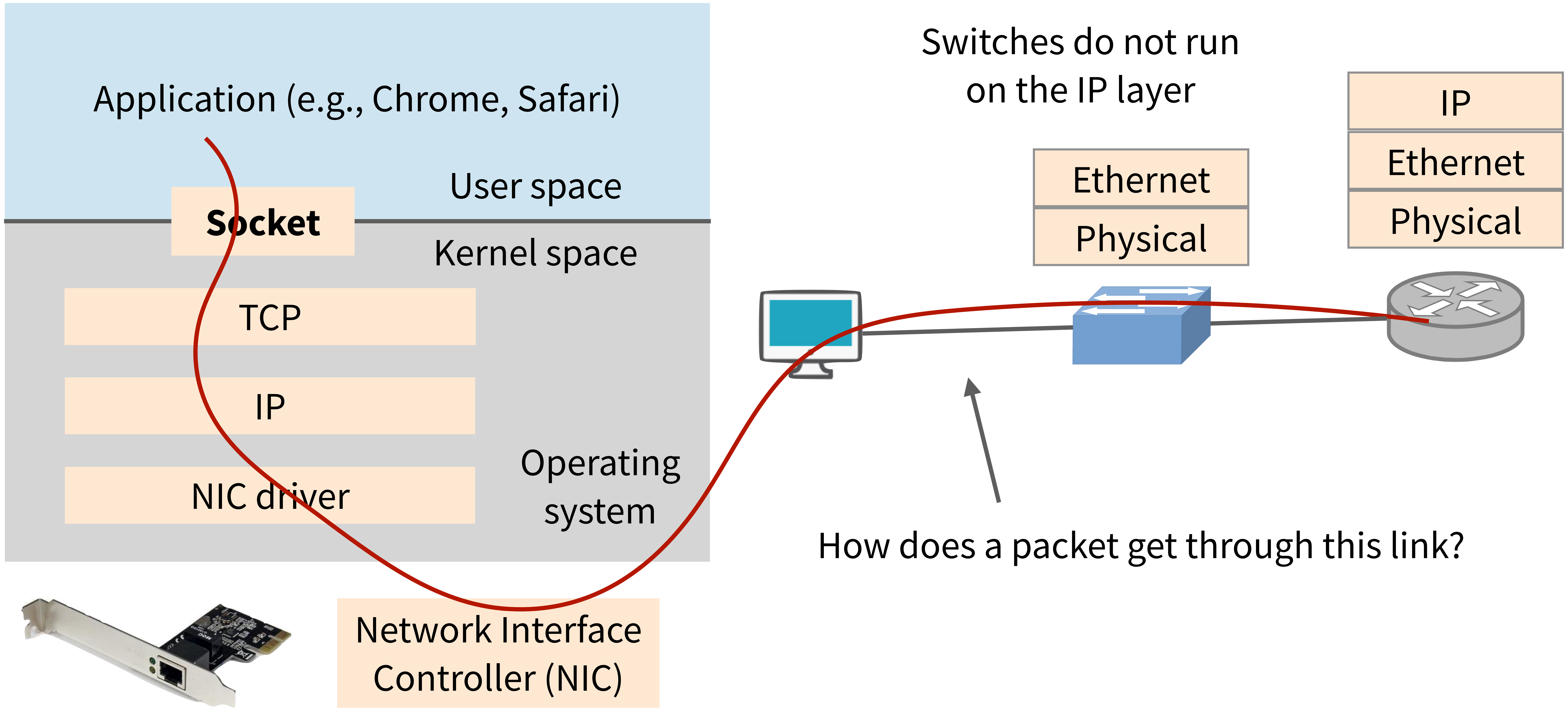
- Also known as BSD sockets or POSIX sockets

Have you ever implemented a client-server chat program?



https://man7.org/linux/man-pages/man2/socket.2.html

How to get a packet onto a link?



Application (e.g., Chrome, Safari)

User space

Socket

Kernel space

TCP

IP

NIC driver

Operating system

Network Interface Controller (NIC)

Switches do not run on the IP layer

Ethernet

Physical

IP

Ethernet

Physical

How does a packet get through this link?

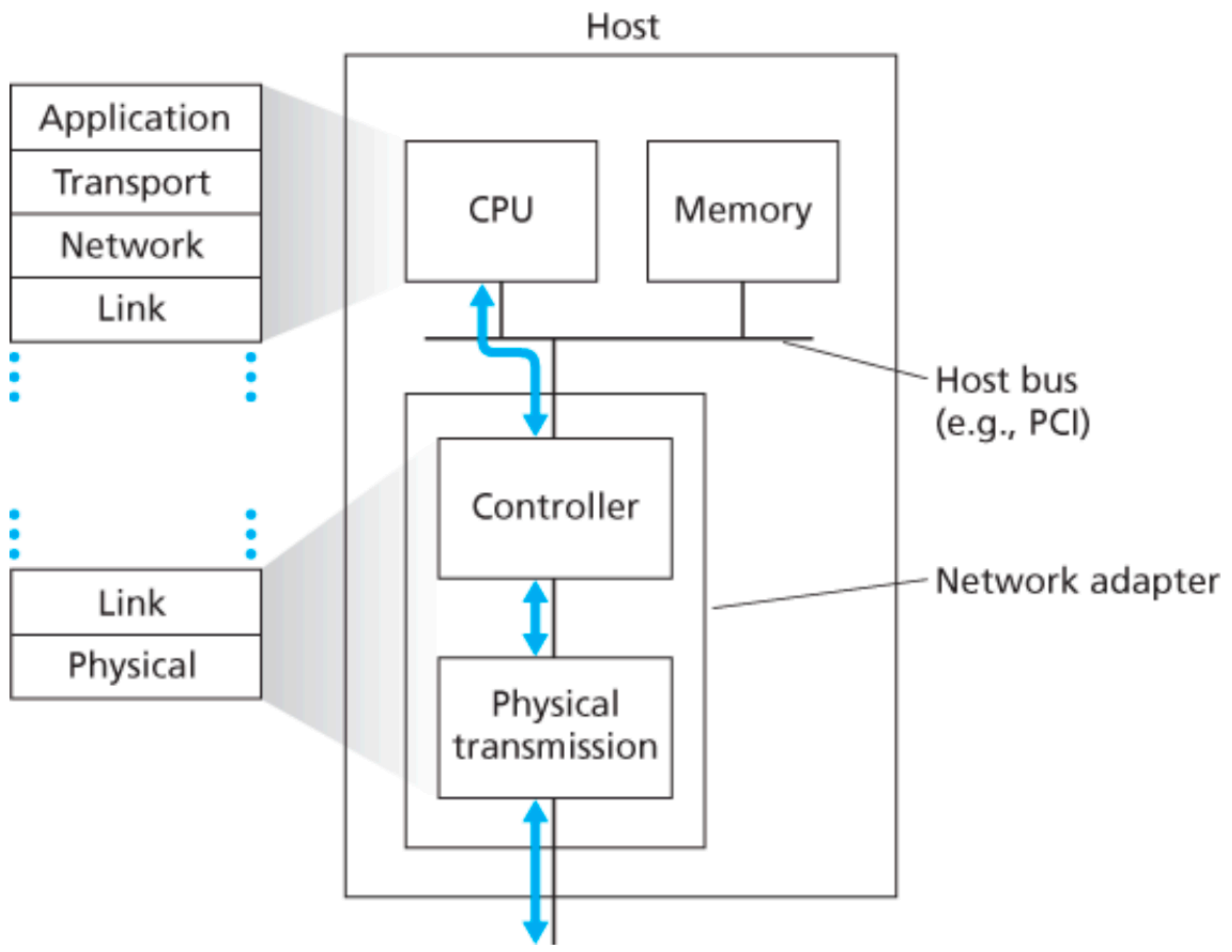
The link layer

The link layer encapsulates network-layer packets into link layer frames and transmits them onto the physical link

- Framing
- Error detection
- Medium access control

Node: any device that runs a link layer protocol, including hosts, routers, switches, and WiFi access points

Link: communication channel that connect adjacent nodes along the communication path



Kurose & Ross, Computer Networks: A Top-Down Approach.

Link layer: framing

Determines where the frame starts and ends

Byte-oriented protocols

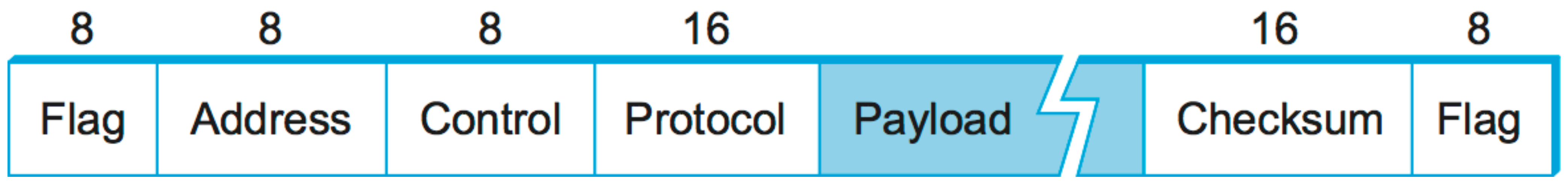
- Each frame as a collection of bytes
- Widely used Point-to-Point Protocol (PPP)

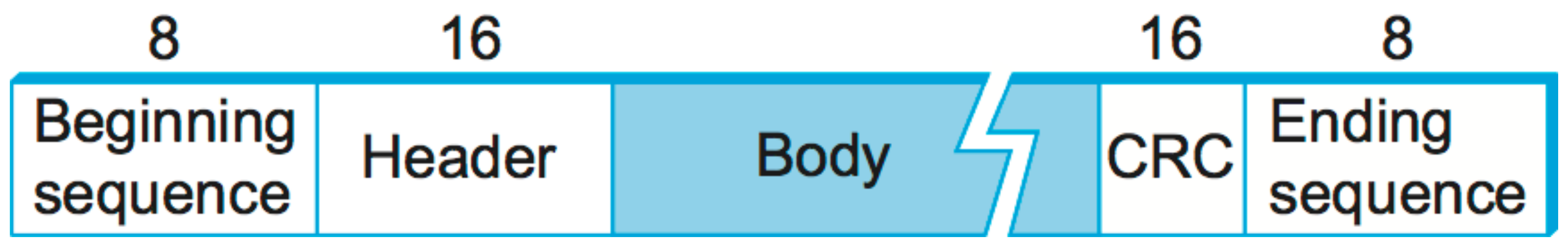
Bit-oriented protocols

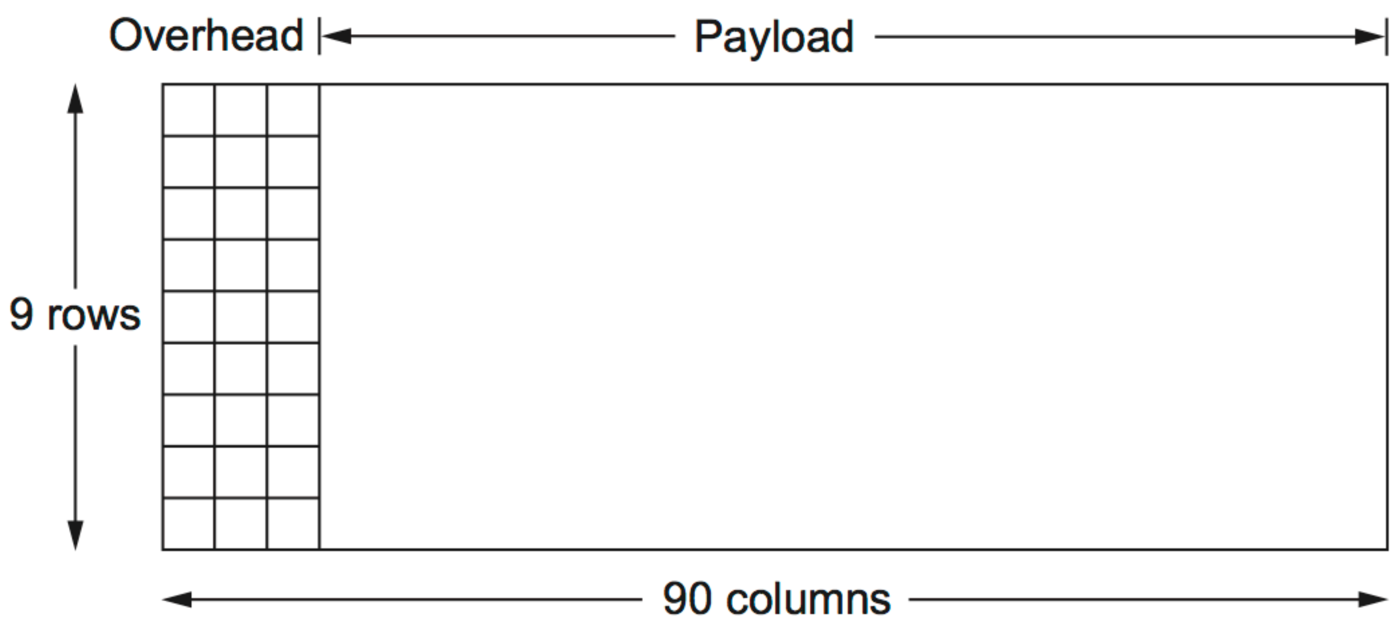
- Each frame as a collection of bits
- High-level data link control (HDLC) protocol

Clock-based framing

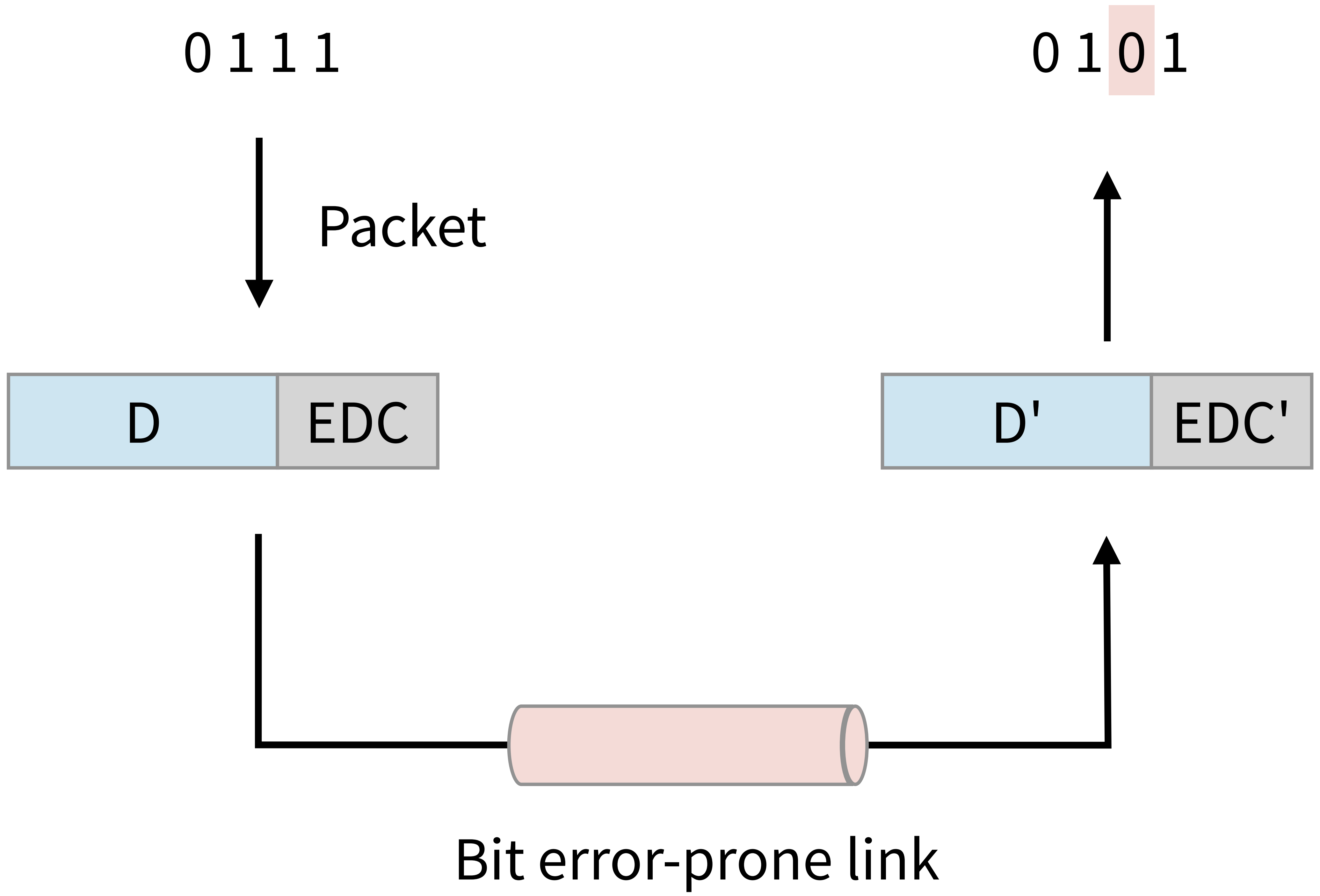
- Synchronous optical network







Link layer: error detection



Detecting bit flips in the frame and discard the frame if errors are found

There are different ways for detecting errors:

- Parity checks
- Checksumming
- Cyclic Redundancy Check (CRC)

Parity checks

Even parity scheme includes one parity bit and chooses its value such that the total number of 1's is even in the given frame

1	0	1	0	0	1	1

Single bit parity checks

1	0	1	0	0	1	1
0	0	0	1	1	1	1
1	1	0	1	0	0	0
0	1	0	0	0	0	1
1	1	0	1	1	1	0
1	1	0	1	0	1	0

Two-dimensional parity checks

How many simultaneous bit errors can each of these techniques detect?

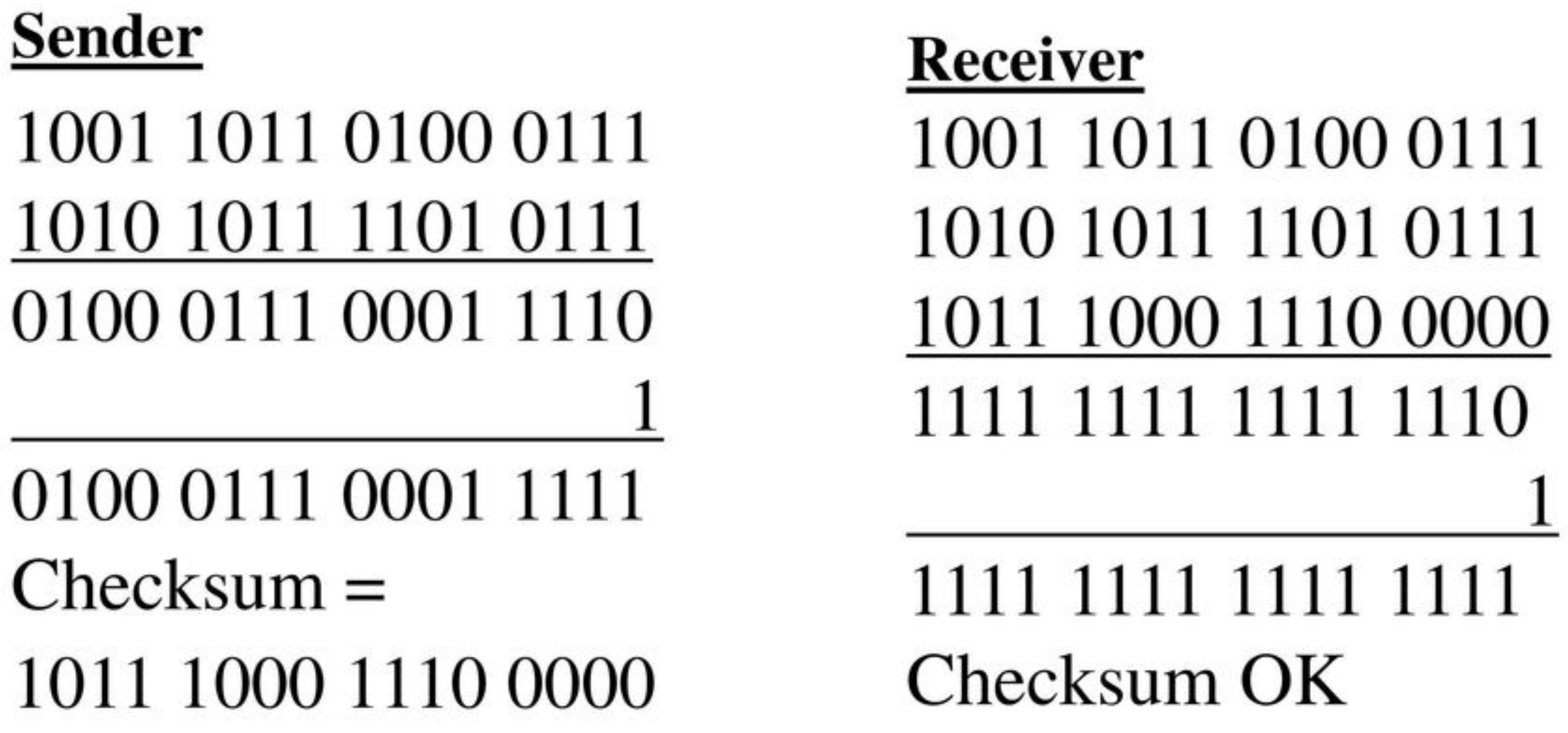
Checksumming

Treats bits as sequence of integers and sums the integers (complementary)

RFC 1071

- Calculate the checksum with the checksum field omitted
- Verify the checksum with the checksum field filled

Typically applied on TCP/UDP header + payload and IP header

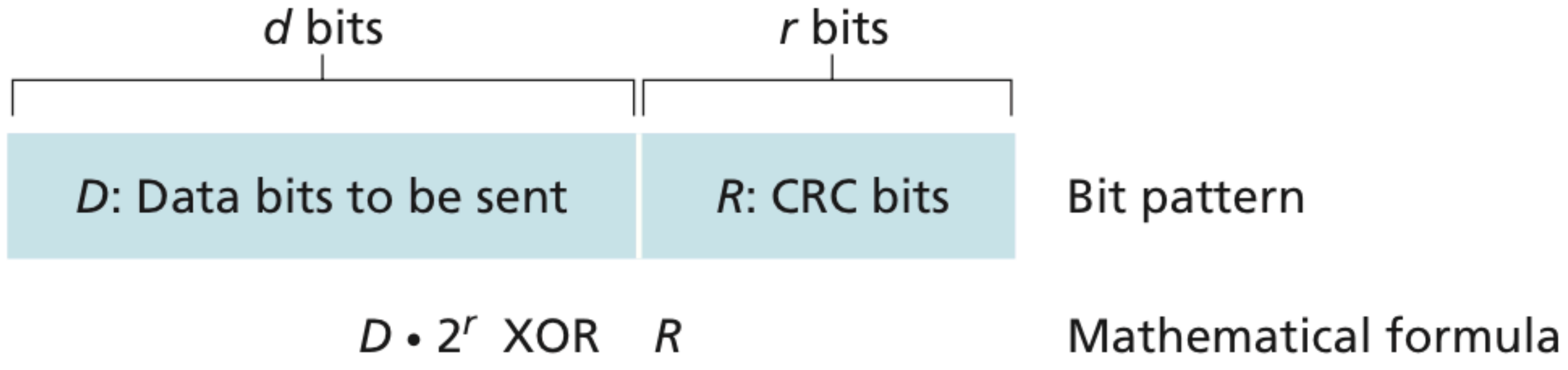


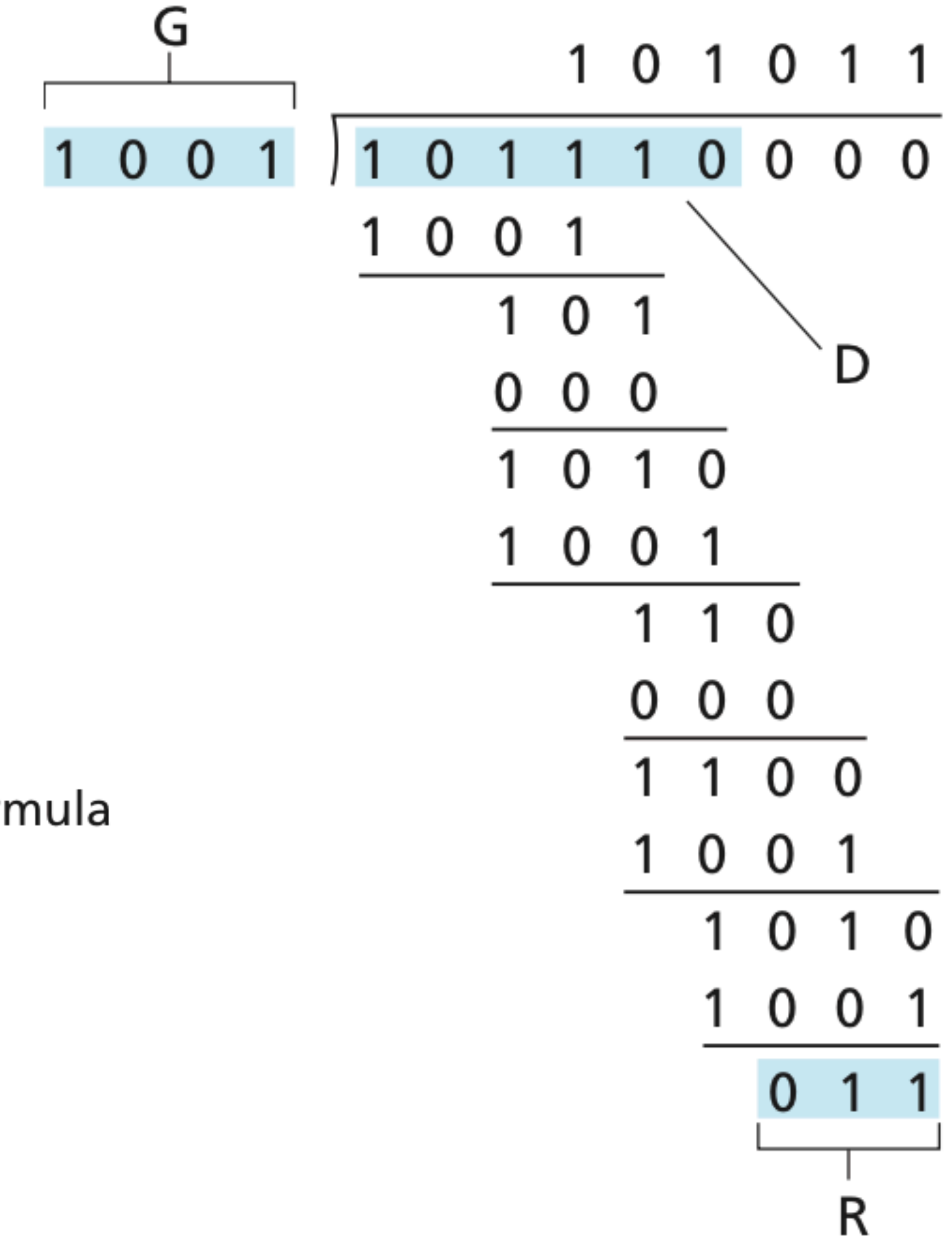
```
Sender                      Receiver
1001 1011 0100 0111         1001 1011 0100 0111
1010 1011 1101 0111         1010 1011 1101 0111
-------------------         1011 1000 1110 0000
0100 0111 0001 1110         -------------------
                  1         1111 1111 1111 1110
-------------------                           1
0100 0111 0001 1111         -------------------
Checksum =                  1111 1111 1111 1111
1011 1000 1110 0000         Checksum OK
```

Cyclic redundancy check (CRC)

Applies polynomial arithmetic operations on the input bit string

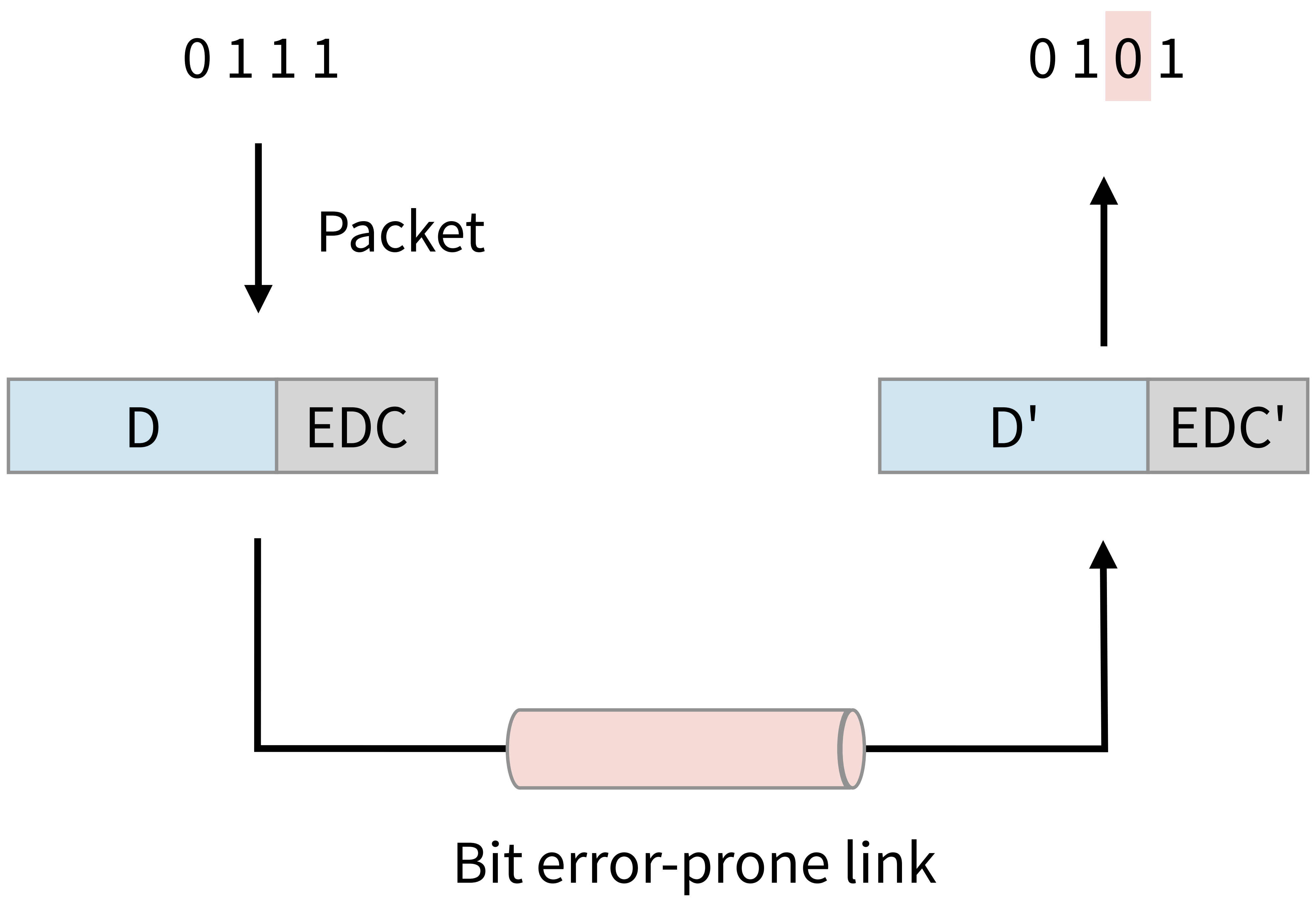
- Smaller chance of collisions, but more computation-intensive
- Adopted by the link layer and implemented in hardware NIC





Why not using MD5, SHA256, etc.?

Link layer: error detection



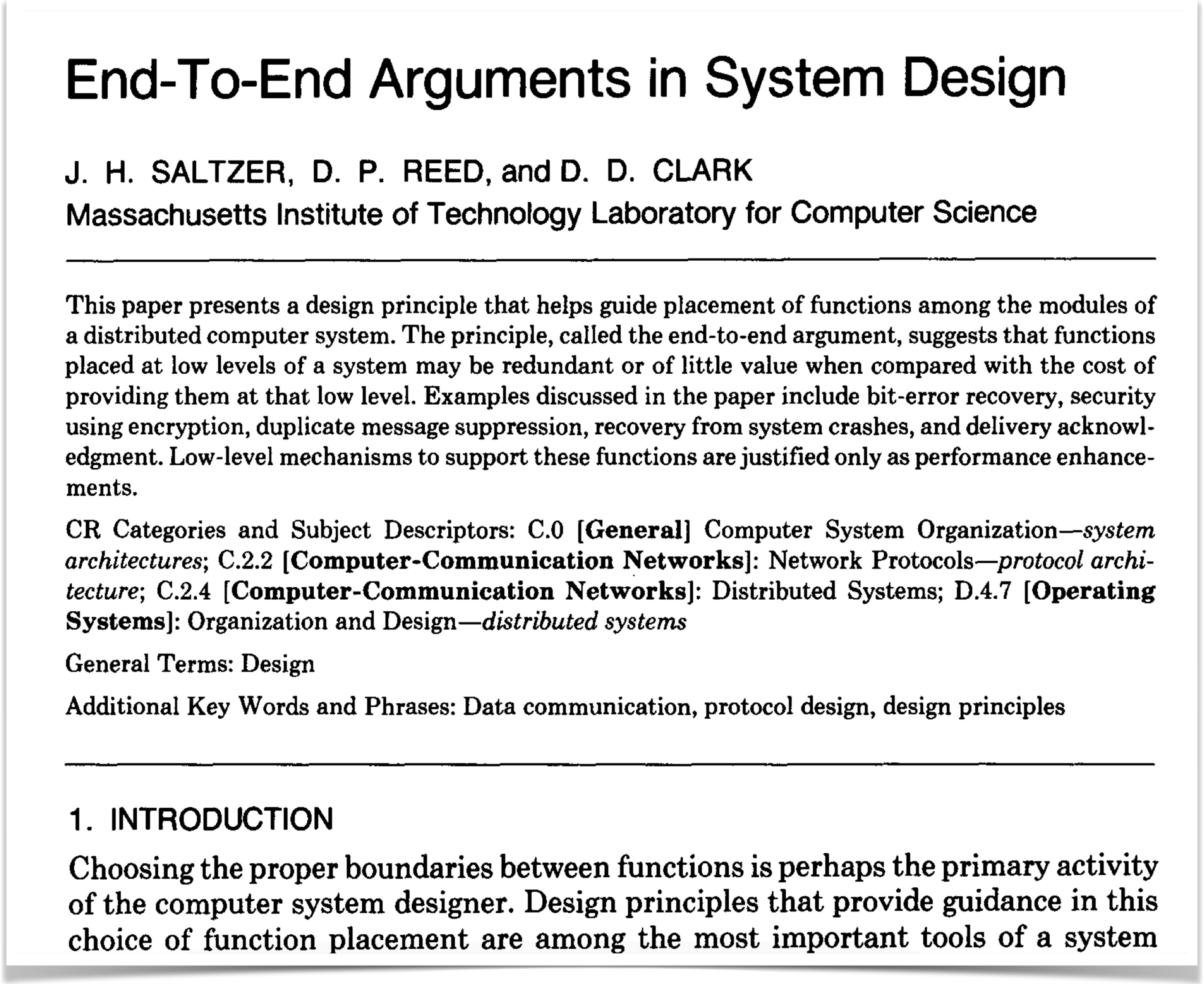
Detecting bit flips in the frame and discard the frame if errors are found

There are different ways for detecting errors:

- Parity checks
- Checksumming
- Cyclic Redundancy Check (CRC)

Why error detection in the link layer given that errors will be checked at upper-layers as well?

The end-to-end argument



End-To-End Arguments in System Design

J. H. SALTZER, D. P. REED, and D. D. CLARK
Massachusetts Institute of Technology Laboratory for Computer Science

This paper presents a design principle that helps guide placement of functions among the modules of a distributed computer system. The principle, called the end-to-end argument, suggests that functions placed at low levels of a system may be redundant or of little value when compared with the cost of providing them at that low level. Examples discussed in the paper include bit-error recovery, security using encryption, duplicate message suppression, recovery from system crashes, and delivery acknowledgment. Low-level mechanisms to support these functions are justified only as performance enhancements.

CR Categories and Subject Descriptors: C.0 [**General**] Computer System Organization—*system architectures*; C.2.2 [**Computer-Communication Networks**]: Network Protocols—*protocol architecture*; C.2.4 [**Computer-Communication Networks**]: Distributed Systems; D.4.7 [**Operating Systems**]: Organization and Design—*distributed systems*

General Terms: Design

Additional Key Words and Phrases: Data communication, protocol design, design principles

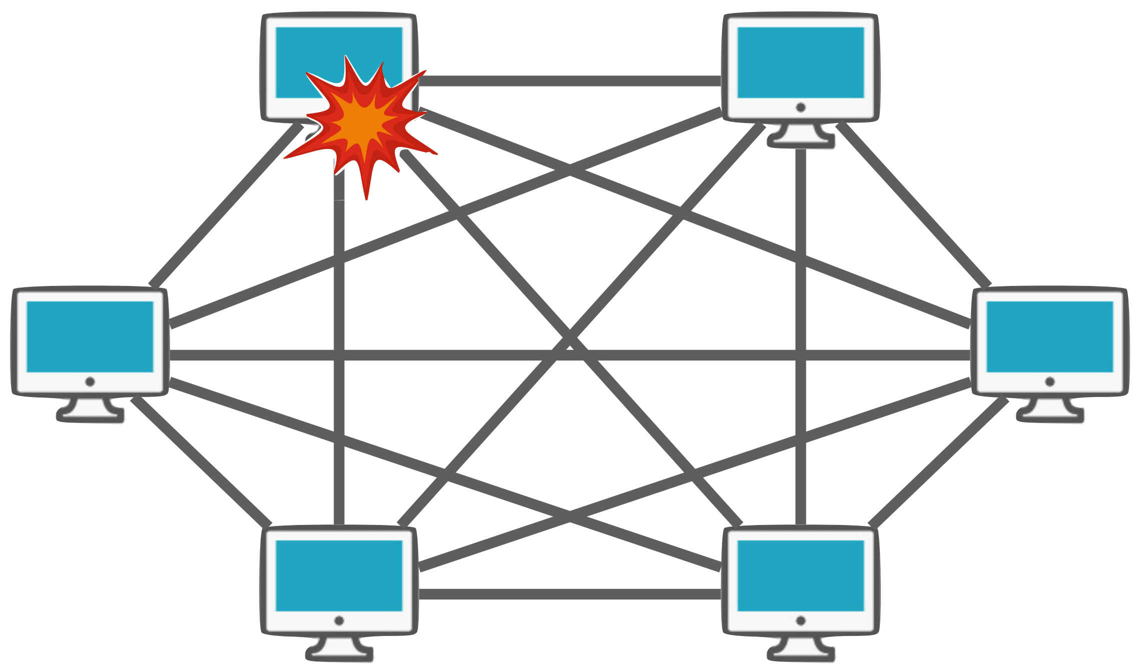
1. INTRODUCTION

Choosing the proper boundaries between functions is perhaps the primary activity of the computer system designer. Design principles that provide guidance in this choice of function placement are among the most important tools of a system

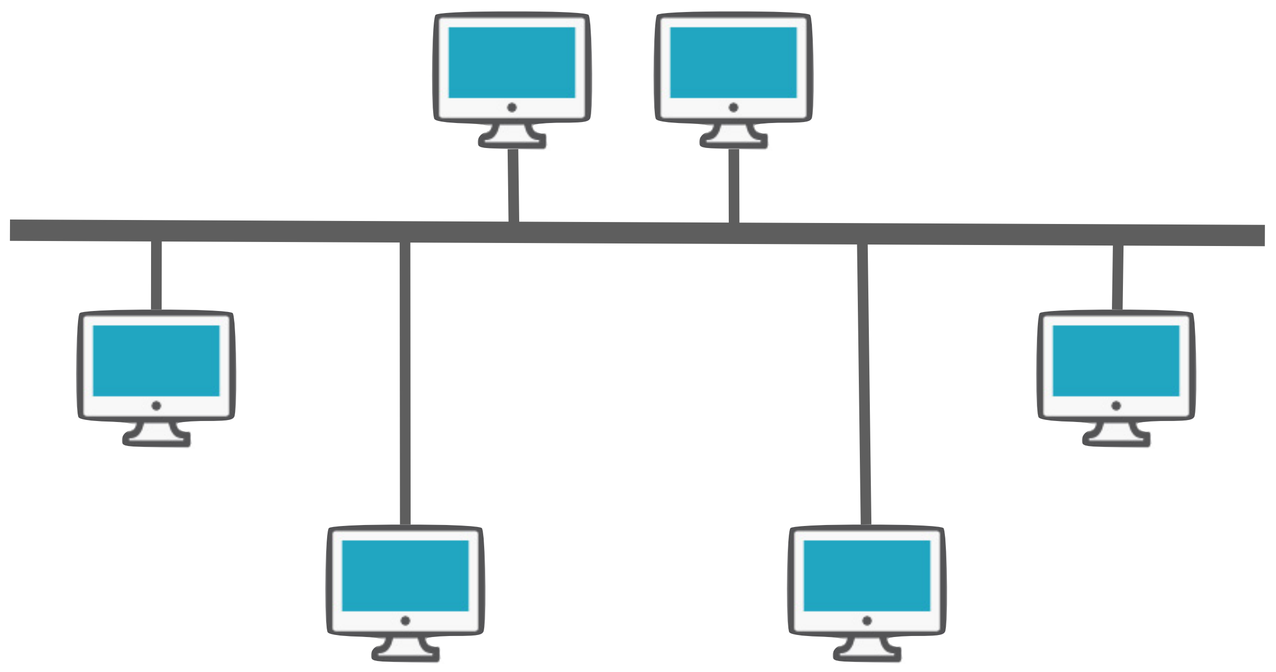
ACM TOCS 1984

"The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the endpoints of the communication system. Therefore, providing that questioned function as a feature of the communication system itself is not possible. (Sometimes an incomplete version of the function provided by the communication system may be useful as a **performance enhancement**.)"

How to connect more than two computers?



Naïve approach: full mesh with direct PPP links connecting all nodes → **does not scale!**

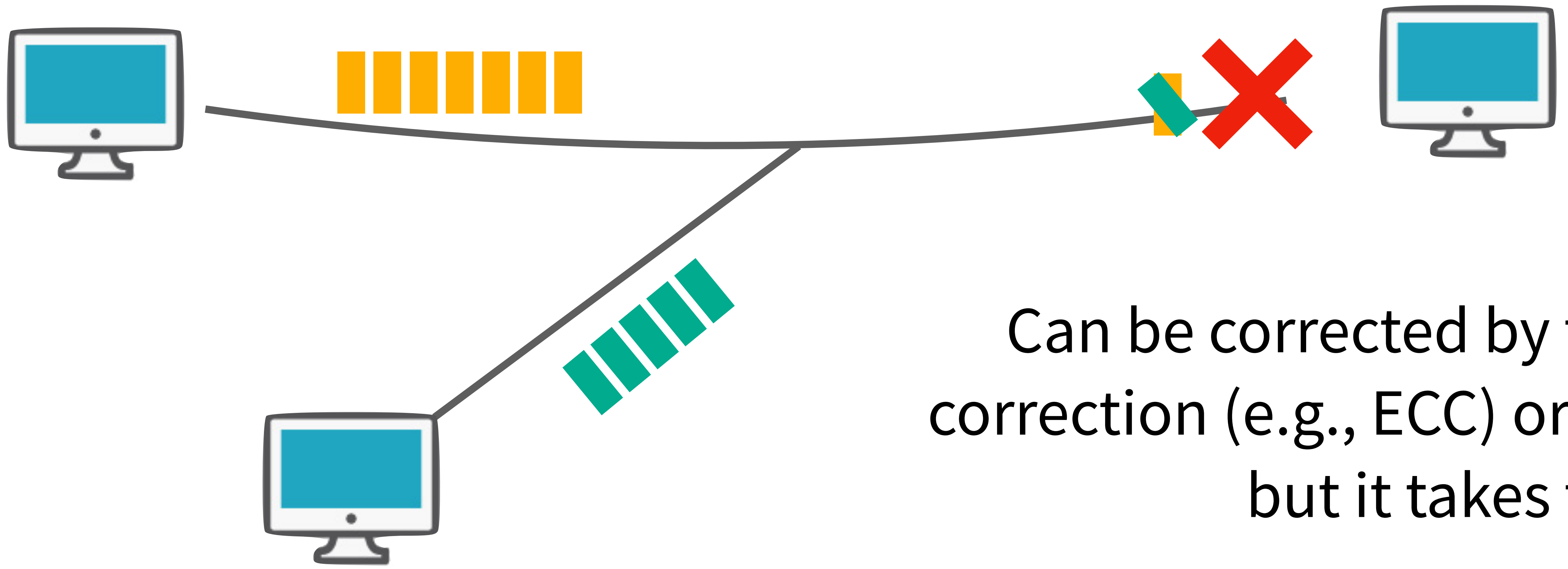


A slightly better approach: shared medium

What could be the problem?

Shared broadcast medium

Signals may collide with each other, leading to frame corruptions



Can be corrected by forward error correction (e.g., ECC) or retransmission, but it takes time

Examples: Ethernet and Wireless LAN

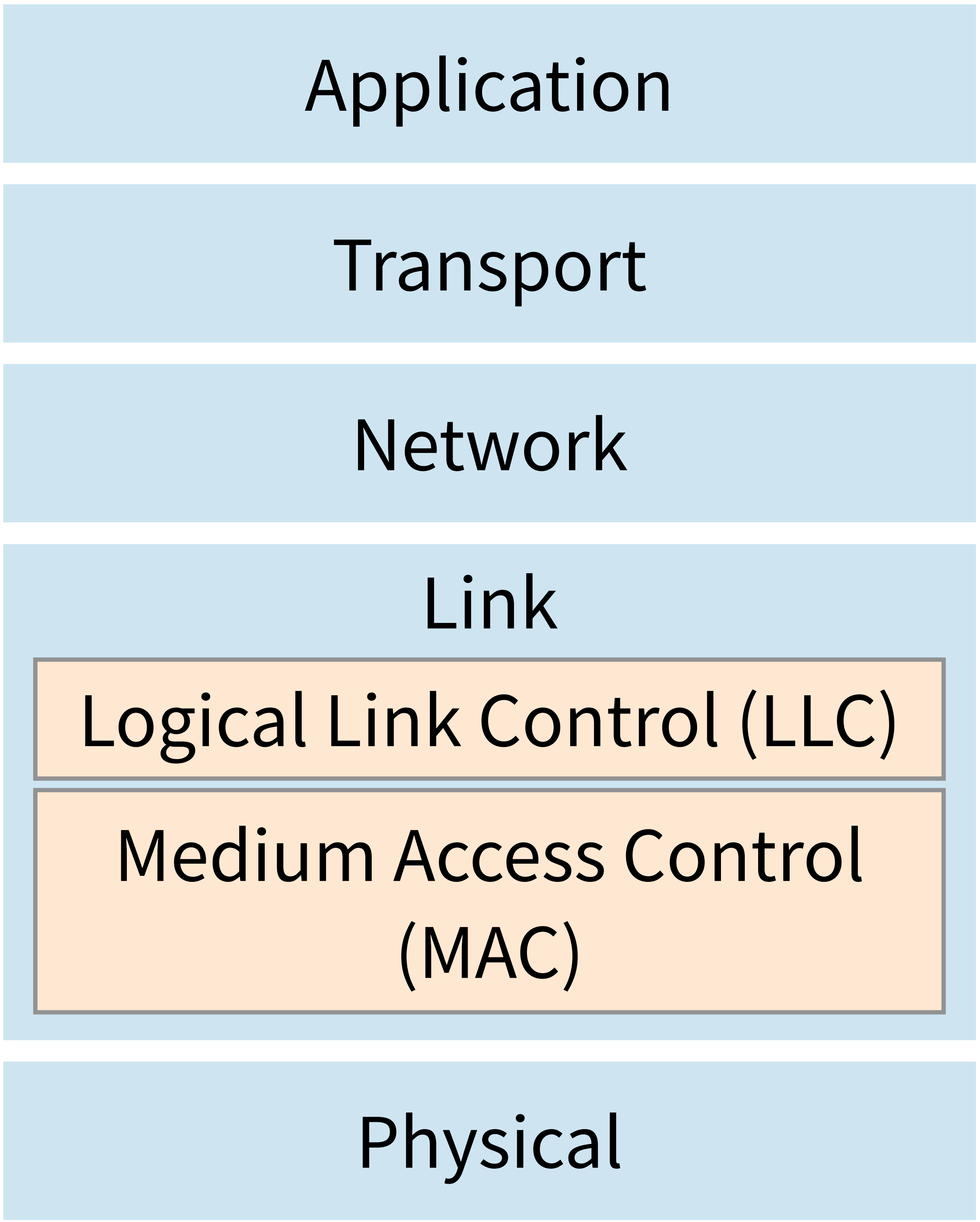
Multiple access protocol

Important **principles** to follow:

- Work-conserving: maximum utilization
- Fairness: equal average utilization
- Decentralized: no master node (single point of failure)
- Simple: inexpensive to implement

Protocols falling into three categories:

- Channel partitioning: TDM, FDM, CDMA
- Random access: Slotted ALOHA, ALOHA, CSMA, CSMA/CD
- Taking-turns: polling, token-passing



Heavy adoptions in wireless networks (WLAN, cellular, LoRa, etc.) due to the shared-medium nature

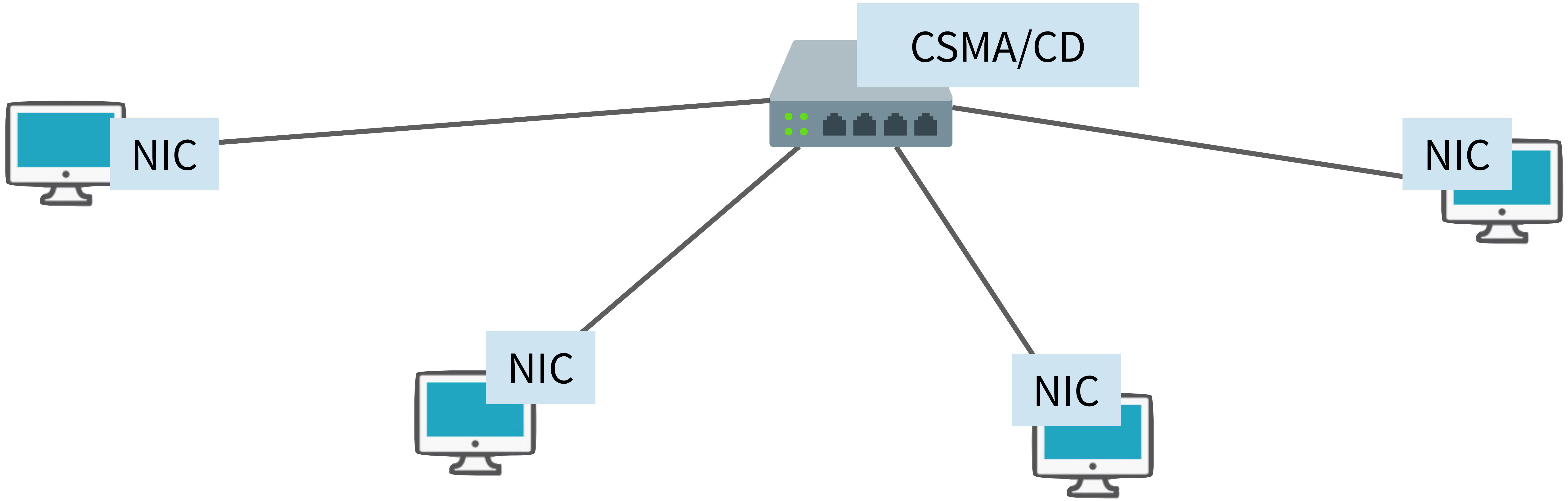
Ethernet

IEEE 802.3

A family of networking technologies commonly used in Local Area Networks (LAN) and other networks

Hub: replicates signals to all ports except the one that signals are received on

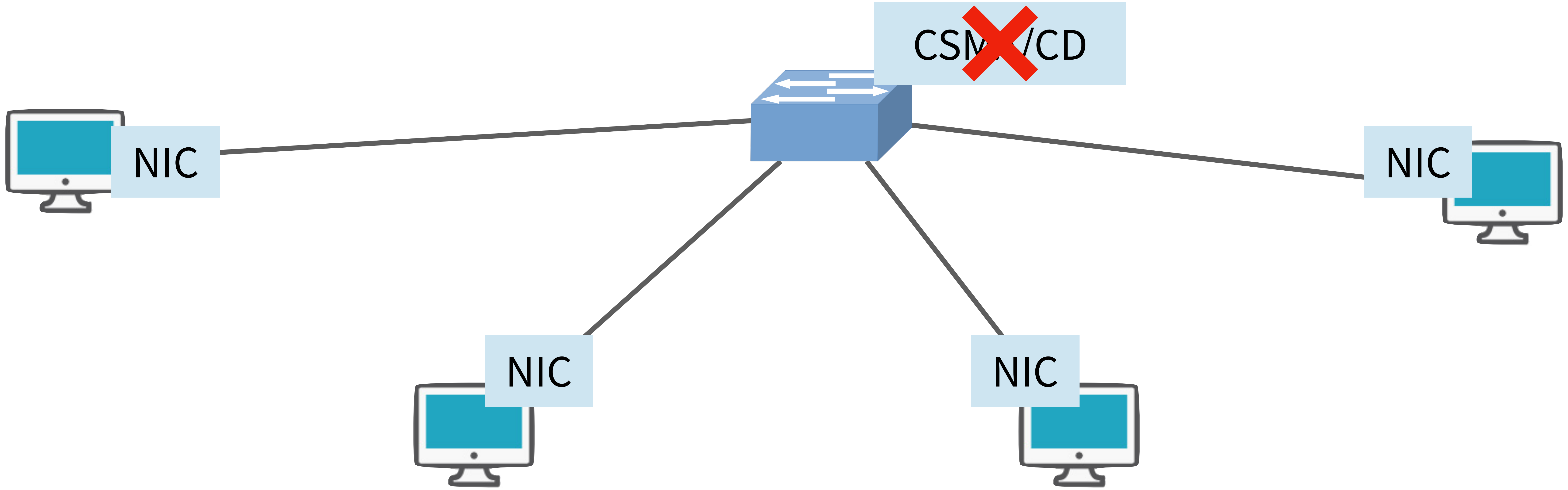
OBSOLETE



Switched Ethernet

Different Ethernet segments are interconnected with switches (that work on the link layer)

Switch: creates Ethernet segments and forwards frames between segments based on the MAC address

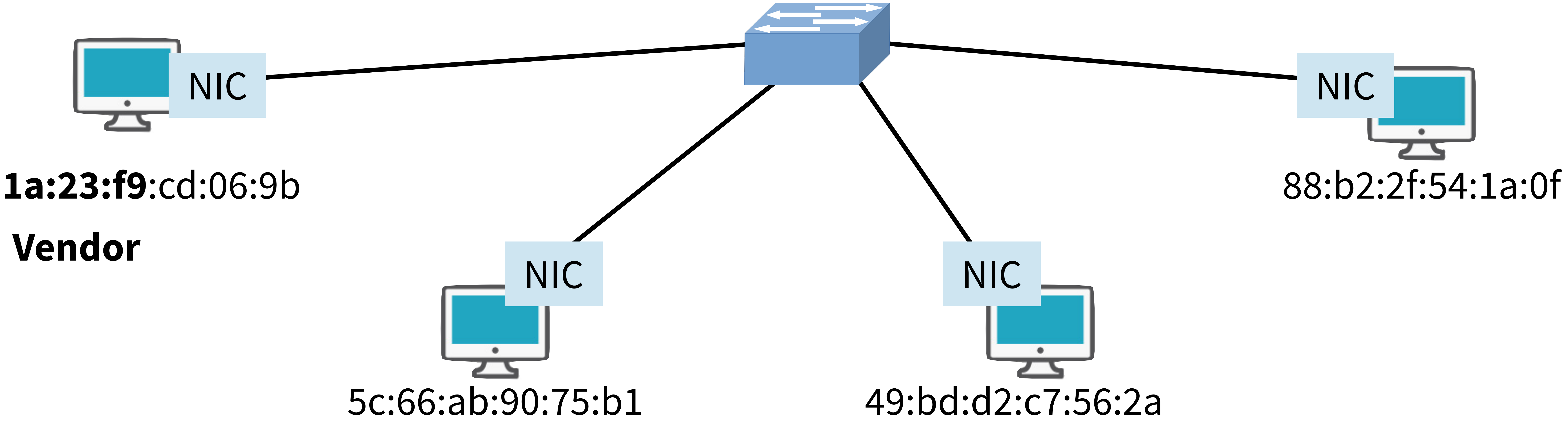


Ethernet MAC address

6-byte long, unique among all network adapters, managed by IEEE

Broadcast MAC address
ff:ff:ff:ff:ff:ff

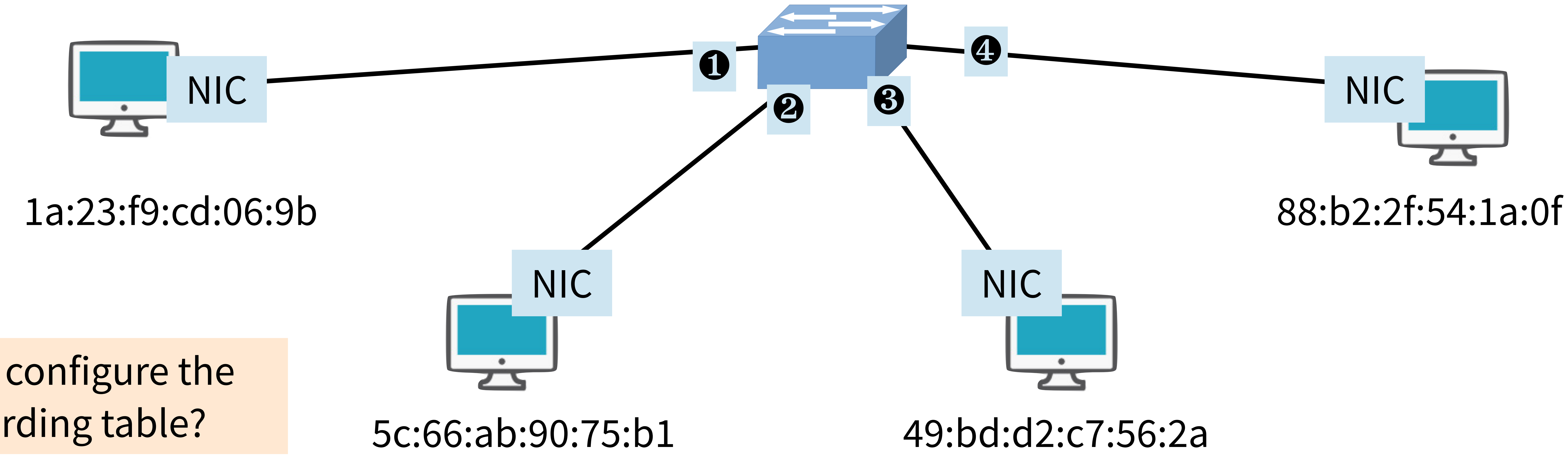
Do switches have MAC addresses? Why?



Link layer switches

Switches forward/broadcast/drop frames based on a switch table (a.k.a. forwarding table) and operate transparently to the hosts, i.e., no need for MAC addresses on them

MAC	Interface	Time
88:b2:2f:54:1a:0f	4	9:32
5c:66:ab:90:75:b1	2	9:34



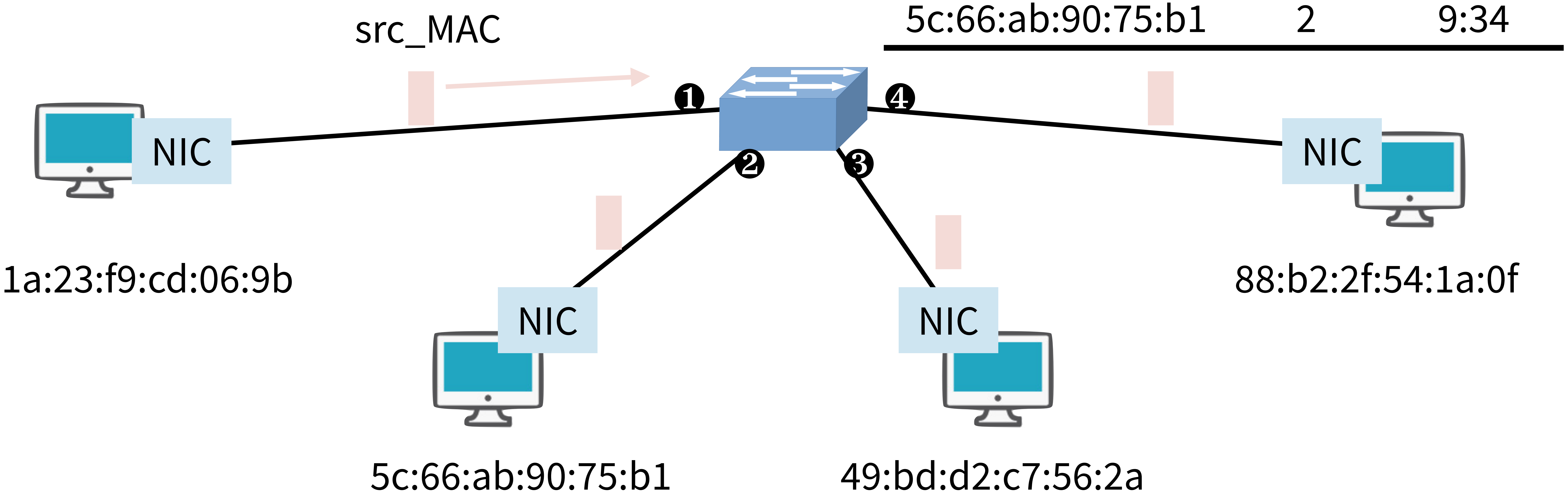
How to configure the forwarding table?

Switches are self-learning

Switches learn the forwarding table automatically, without any human intervention → plug-and-play

- Initially empty forwarding table
- For each incoming frame received on an interface, store the **source MAC** of the frame and map it to the **receiving interface**, with the current time
- An entry is deleted if the aging time has elapsed

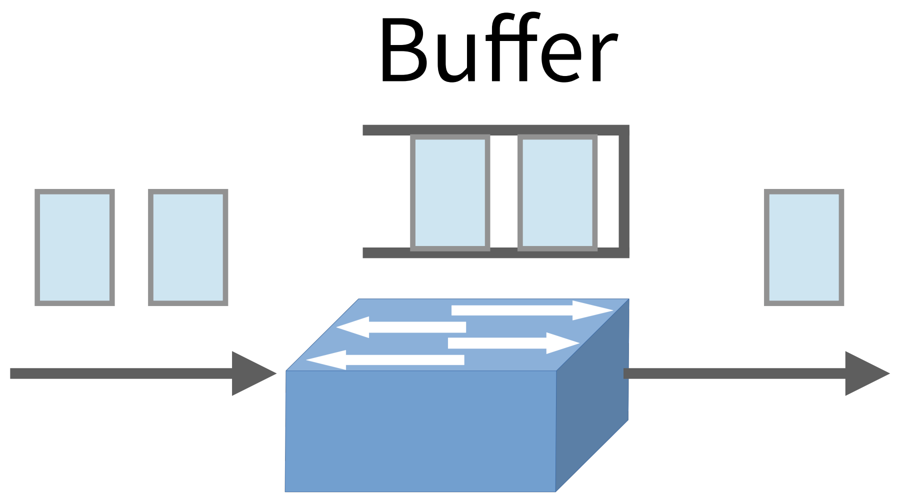
MAC	Interfac	Time
88:b2:2f:54:1a:0f	4	9:32
5c:66:ab:90:75:b1	2	9:34



This week’s project lab is to implement a learning switch by yourself!

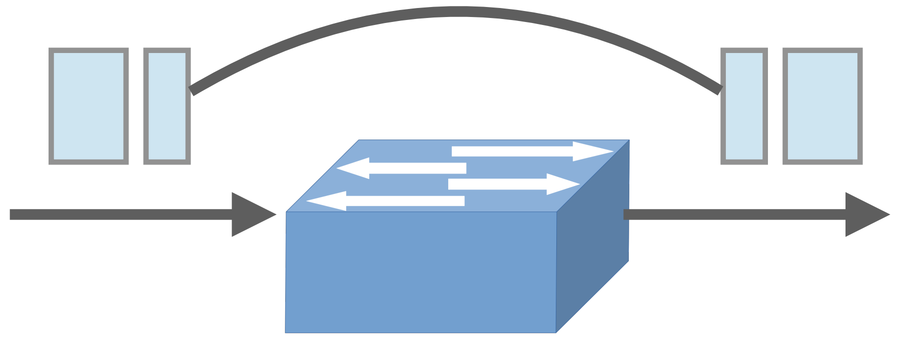
Store-and-forward vs. cut-through

Store-and-forward



Packets are received in full, buffered, and forwarded onto the output link.

Cut-through



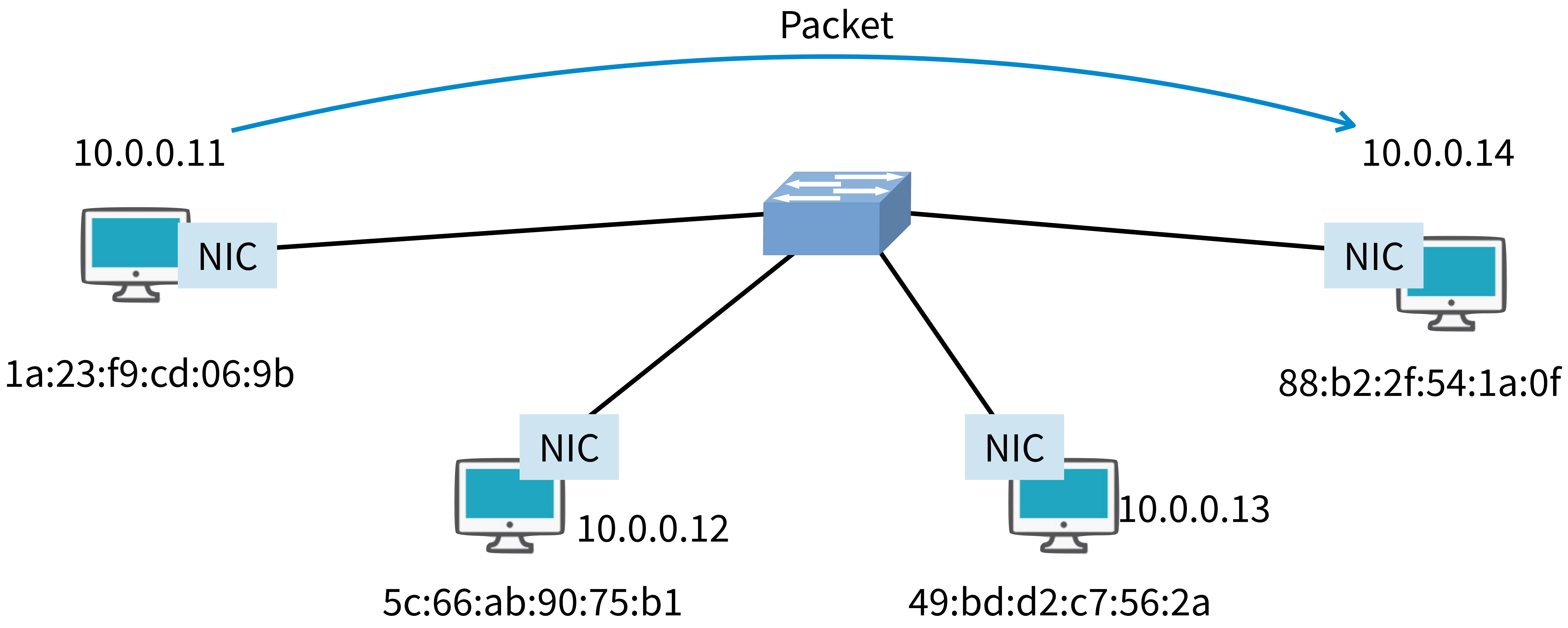
Once lookup is done, packet receiving and sending happen at the same time.

What are the pros and cons of each approach?

Questions?

How to obtain destination MAC addresses?

Assume we want to send a packet from 10.0.0.11 to 10.0.0.14 on the same subnet. The first step is to know where to forward the packet (or more precisely the frame containing the packet), i.e., obtaining the MAC address of the destination.

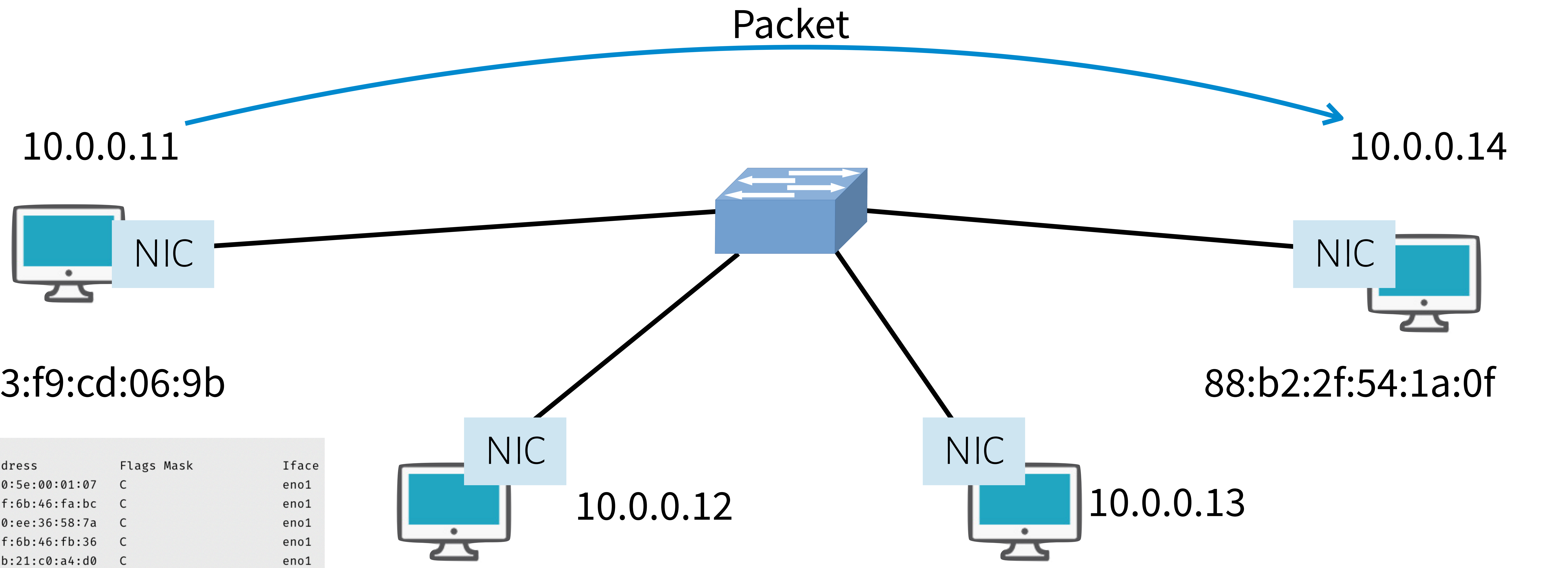


ARP

RFC 826

ARP query: Whoever has the IP address 10.0.0.14, please tell me your MAC address

ARP reply: that is me, my MAC address is 88:b2:2f:54:1a:0f

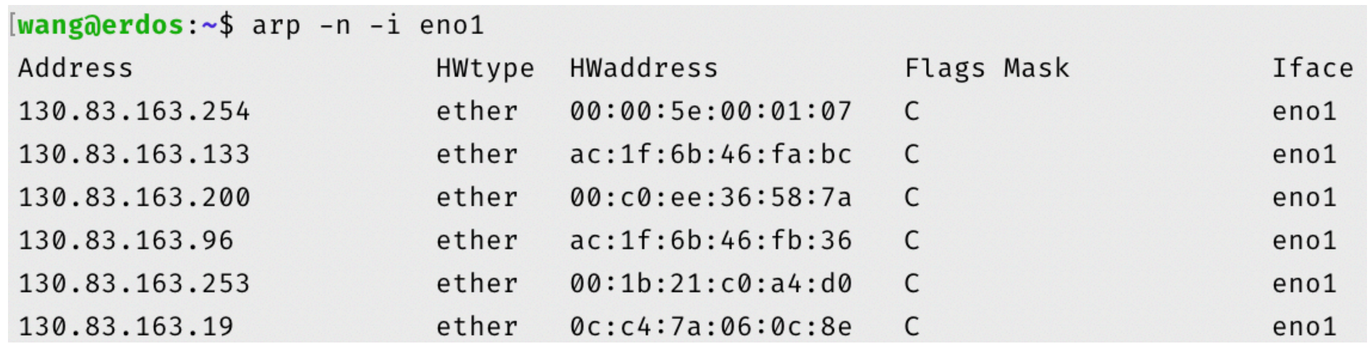


1a:23:f9:cd:06:9b

88:b2:2f:54:1a:0f

5c:66:ab:90:75:b1

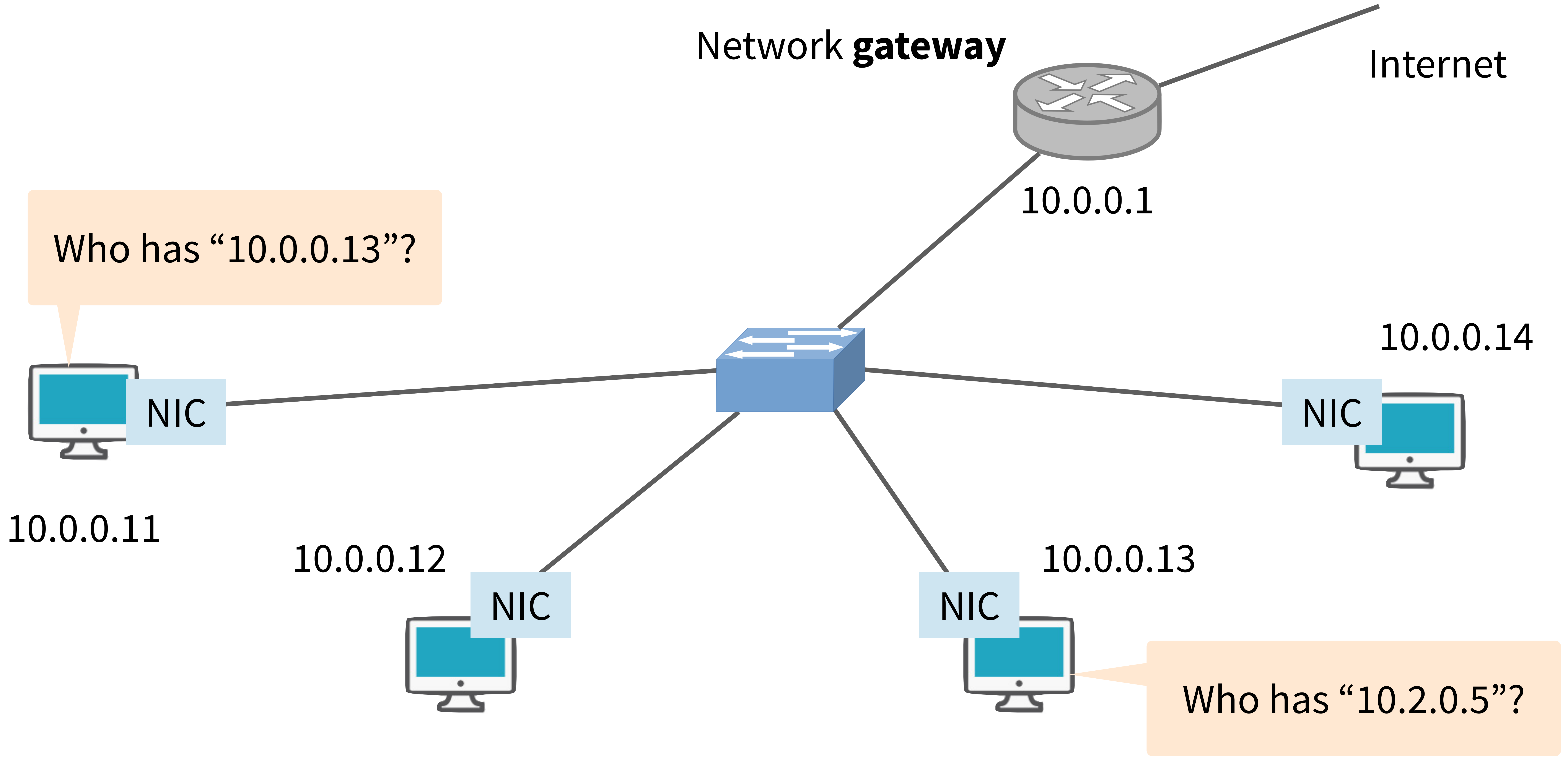
49:bd:d2:c7:56:2a



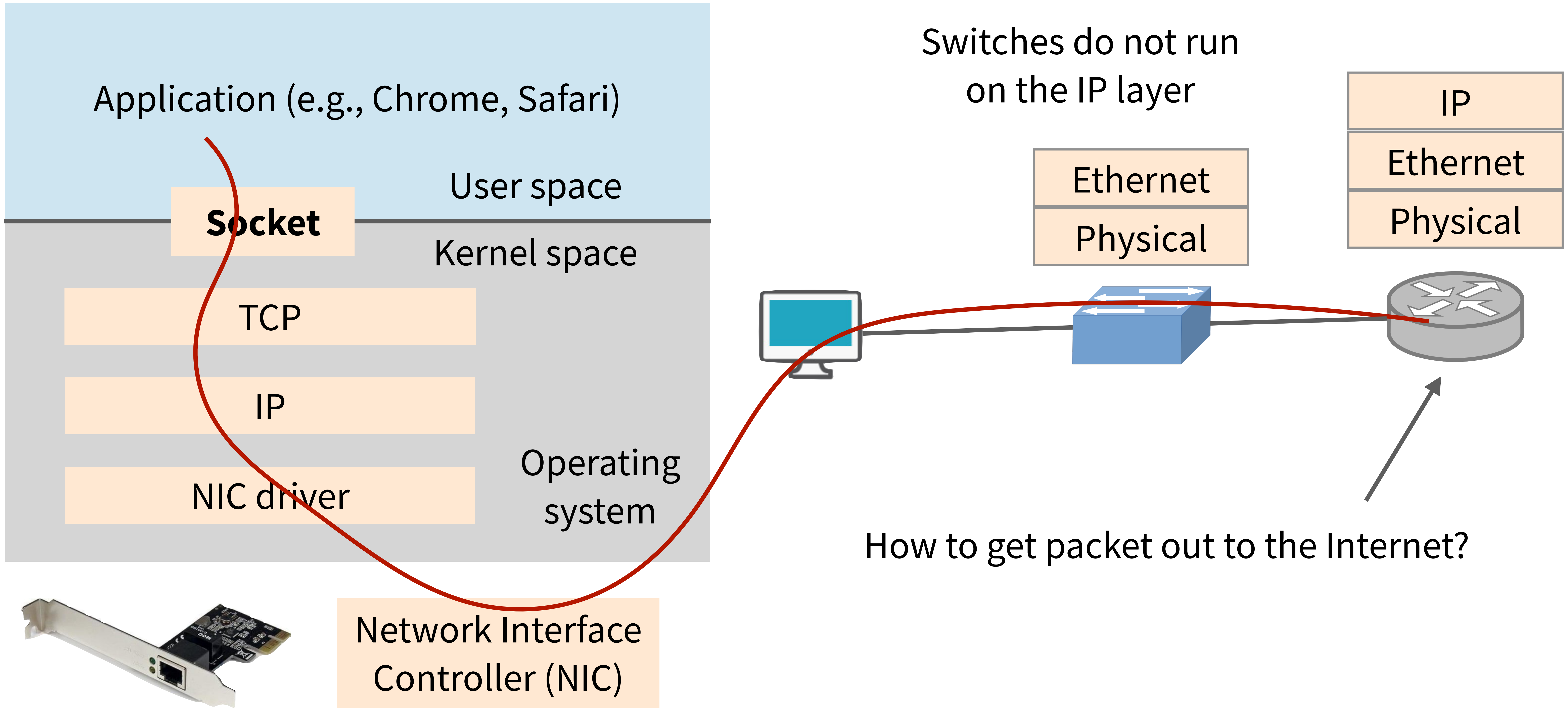
```
wang@erdos:~$ arp -n -i eno1
Address                  HWtype  HWaddress           Flags Mask            Iface
130.83.163.254           ether   00:00:5e:00:01:07   C                     eno1
130.83.163.133           ether   ac:1f:6b:46:fa:bc   C                     eno1
130.83.163.200           ether   00:c0:ee:36:58:7a   C                     eno1
130.83.163.96            ether   ac:1f:6b:46:fb:36   C                     eno1
130.83.163.253           ether   00:1b:21:c0:a4:d0   C                     eno1
130.83.163.19            ether   0c:c4:7a:06:0c:8e   C                     eno1
```

ARP table

ARP exercise



How to get the packet out to the Internet?



Application (e.g., Chrome, Safari)

User space

Socket

Kernel space

TCP

IP

NIC driver

Operating system

Network Interface Controller (NIC)

Switches do not run on the IP layer

Ethernet

Physical

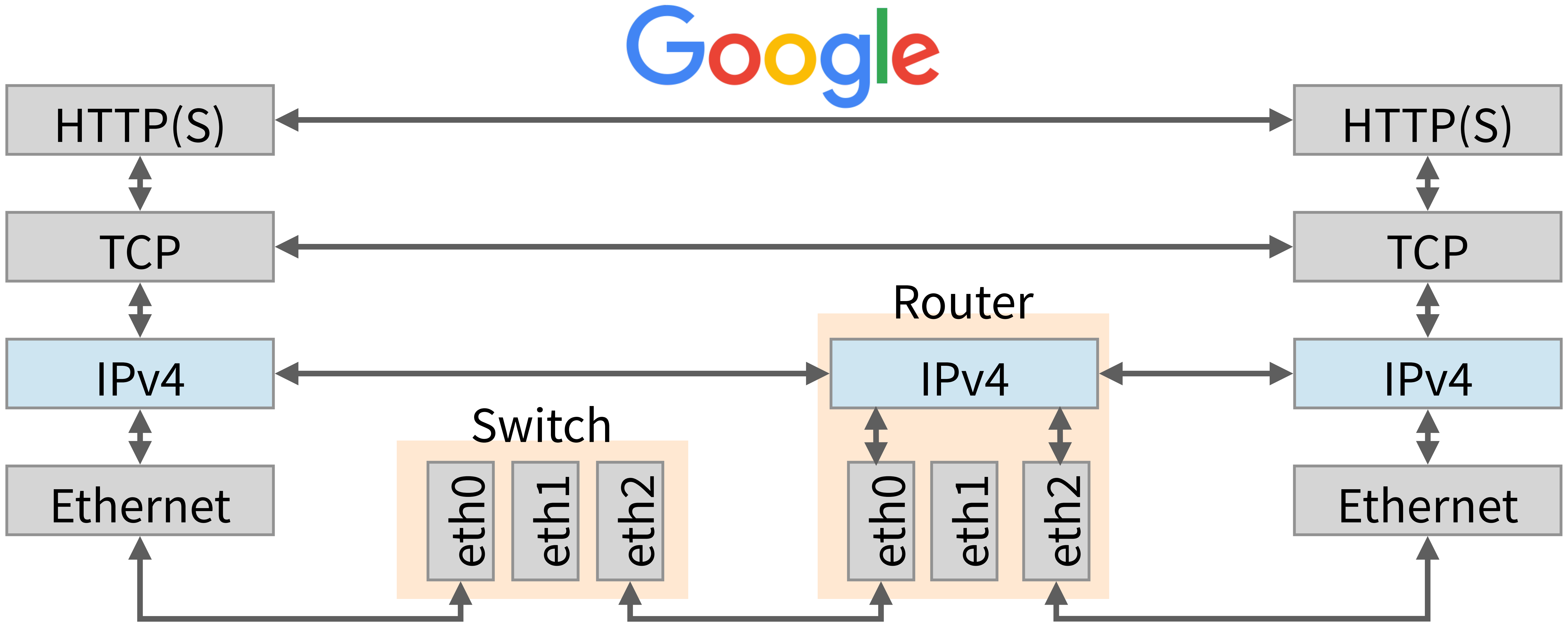
IP

Ethernet

Physical

How to get packet out to the Internet?

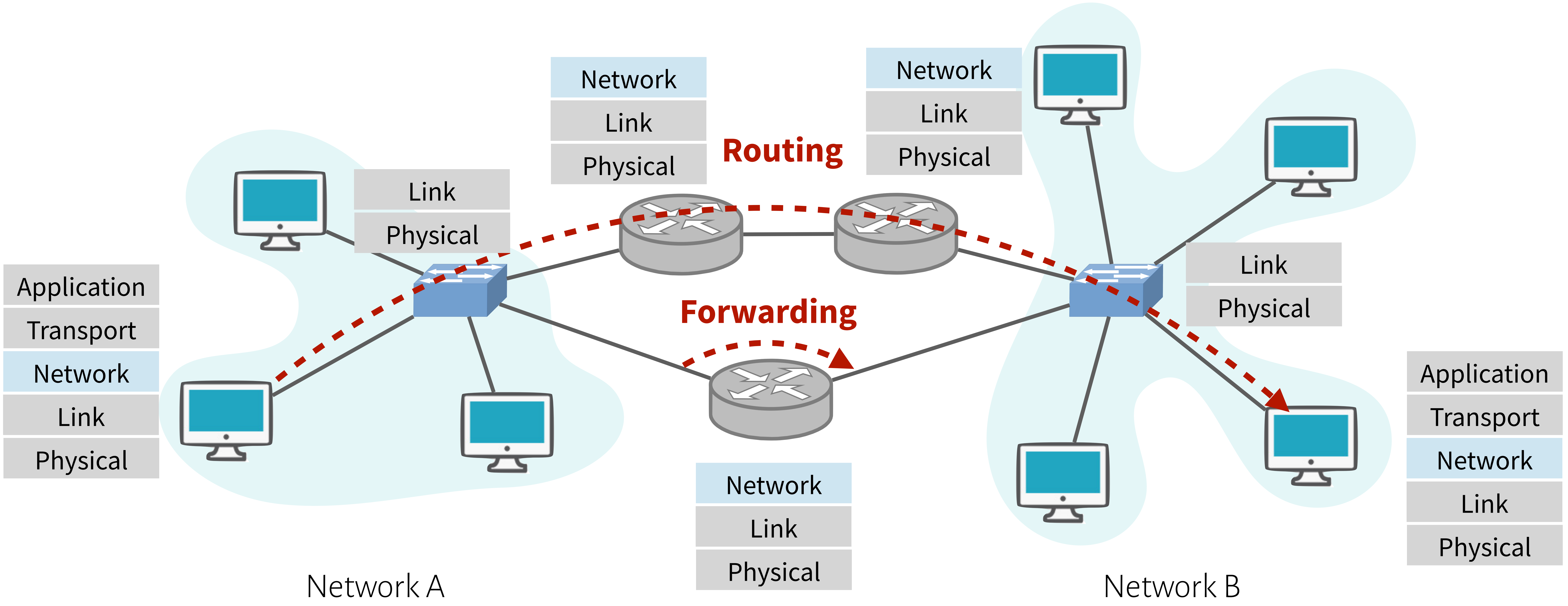
The network layer



Separation of network-layer functionalities:

- **Data plane**: forwarding – router-local action of moving packets from an input link to an appropriate output link
- **Control plane**: routing – network-wide process of determining the end-to-end path that packets take from source to destination

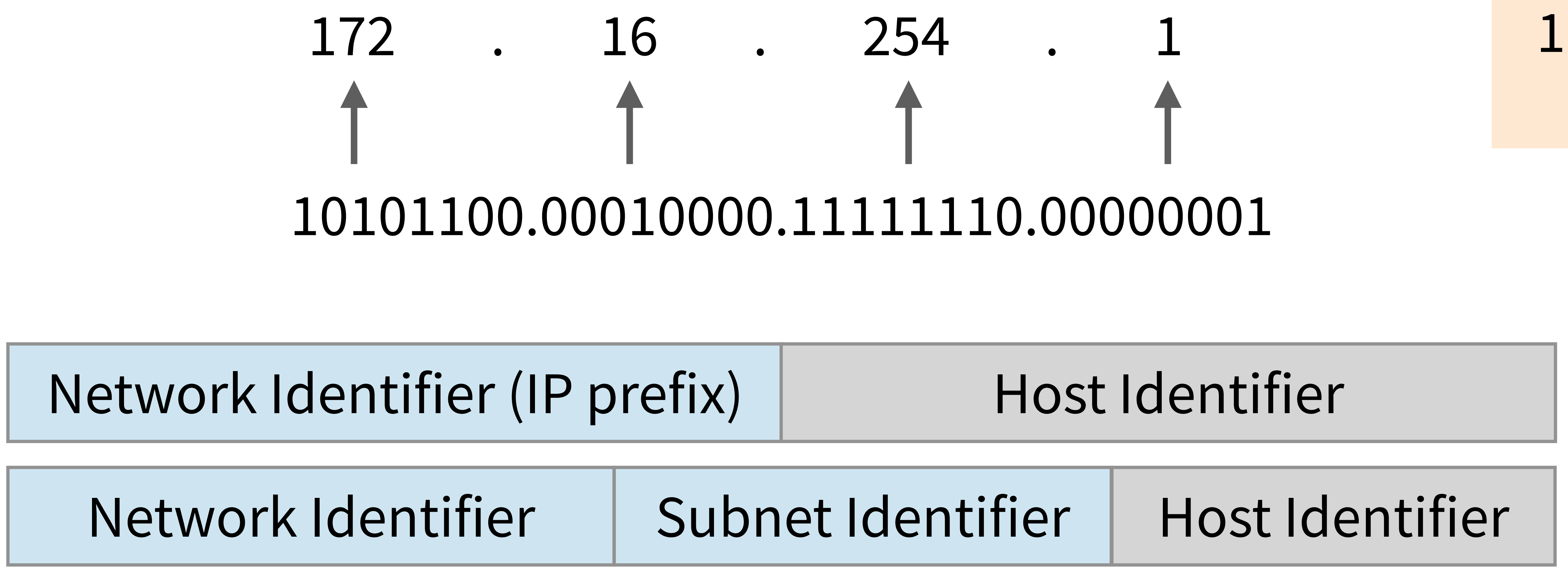
Forwarding and routing



Network layer address: IPv4

RFC7020

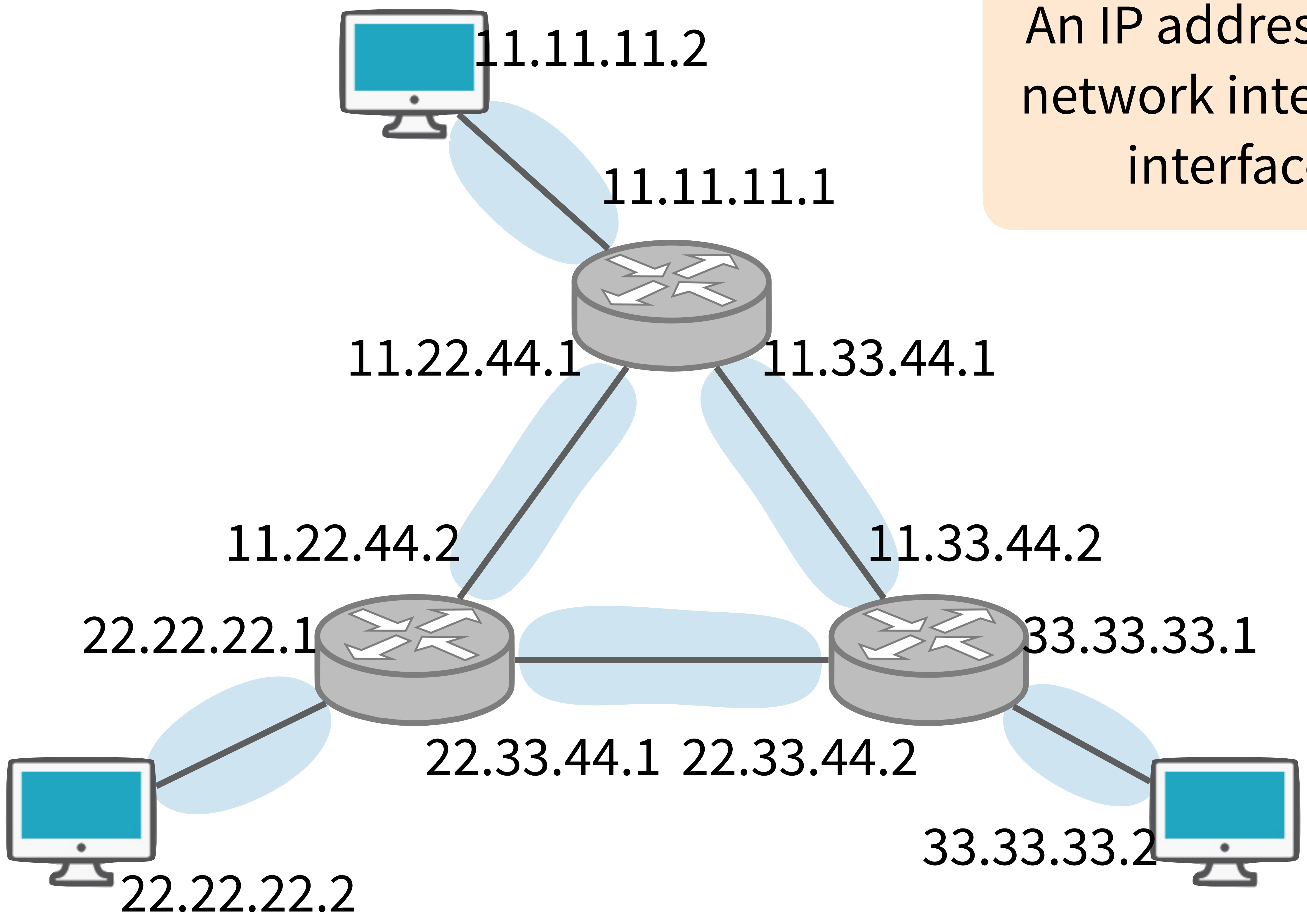
Private addresses:
10.0.0.0/8, 172.16.0.0/12,
192.168.0.0/16



Classless Inter-Domain Routing (CIDR) notation: 10.0.0.1/24

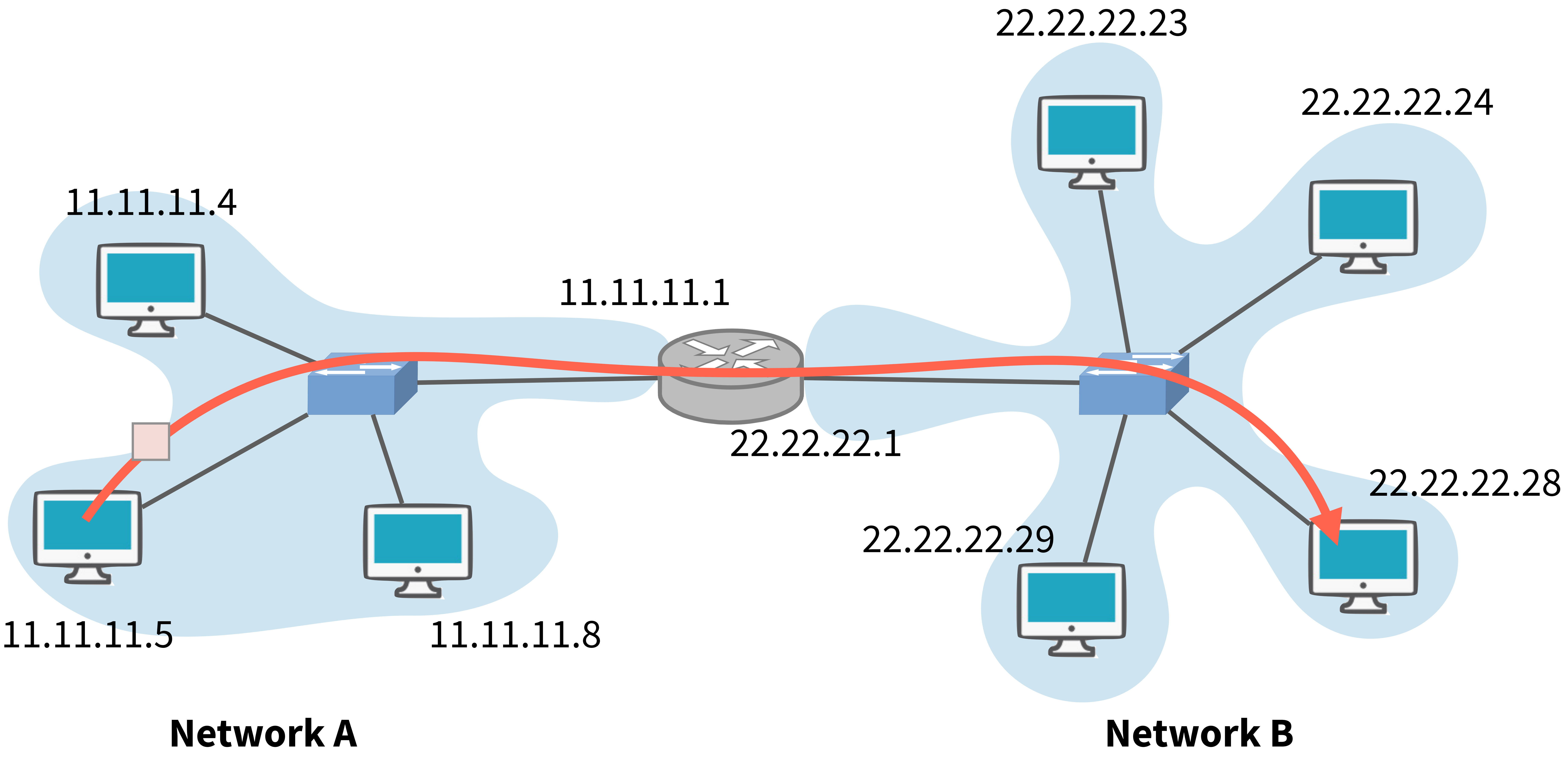
Subnet mask notation: 255.255.255.0

Routers interconnecting subnets

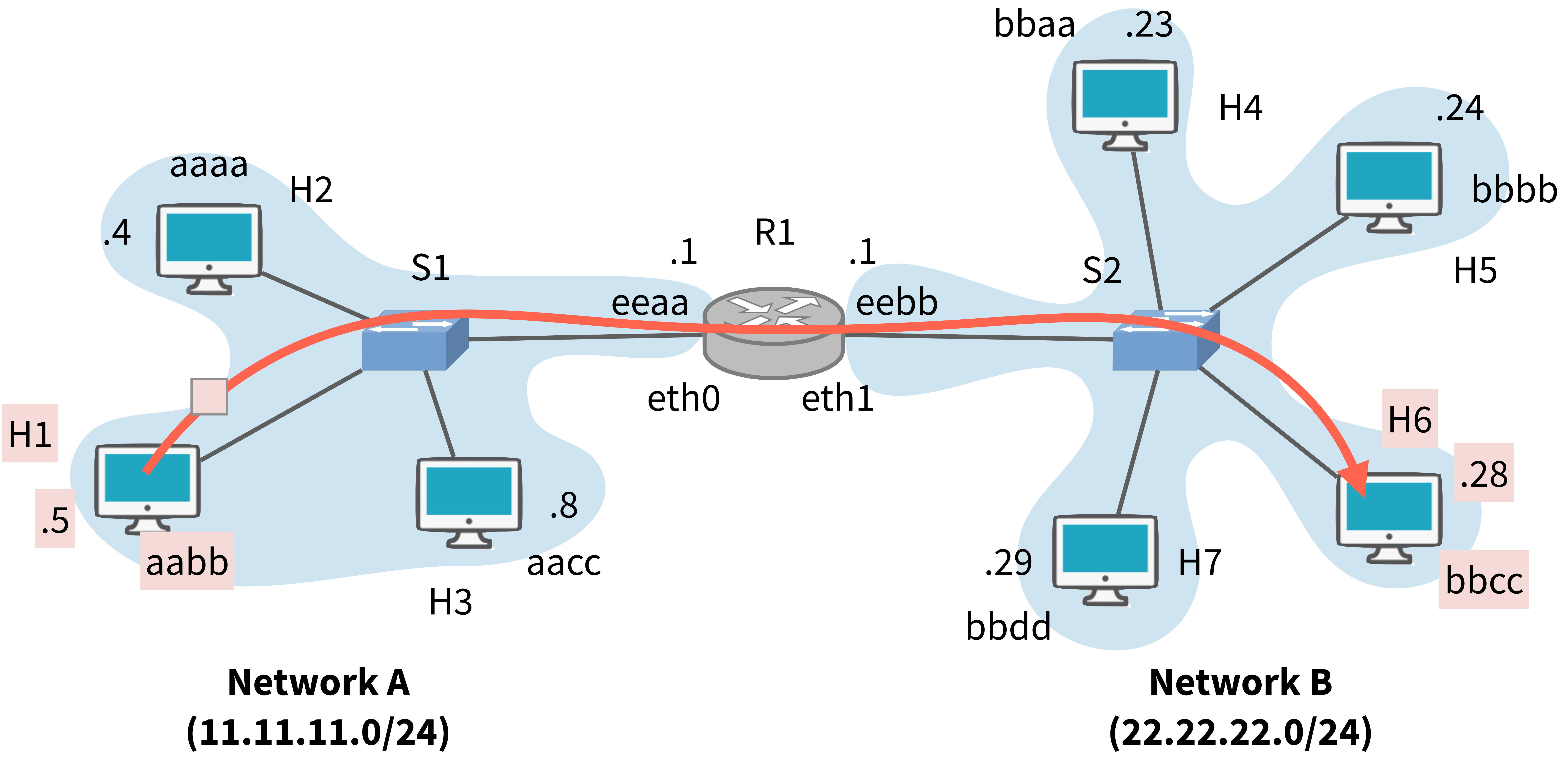


An IP address is assigned to every network interface and each router interface forms a subnet.

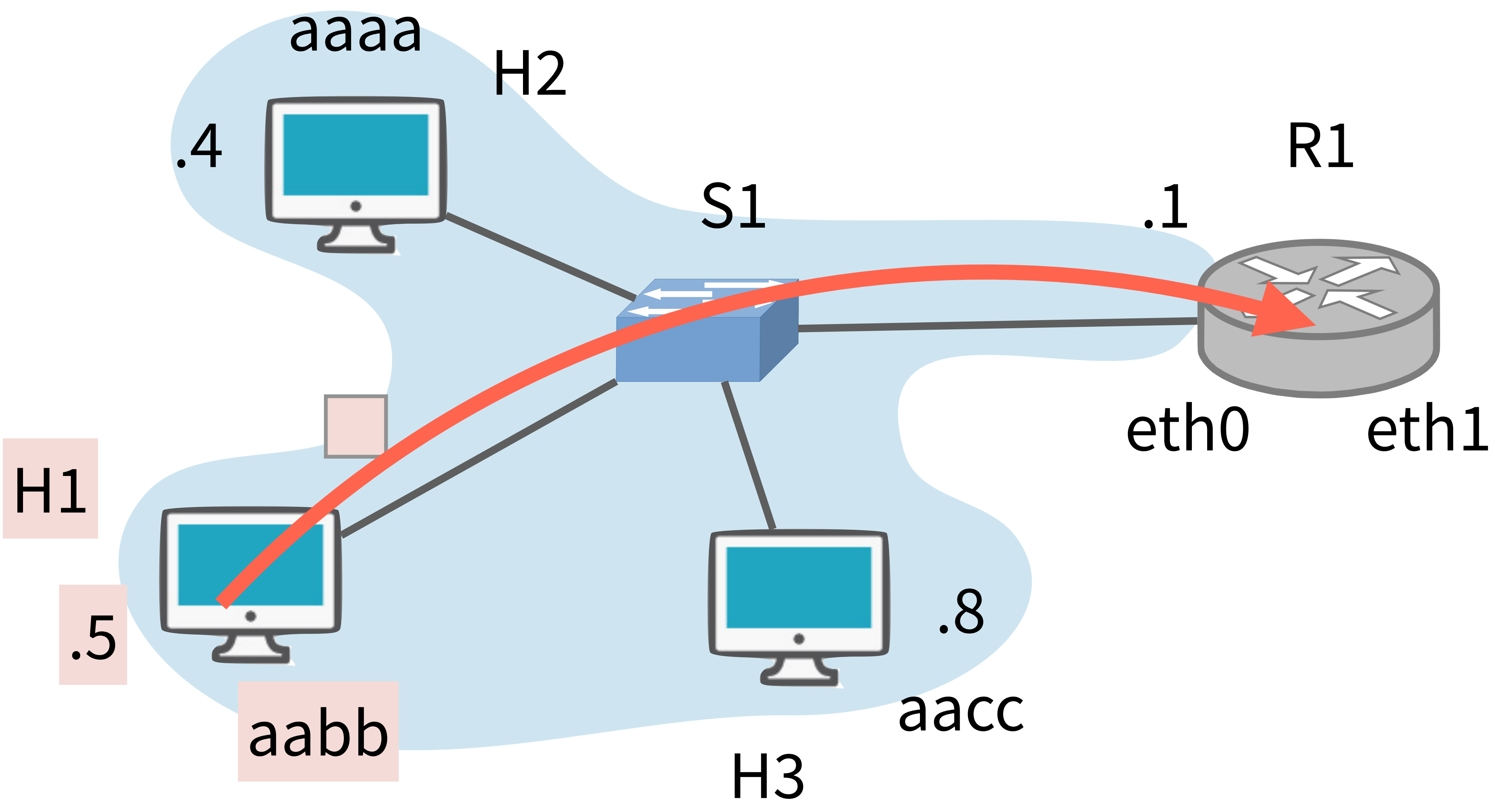
Routing between (IP) networks/subnets



Routing between (IP) networks/subnets



Journey of a packet



Network A

On H1

- Decide if the packet belongs to the same network by comparing the destination IP with its own IP on the masked IP bits
- If so, send a frame containing the packet to the destination IP with its MAC address
- If not, send a frame containing the packet to the default gateway

On S1

- Forward the frame to the Ethernet segment connecting eth0 of router R1

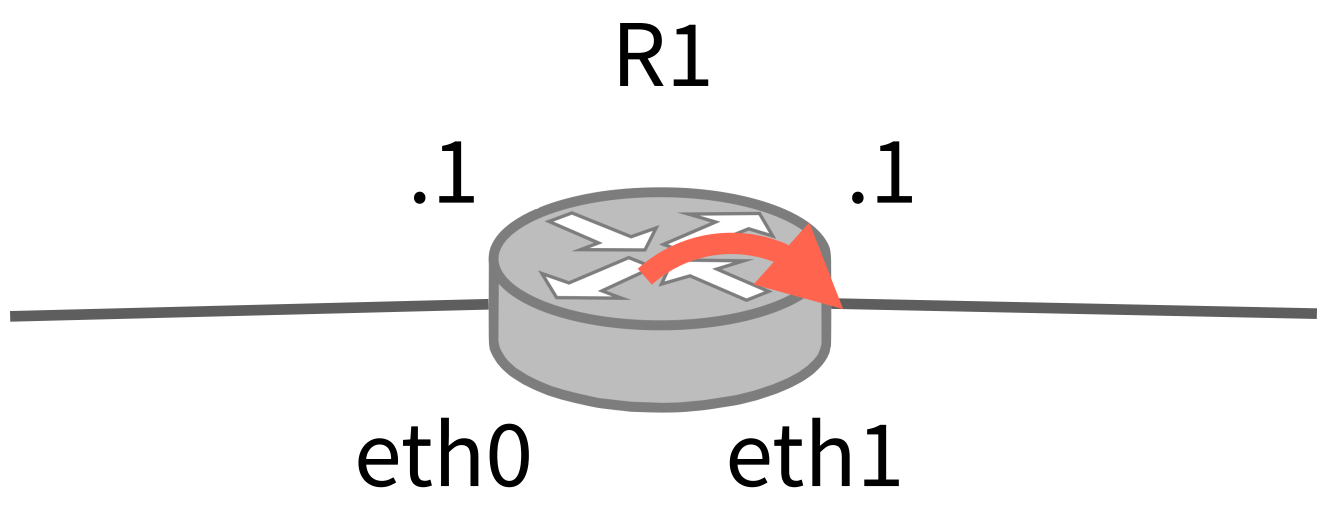
Journey of a packet

On R1: routing

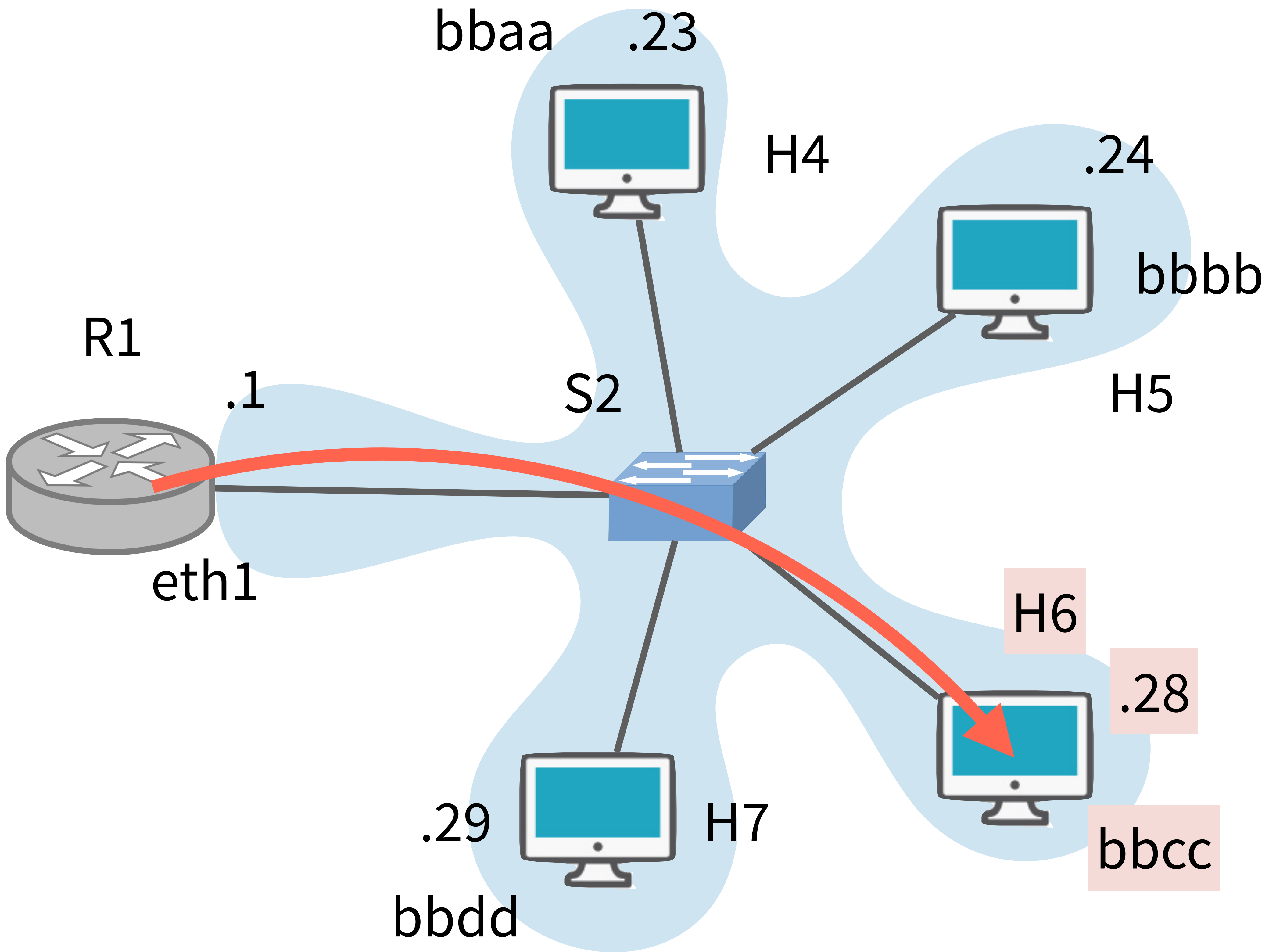
- Unpack the frame, get the IP packet
- Check the packet header checksum
- Look for the destination IP (longest prefix matching) in the **forwarding table**
- If found, forward the packet to the next hop – interface given by the forwarding table
- If not, packet will be forwarded to the default interface if specified, or dropped otherwise

Forwarding table

IP (prefix)	Next hop
22.22.22.0/24	eth2
11.11.11.0/24	eth1



Journey of a packet



Network B
(22.22.22.0/24)

On R1

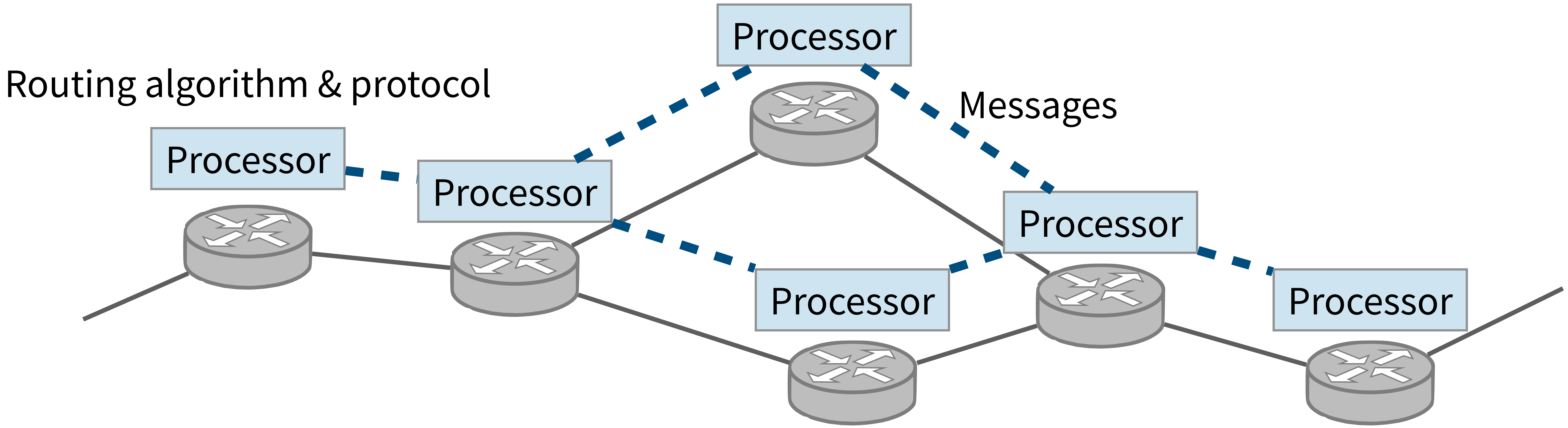
- Obtain the destination MAC of host H6
- Send a frame containing the packet with the MAC of H6 as destination MAC

On S2

- Forward the frame to the Ethernet segment connecting H6 based on the forwarding table maintained by S2

How to generate forwarding tables?

Control plane: modern routers employ a **distributed protocol** to exchange messages and compute shortest paths to other routers to generate the forwarding table: OSPF (link state), BGP (distance vector)



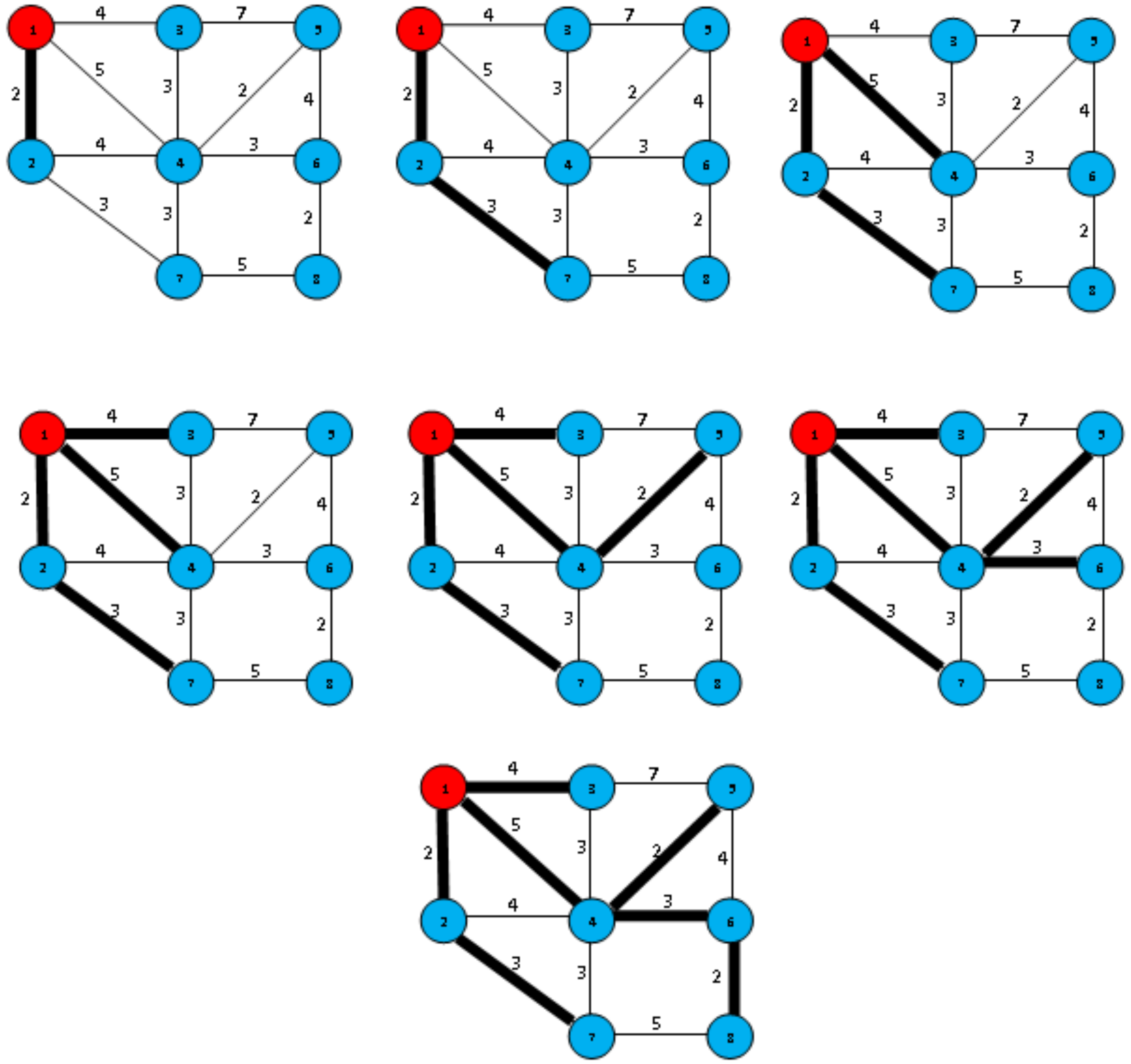
We will discuss a completely different way next week!

Routing protocol

Open Shortest Path First (OSPF):

- Routers exchange link-state messages to learn the topology
- Each router runs the **Dijkstra's algorithm** to computer the shortest paths to other routers
- Each router generates the forwarding table entries based on the shortest paths

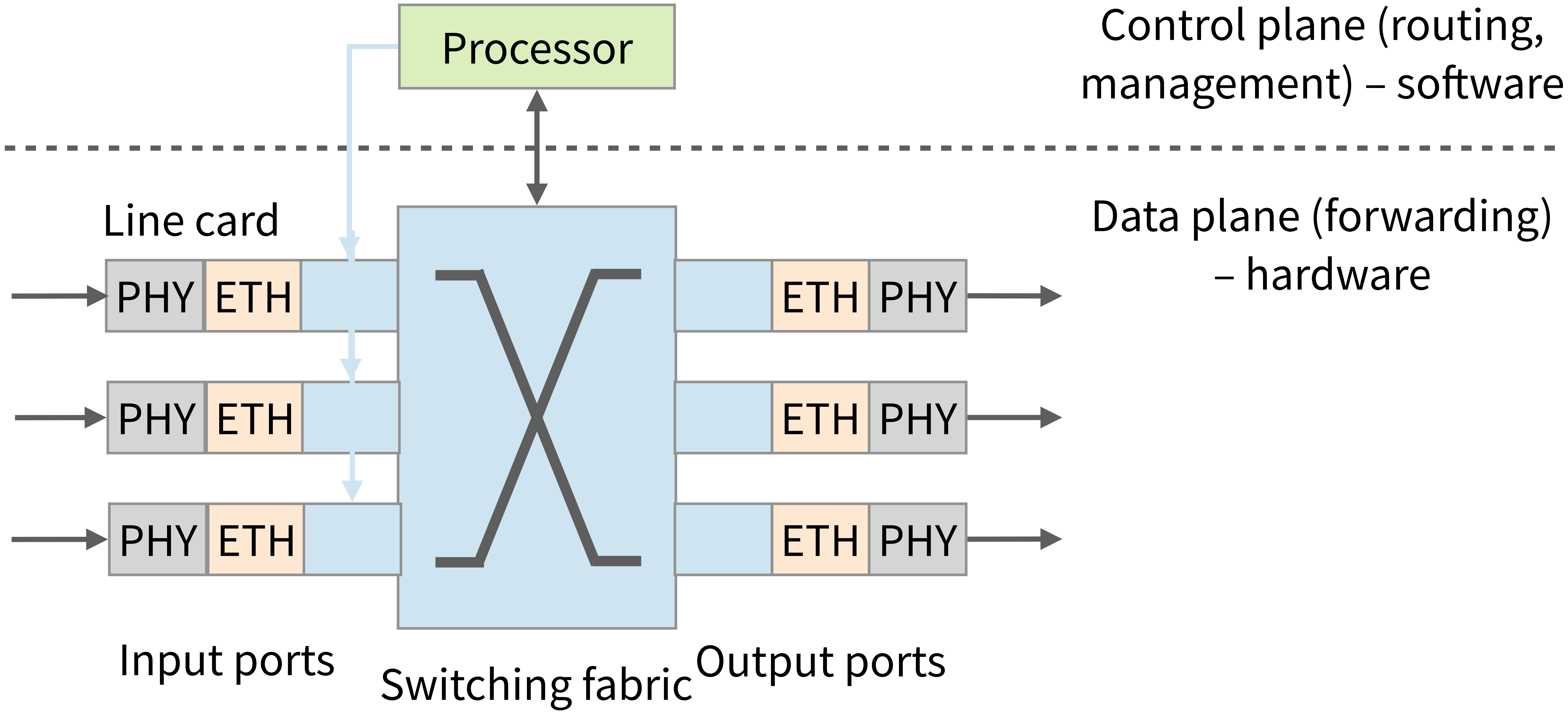
Border Gateway Protocol (BGP)



Router architecture

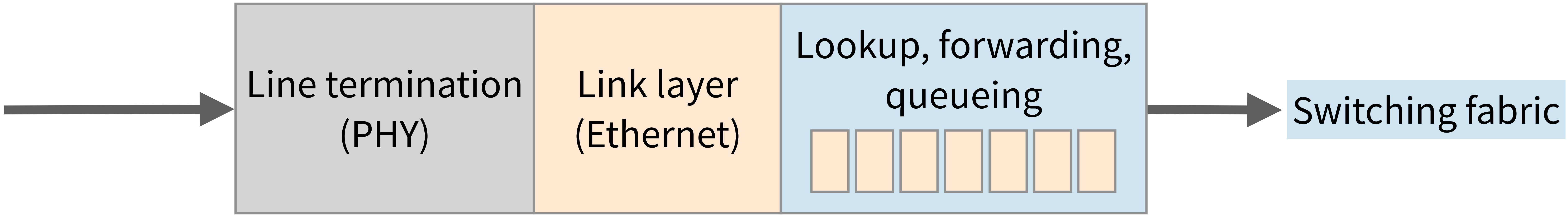
Two general functions:

- **Routing:** run routing protocols/algorithms (e.g., OSPF, BGP) to generate forwarding tables
- **Forwarding:** forwarding packets from incoming to outgoing links



Input port functions

Match+action is a very powerful abstraction in computer networking: firewall, NAT, and more in coming lectures!



Decentralized switching:

- **Match + action:** given packet destination IP, look up output port using the forwarding table in the fast input port memory (e.g., TCAM) at line rate
- Queueing: if packets arrive faster than the forwarding rate of the switching fabric, buffer the packet
- Other actions: (1) check version number, checksum, TTL, (2) update checksum, TTL, (3) update monitoring counter

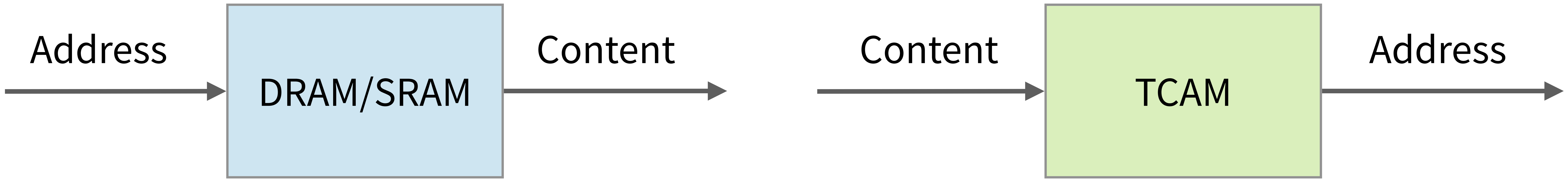
IP lookup

Match and action:

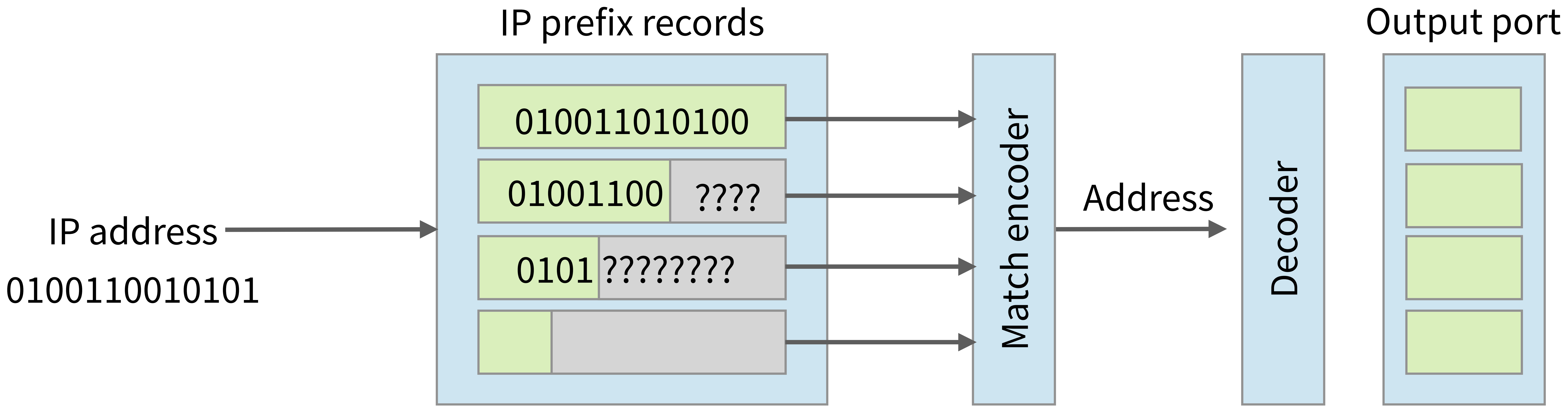
- Match on **IP prefix**: longest prefix matching rule
- Based on the match results, take an action: **forward** to an output port, **drop**, **replicate**, etc.

How to achieve **high matching performance**?

- Software implementation (binary search) with SRAM is not fast enough to achieve line rate: consider 10Gbps link with 64-byte IP packet, only <51.2ns to process a packet (assuming one port per line card)
- Use special hardware: Ternary Content Addressable Memory (TCAM) for IP prefix matching



TCAM



TCAM is a hardware device that supports to match on a set of records in constant time (one iteration)

- CAM supports only two states (0/1) in each bit position: widely used in switches for MAC address matching
- TCAM extends CAM by allowing for 3 states (0/1/?) in each position: useful for IP prefix matching
- Disadvantages: expensive, power-consuming

Switching fabric

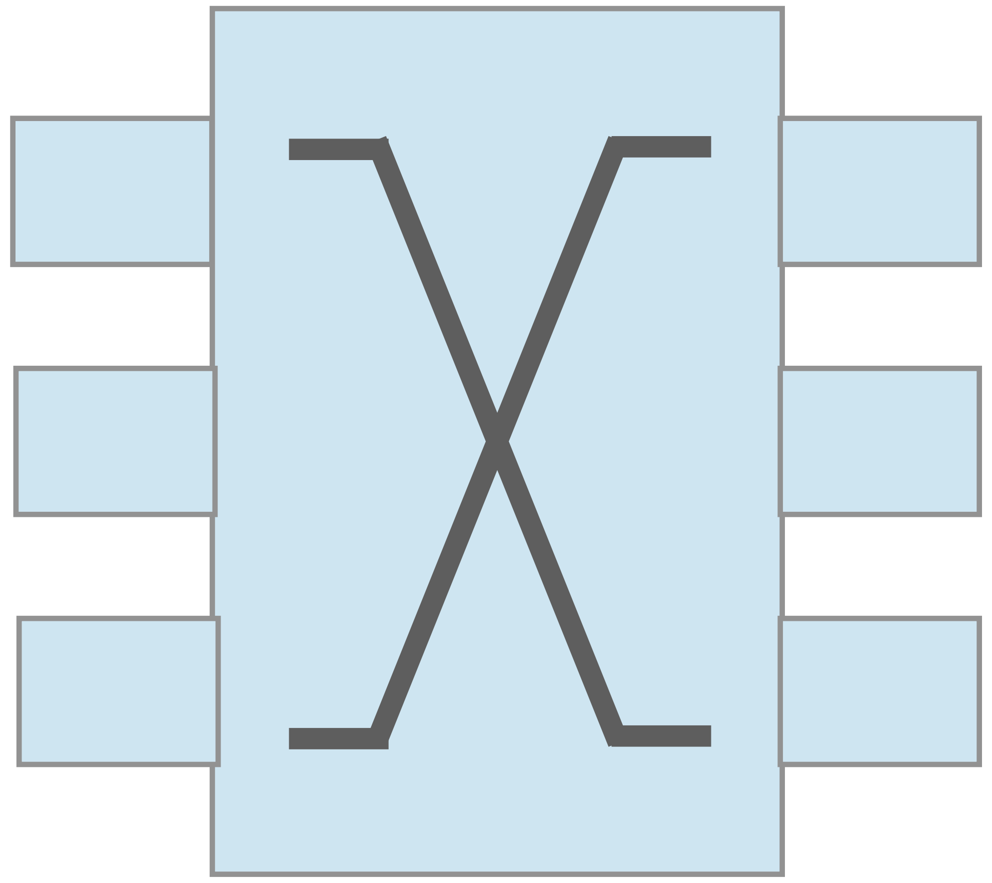
Transfer packet from input port to appropriate output port

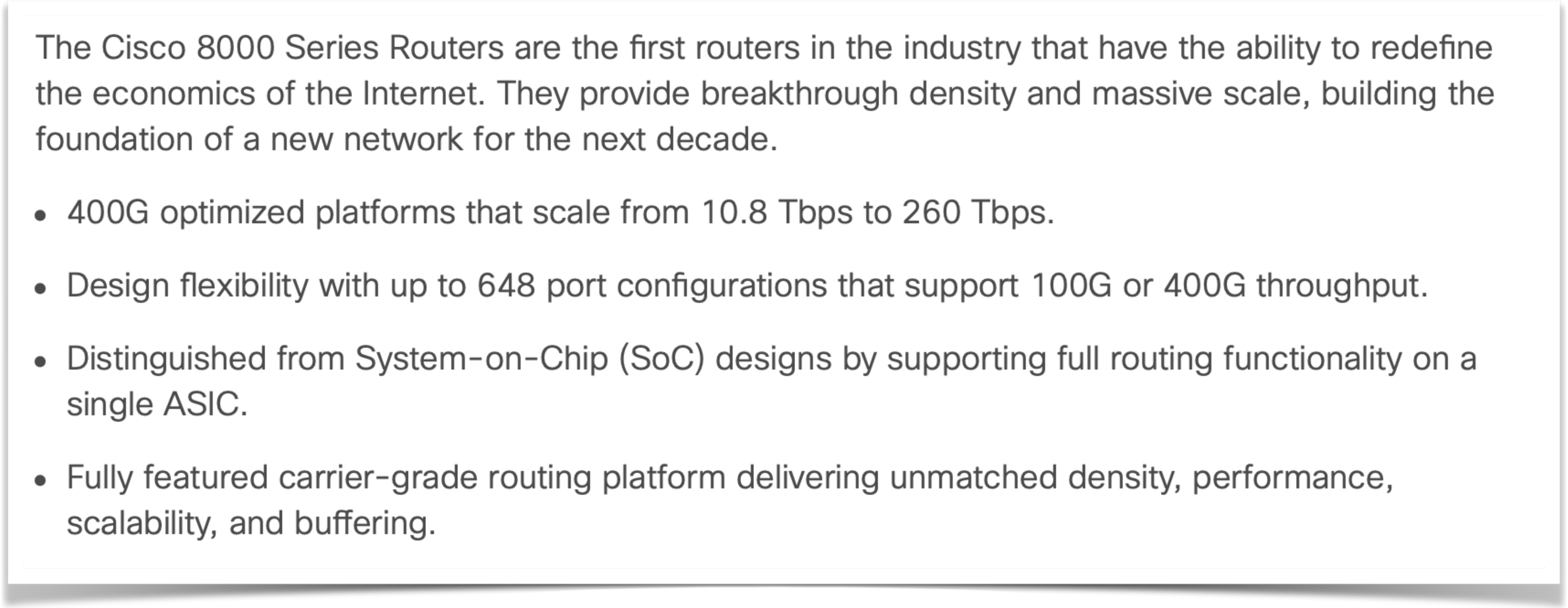
Switching rate: rate at which packets can be transferred from inputs to outputs

- Often measured as multiple of input/output line rate
- *N* inputs: switching rate *N* times line rate desirable

Generally three types of switching fabrics

- Via memory
- Via bus
- Via interconnection network (e.g., crossbar, multistage network)



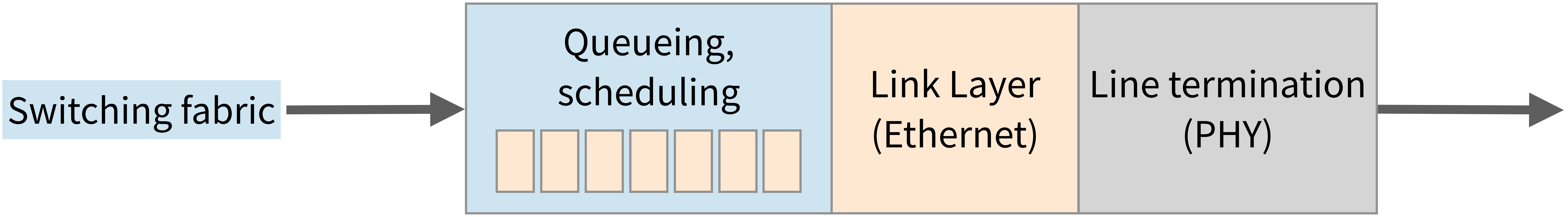


The Cisco 8000 Series Routers are the first routers in the industry that have the ability to redefine the economics of the Internet. They provide breakthrough density and massive scale, building the foundation of a new network for the next decade.

- 400G optimized platforms that scale from 10.8 Tbps to 260 Tbps.
- Design flexibility with up to 648 port configurations that support 100G or 400G throughput.
- Distinguished from System-on-Chip (SoC) designs by supporting full routing functionality on a single ASIC.
- Fully featured carrier-grade routing platform delivering unmatched density, performance, scalability, and buffering.

https://www.cisco.com/c/en/us/products/routers/8000-series-routers/index.html

Output port functions



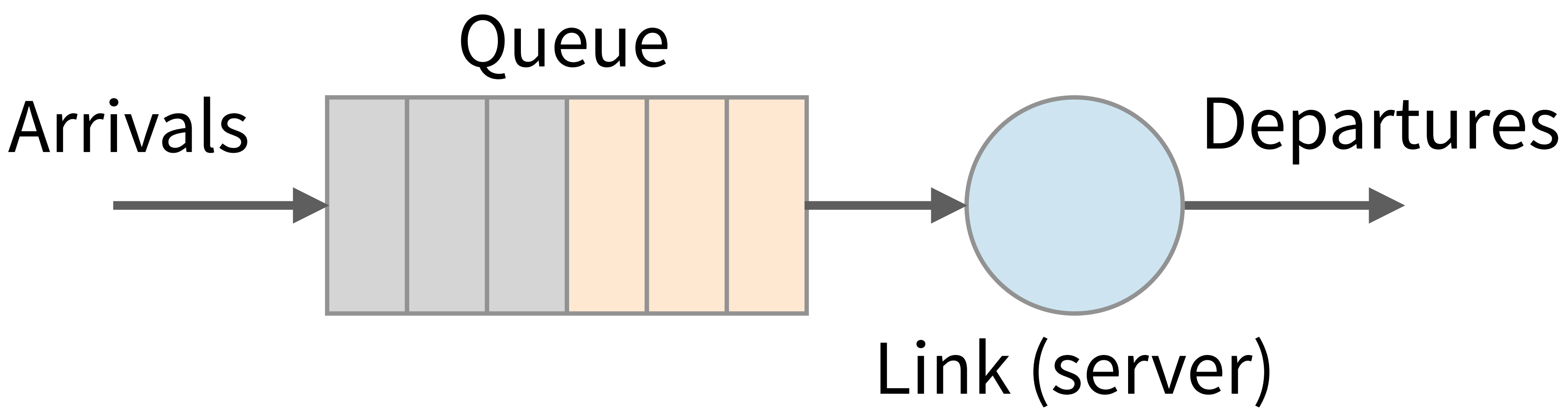
Queueing: required to handle the case where the speed packets depart from the switching fabric is faster than the transmission rate

- What happens if the queue is full? Packet will be dropped, unless **active queue management (AQM)** mechanisms (like random early detection, RED) are enabled
- What should the queue size be? Rule of thumb *B* = *RTT* x *C*

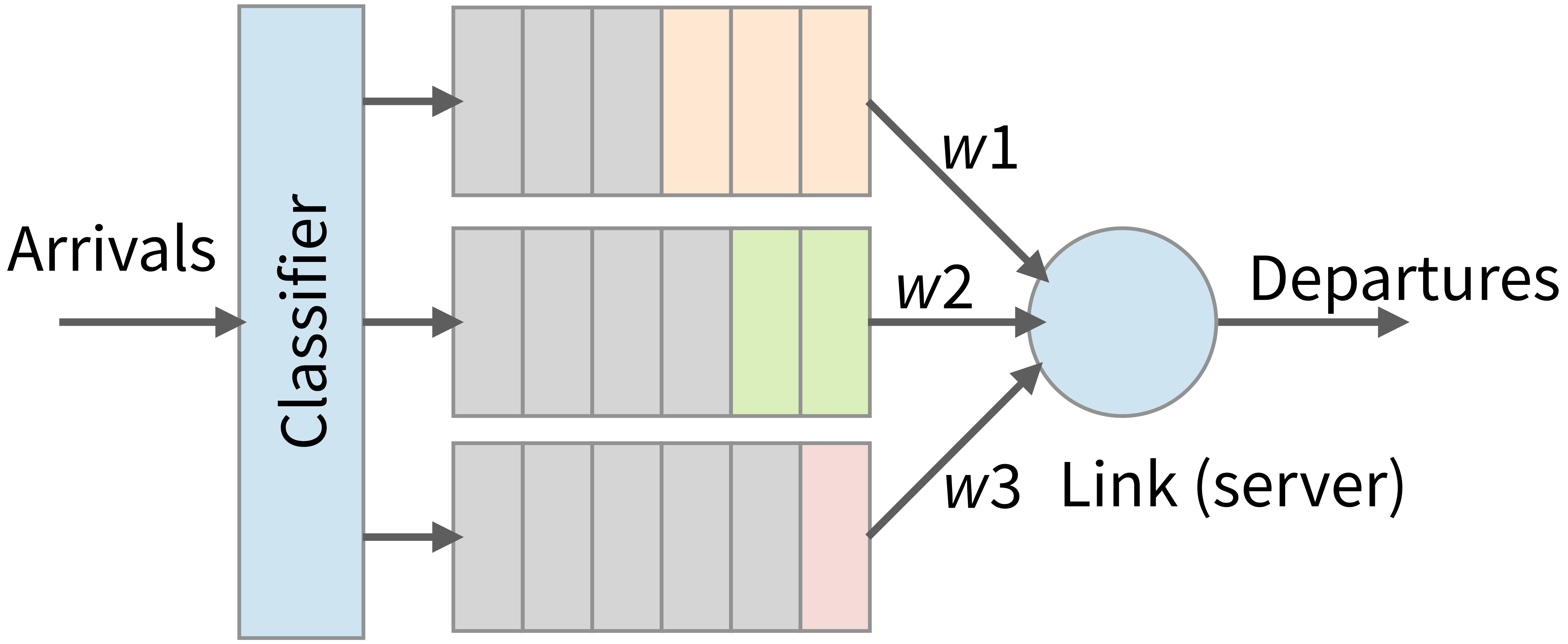
RFC 3439

Scheduling: decide which packet to go first on the wire

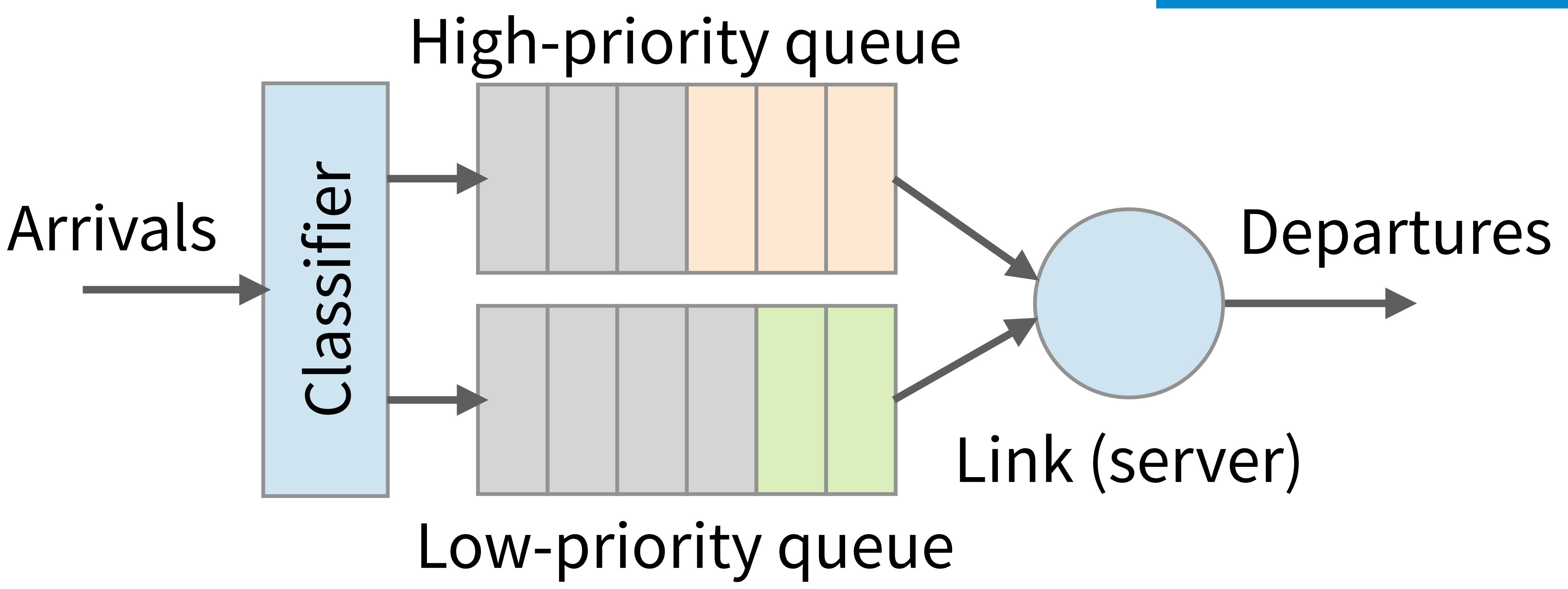
Packet scheduling policies



FIFO queueing model



Weighted fair queueing model



Priority (non-preemptive) queueing model

Summary

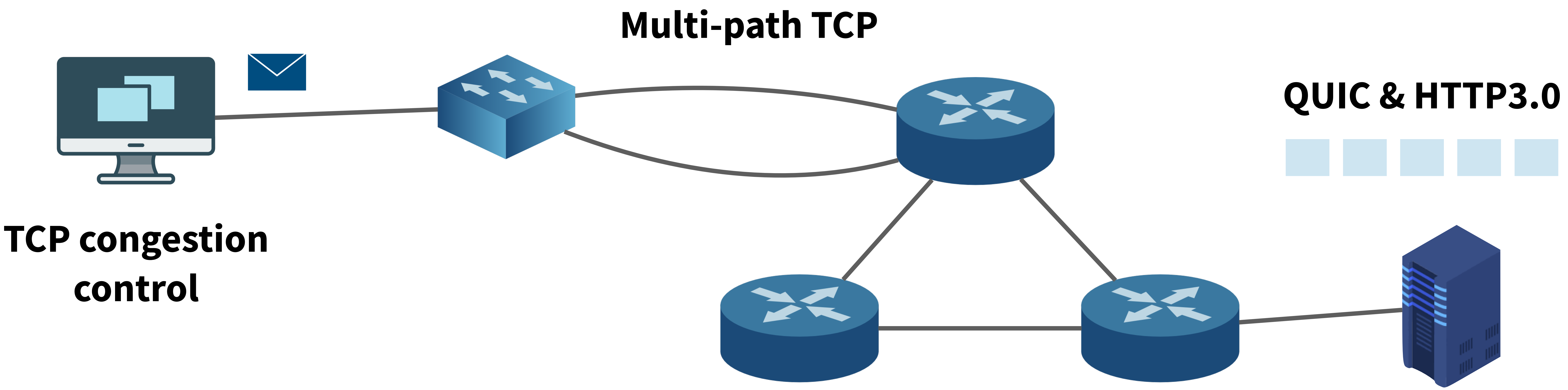
Key networking concepts revival

- **DNS:** domain name server, translates a URL to an IP address
- **Socket:** communication endpoints, provides OS abstractions for network functionalities
- **Switching:** a link-layer functionality to move packets in a local area network
- **ARP:** a link-layer protocol which resolves the IP address to MAC address
- **Routing:** a network-layer functionality which routes a packet across different (sub)networks

Router architecture

- IP lookup, TCAM
- Switching fabric
- Input/output, buffer, scheduling

Next time: network transport



Multi-path TCP

QUIC & HTTP3.0

TCP congestion control