Advanced Networked Systems SS24

Networking Fundamentals

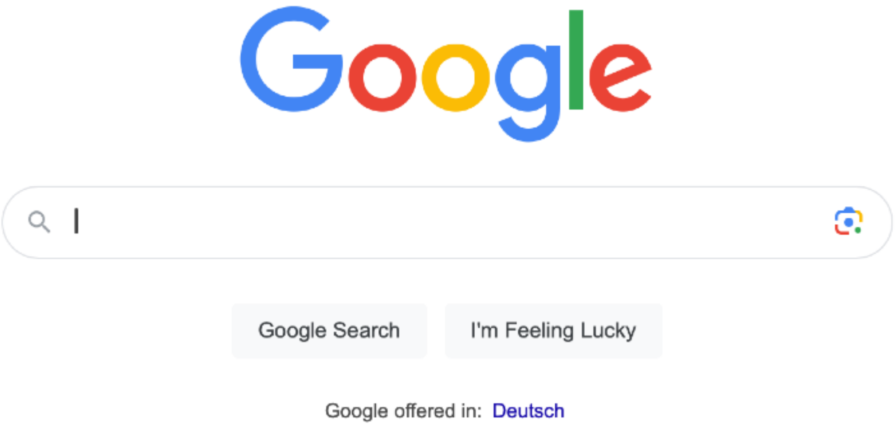
Prof. Lin Wang, Ph.D.

Computer Networks Group

Paderborn University

https://en.cs.uni-paderborn.de/cn

Learning objectives



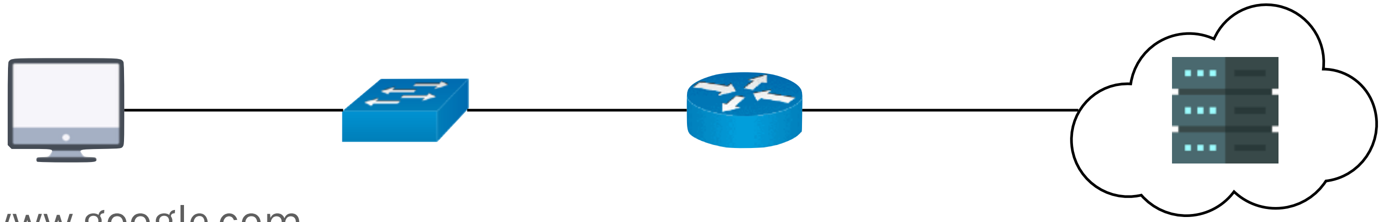
Google Search

I'm Feeling Lucky

Google offered in: Deutsch

What happens under the hood when you visit https://www.google.com?

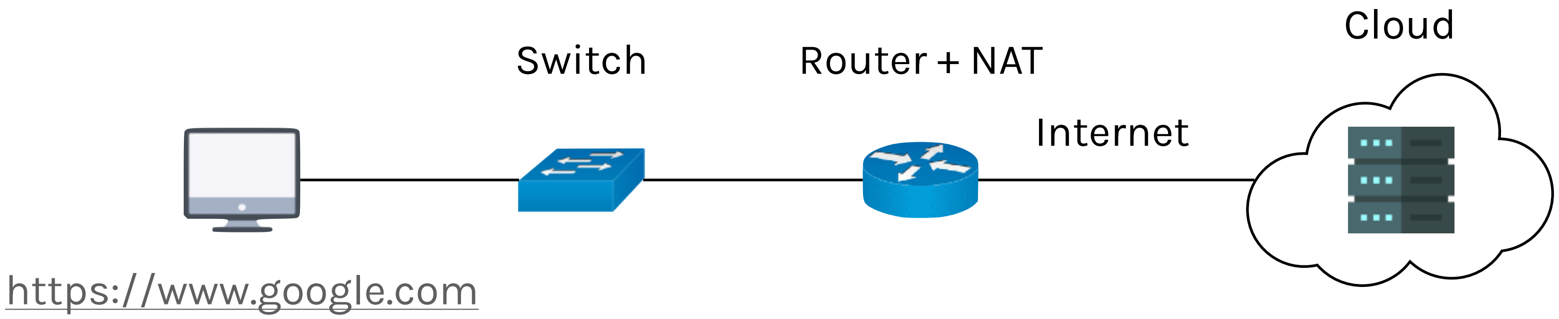
A simplified networking scenario



https://www.google.com

What networking concepts are involved?

A simplified networking scenario



Key networking concepts: DNS, Socket, TCP, IP routing, Ethernet, ARP, NAT

Domain Name System (DNS)

Domain name system (DNS)

If you want to mail someone

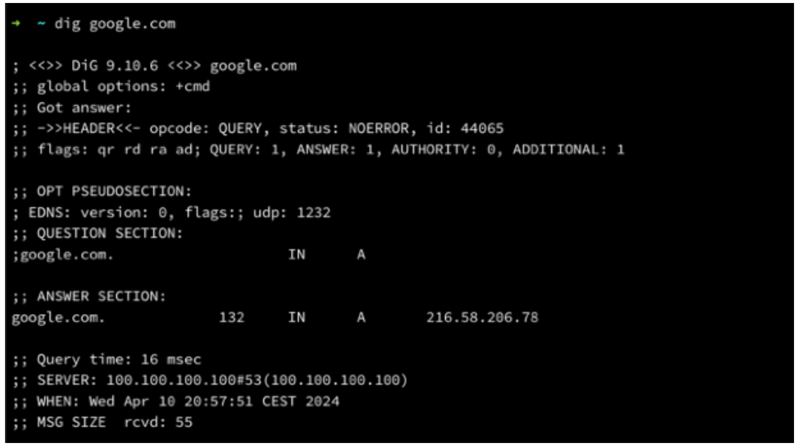
- You need to get their address first

What about the Internet?

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

Problem

- People are bad at remembering IP addresses
- Need human readable names that map to IPs

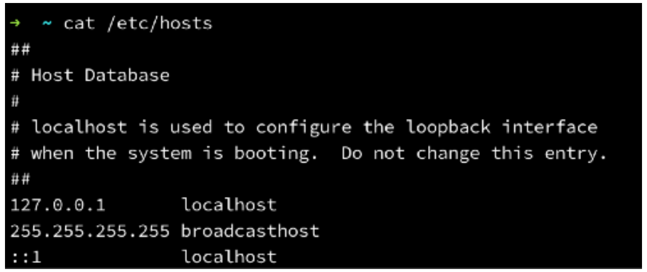


```
➜  ~ dig google.com

; <<>> DiG 9.10.6 <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 44065
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;google.com.                    IN      A

;; ANSWER SECTION:
google.com.             132     IN      A       216.58.206.78

;; Query time: 16 msec
;; SERVER: 100.100.100.100#53(100.100.100.100)
;; WHEN: Wed Apr 10 20:57:51 CEST 2024
;; 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_upb

Not scalable

Hard to enforce uniqueness

Consistency issue

DNS overview

Distributed database

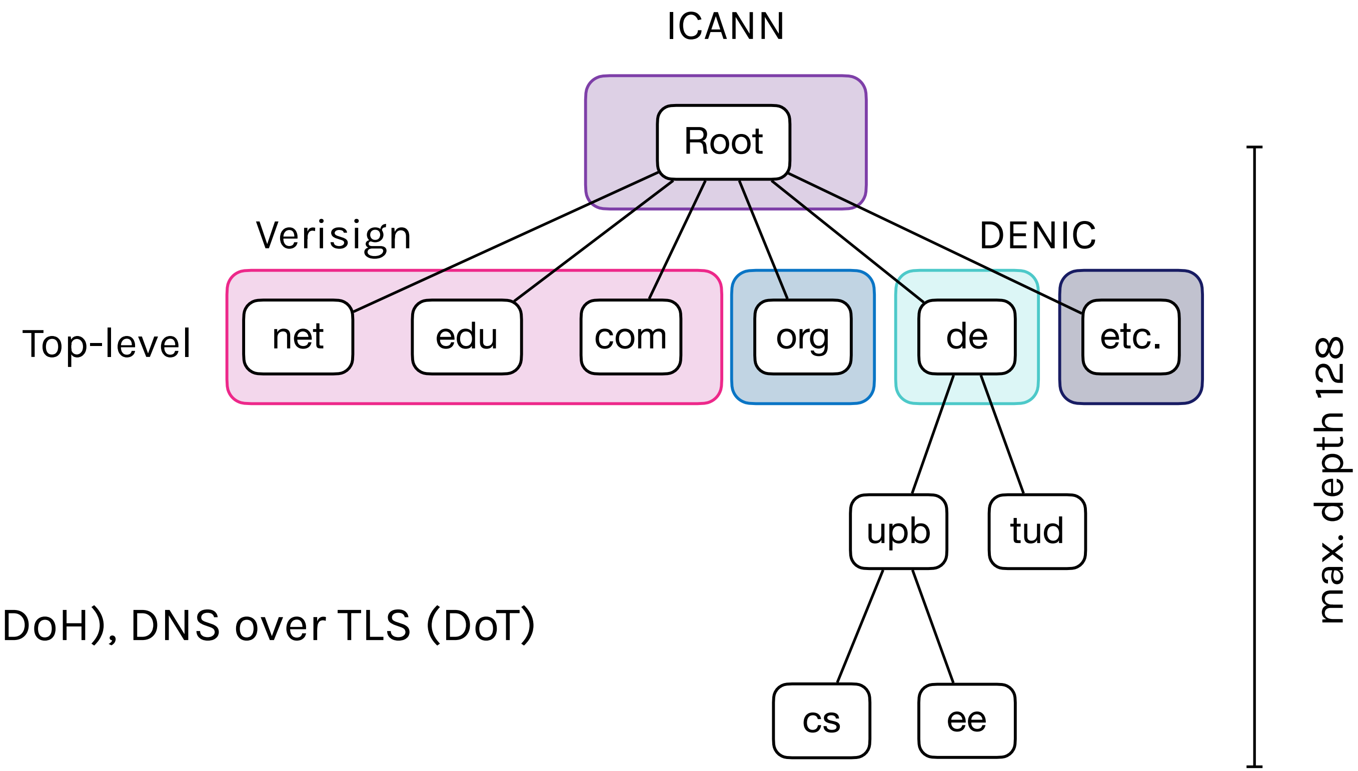
- No centralization → scalability

Simple client/server architecture

- UDP port 53
- Some use TCP: DNS over HTTPS (DoH), DNS over TLS (DoT)

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

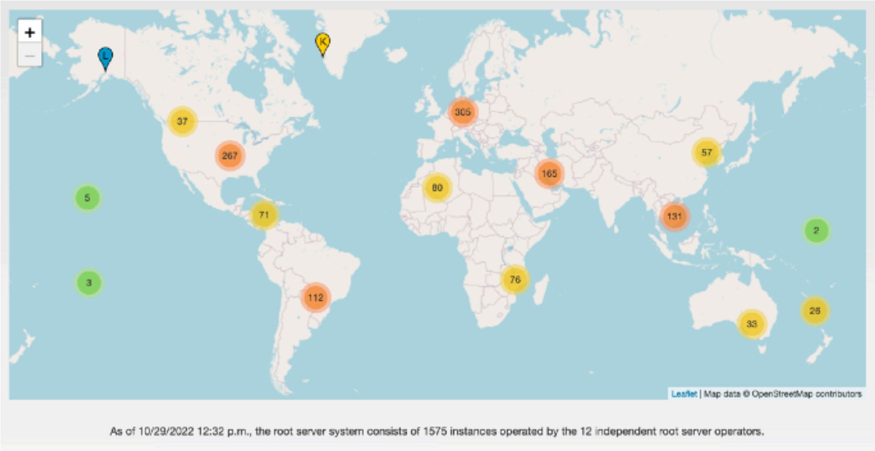
- Lists the top-level domains (TLDs) and who controls them, ~2MB file size

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



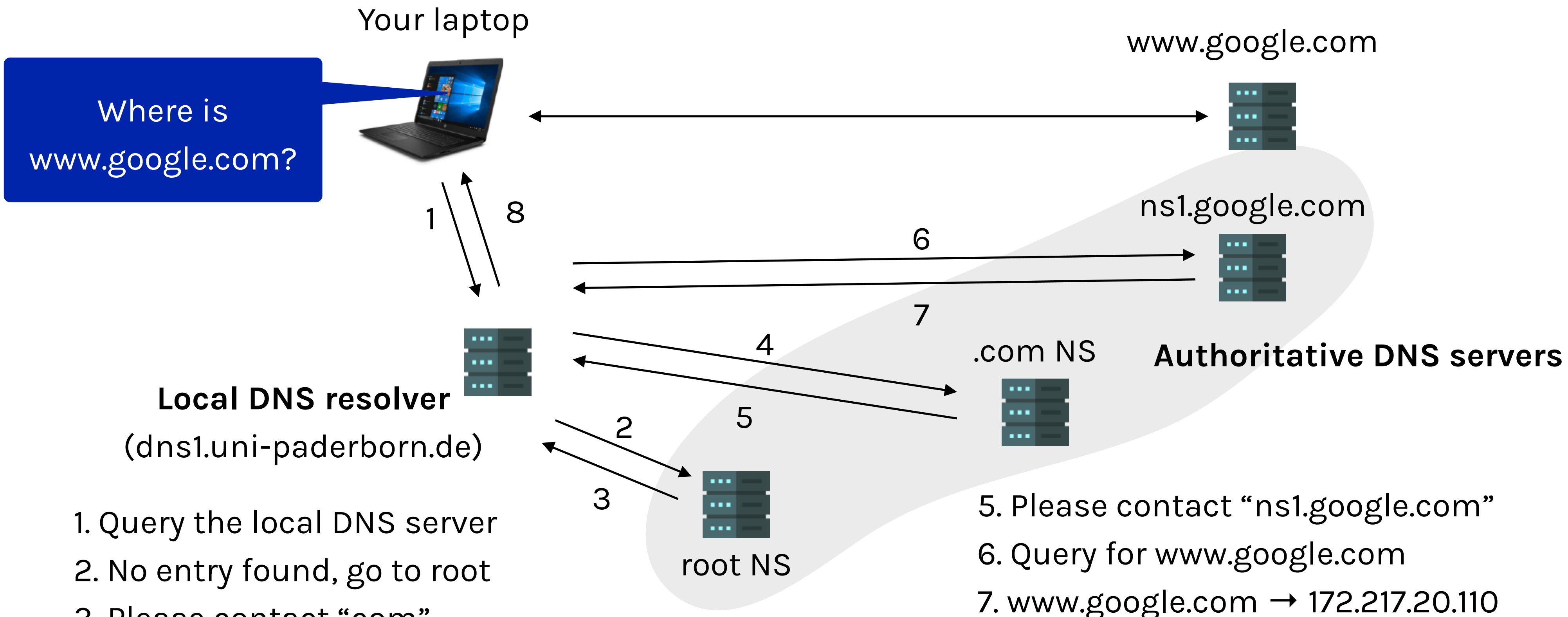
As of 10/29/2022 12:32 p.m., the root server system consists of 1575 instances operated by the 12 independent root server operators.

https://root-servers.org

How does a URL get resolved to an IP address?

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: `cs.upb.de`

Type: A (or AAAA)

Resp.

Name: `cs.upb.de`

Value: `130.37.164.171`

DNS resolution (AAAA for IPv6)

Query

Name: `cs.upb.de`

Type: CNAME

Resp.

Name: `cs.upb.de`

Value: `cs.uni-paderborn.de.`

Look for alias (canonical hostname)

Query

Name: `cs.upb.de`

Type: NS

Resp.

Name: `cs.upb.de`

Value: `131.234.9.34`

Query for DNS server
responsible for the partial name

Query

Name: `cs.upb.de`

Type: MX

Resp.

Name: `cs.upb.de`

Value: `mail.cs.upb.de`

Look for the mail server

DNS security

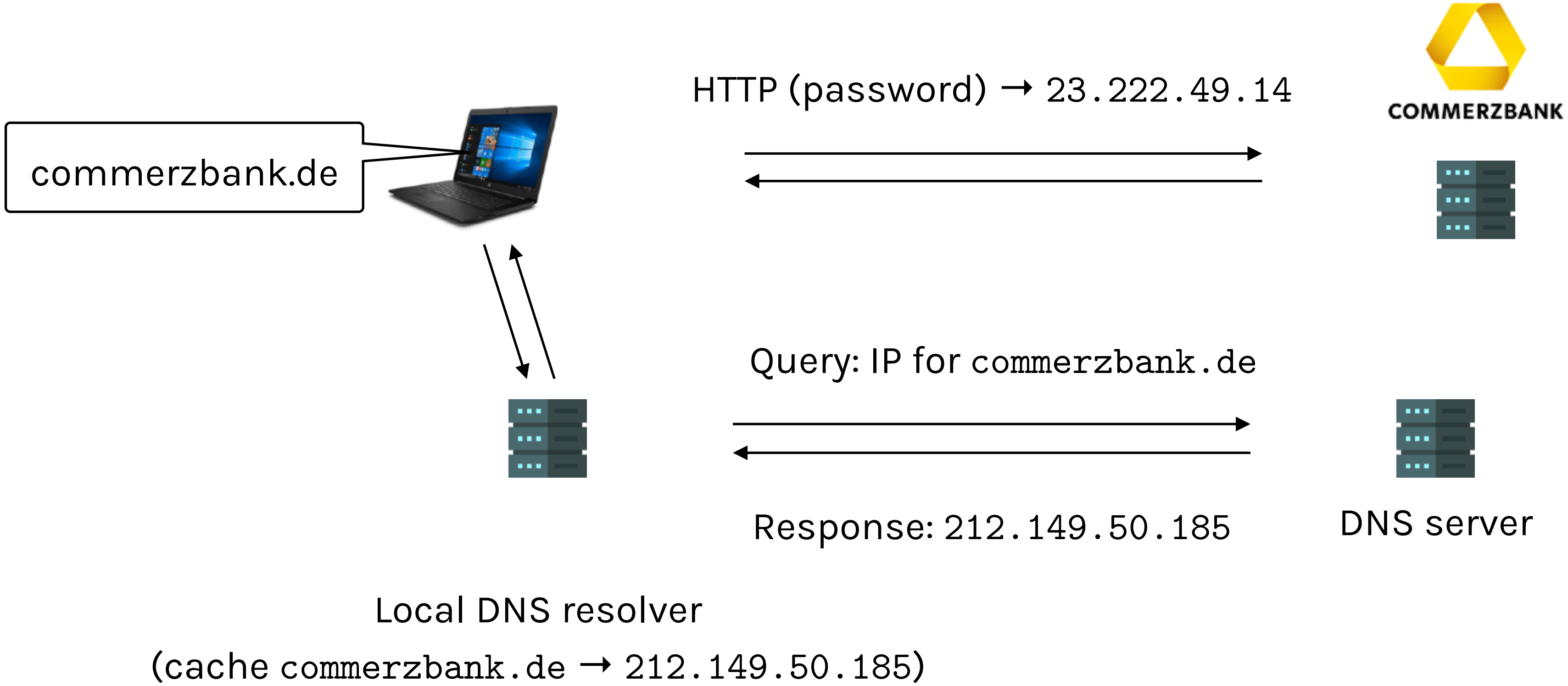
You are your mail server

- 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 `commerzbank.de`, they expect to be taken to their bank's website
- What if the DNS record is compromised?

DNS security



commerzbank.de

HTTP (password) → `23.222.49.14`

COMMERZBANK

Query: IP for `commerzbank.de`

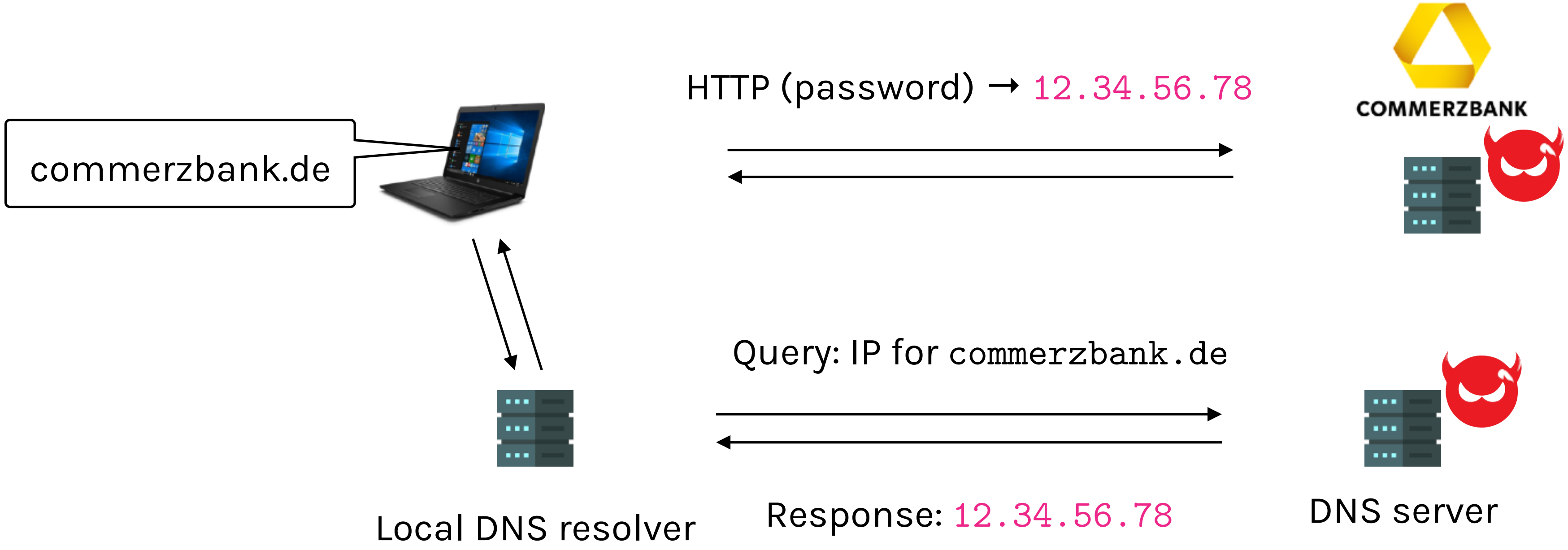
Response: `212.149.50.185`

DNS server

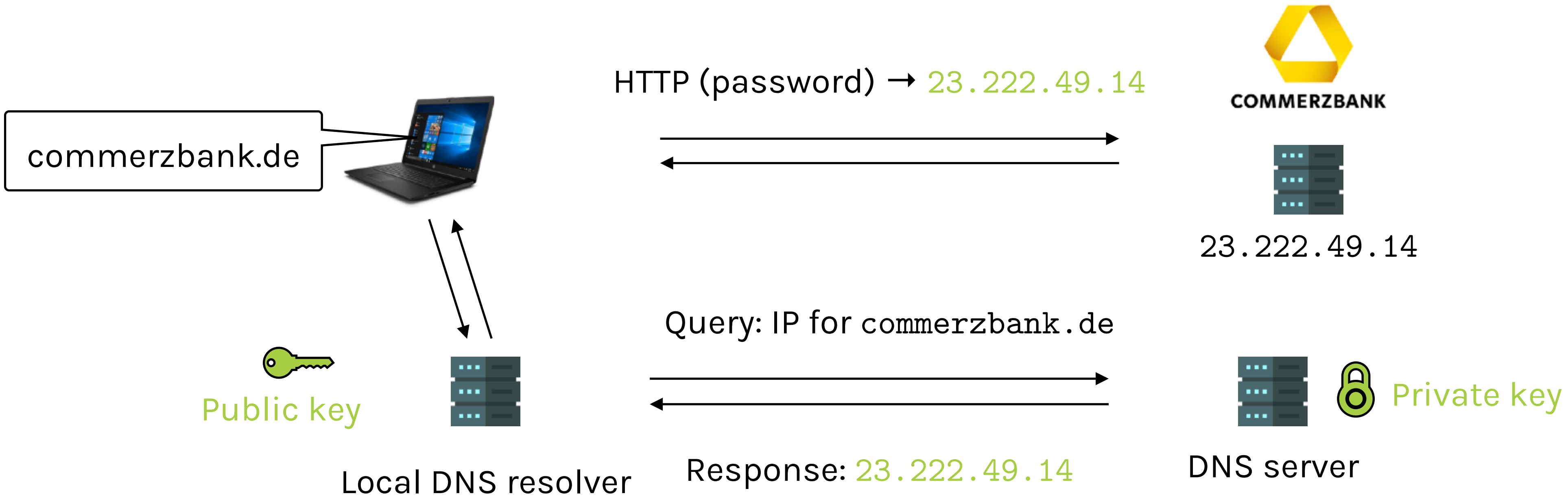
Local DNS resolver

(cache `commerzbank.de` → `212.149.50.185`)

DNS security



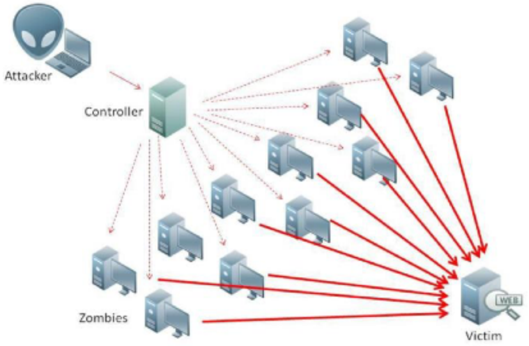
DNS security



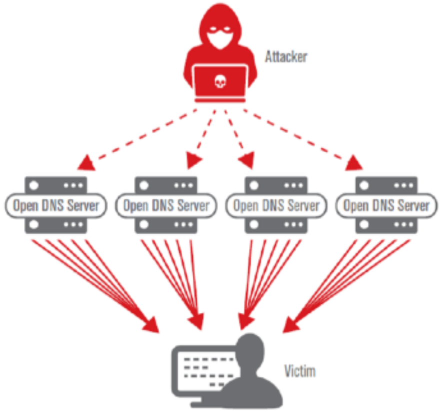
DNSSEC: **data origin authentication** and **data integrity protection**

How to make sure the public key is authentic? What about privacy?

DNS security



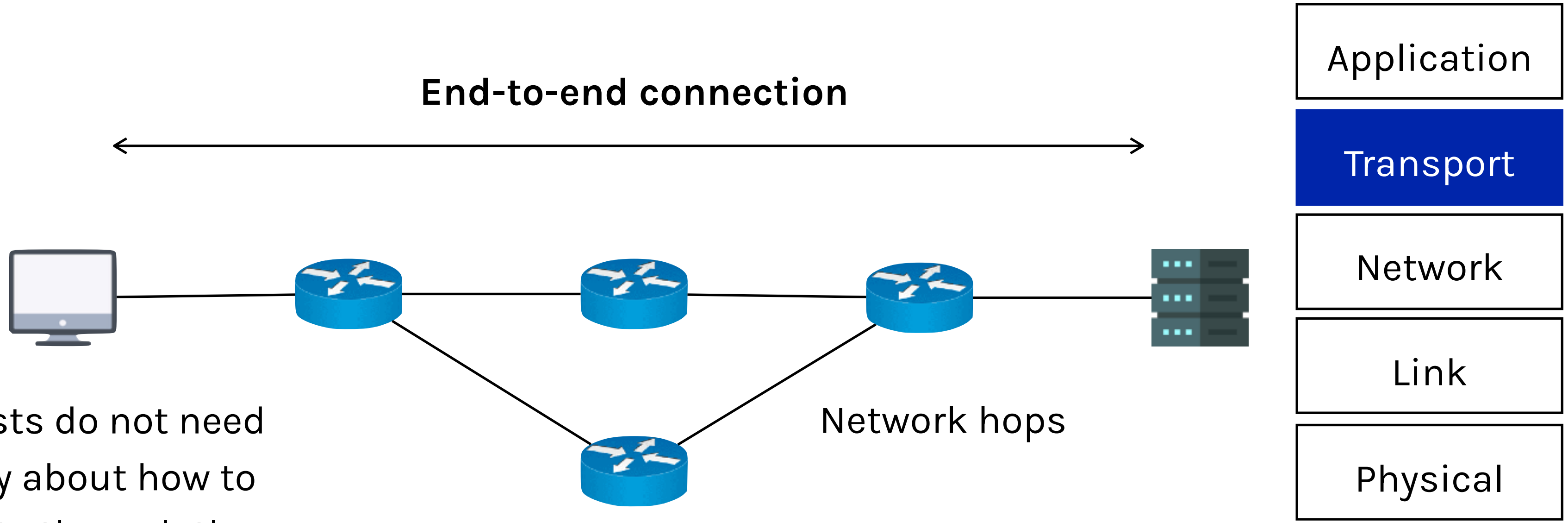
Distributed Denial of Service (DDoS)



DNS amplification attack

Socket and TCP

The transport layer

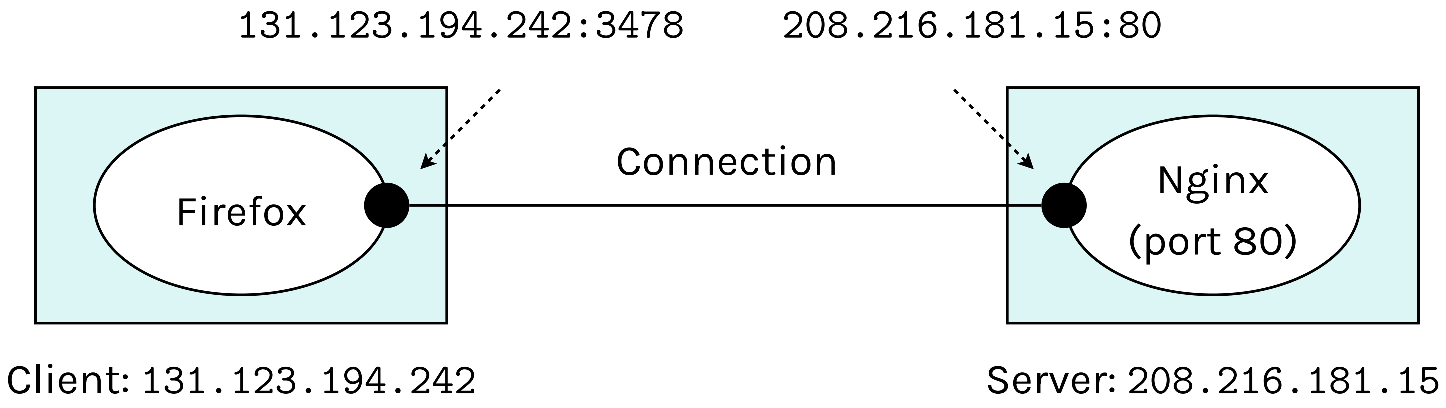


End-hosts do not need to worry about how to navigate through the network hops

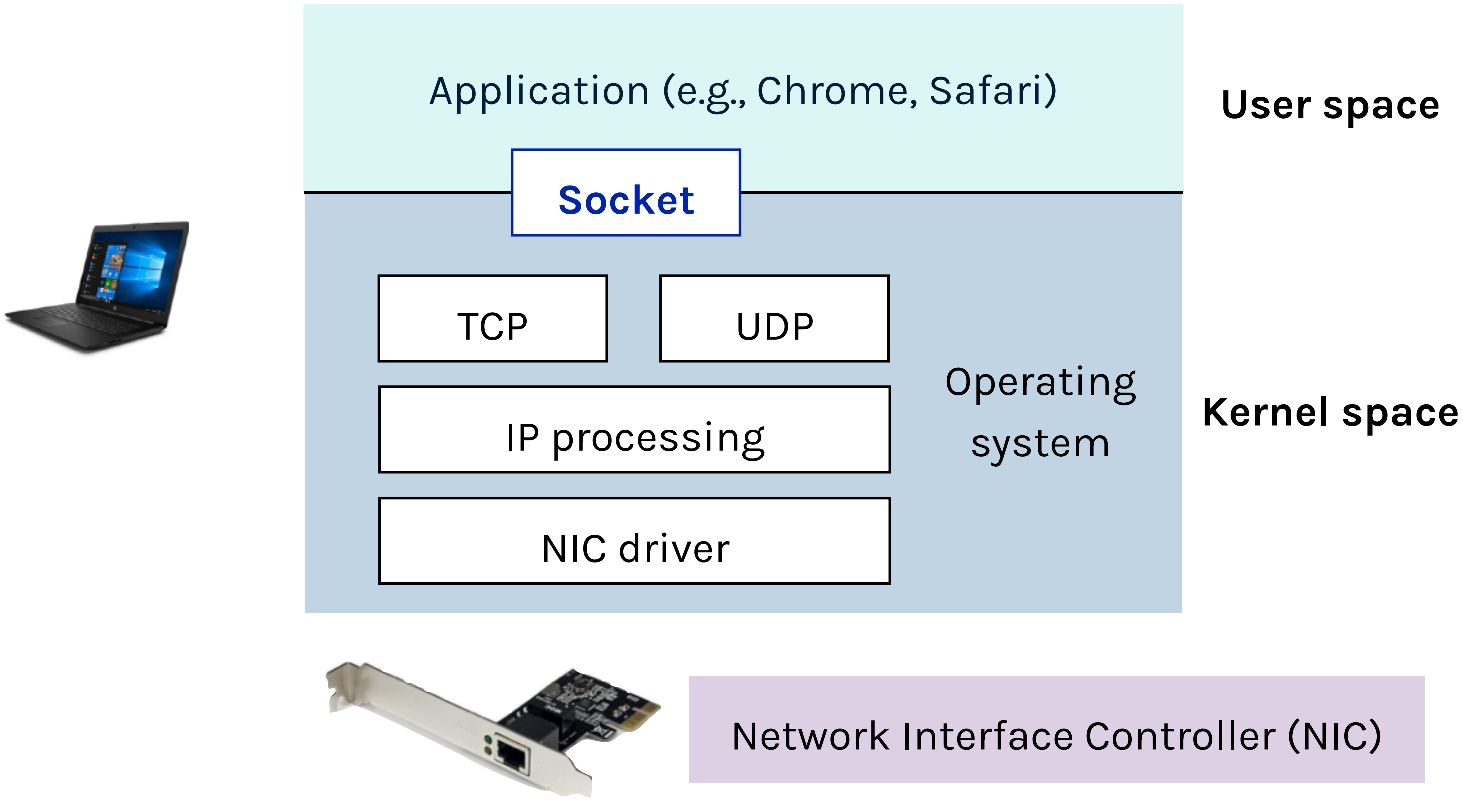
Network connections

Inter-process communication (IPC)

- Address the machine on the network: by IP address
- Address the process on the machine: by the port number
- The pair of IP:port makes up a socket address
- Socket is an endpoint of communication to the OS kernel

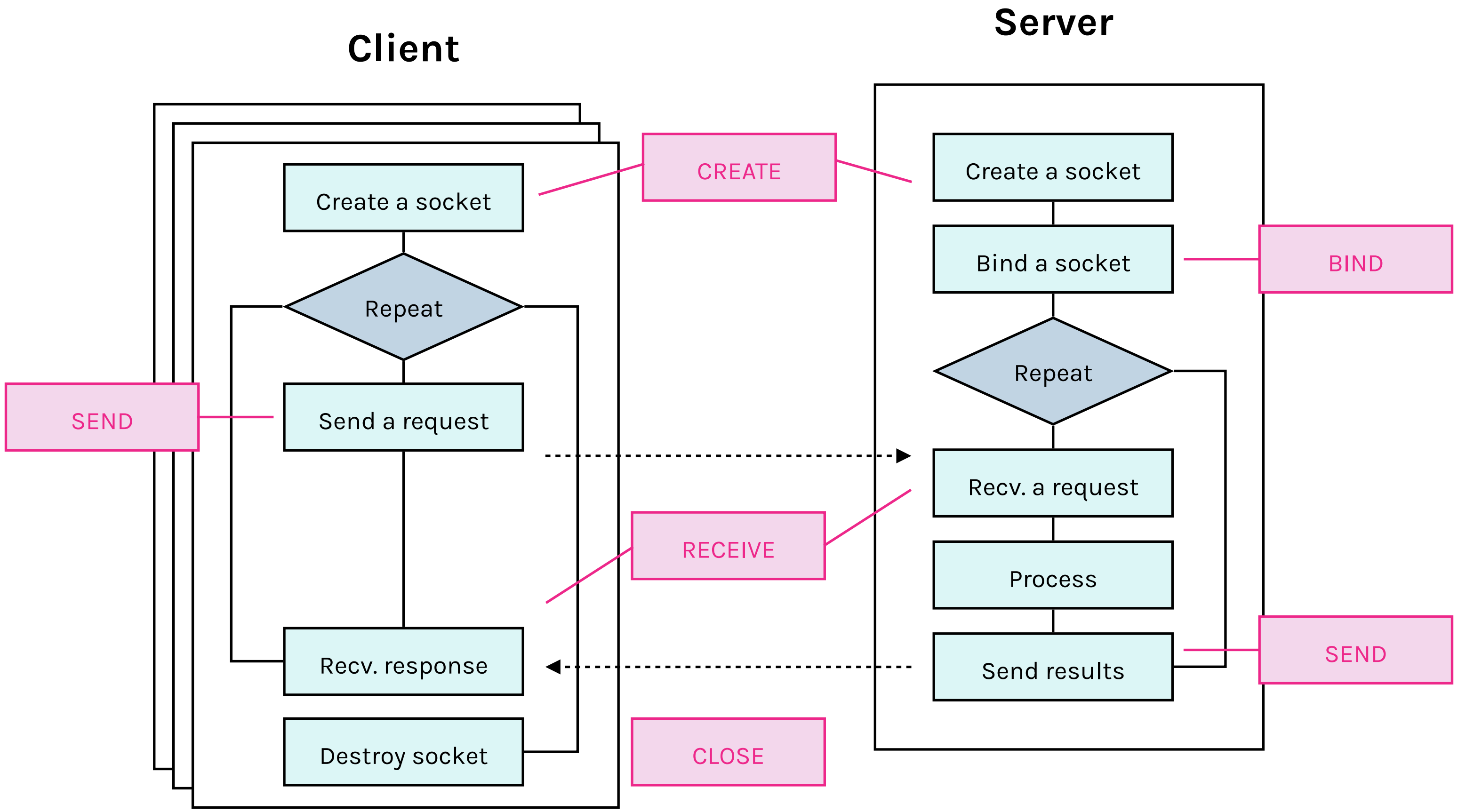


Making a connection through the socket interface

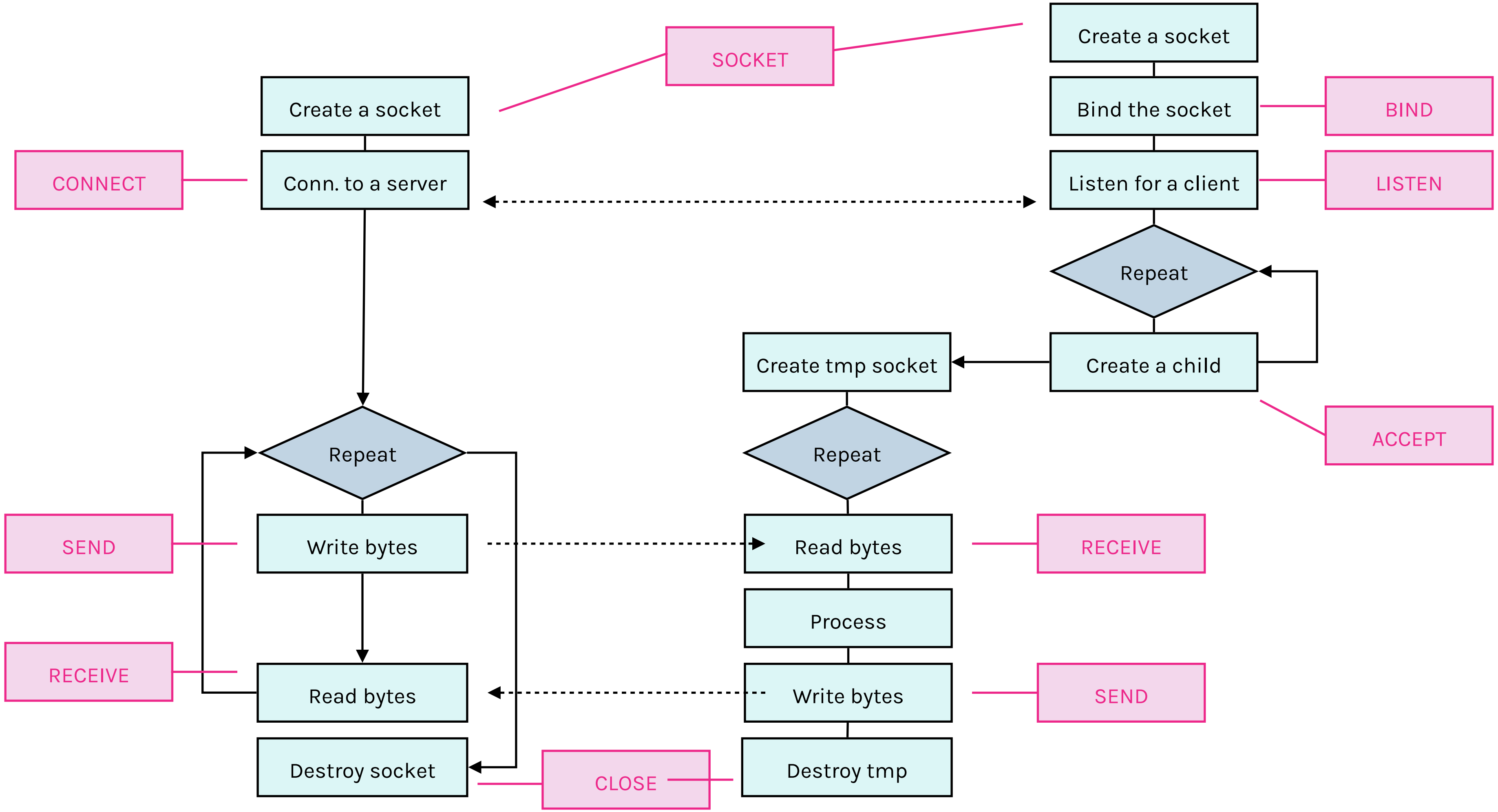


Socket represents the **communication endpoint**. It is an abstraction for user applications to access network functionalities implemented in the OS kernel.

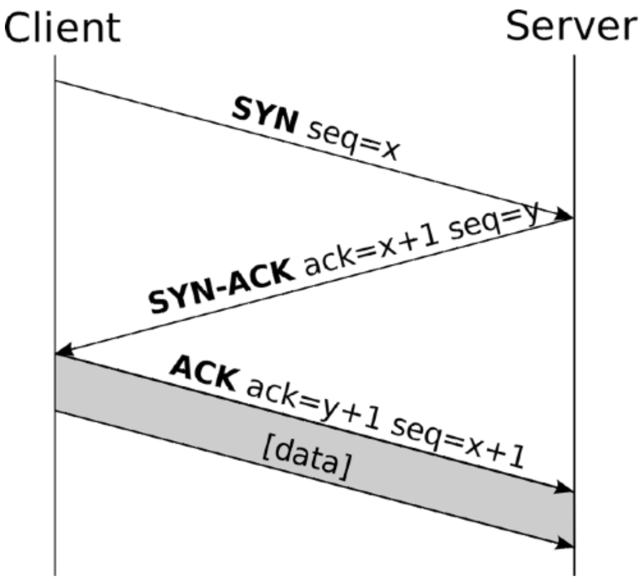
POSIX sockets: connectionless



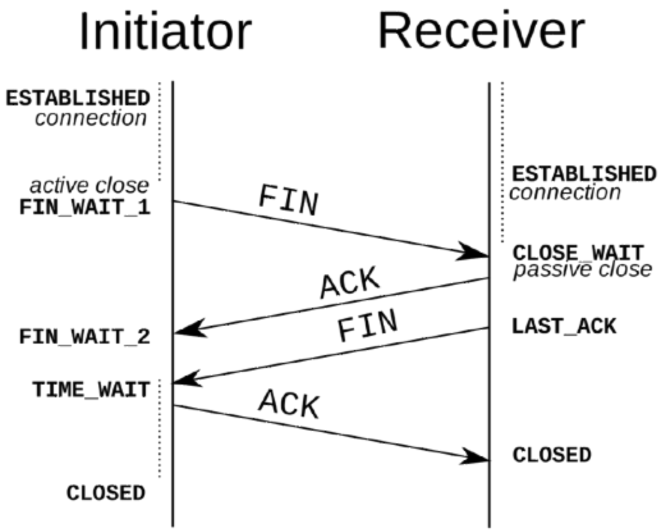
POSIX sockets: connection-oriented



TCP connect() and close()



TCP connection establishment



TCP connection termination

What promises does TCP provide?

TCP promises

Reliable delivery

- Integrity check
- Packet retransmission upon losses
- Packet reordering

Flow and congestion control

- Flow control: the receiver is not overrun by the sender
- Congestion control: the network is not overrun by the sender

How are these functionalities achieved by TCP?

TCP promises

Reliable delivery

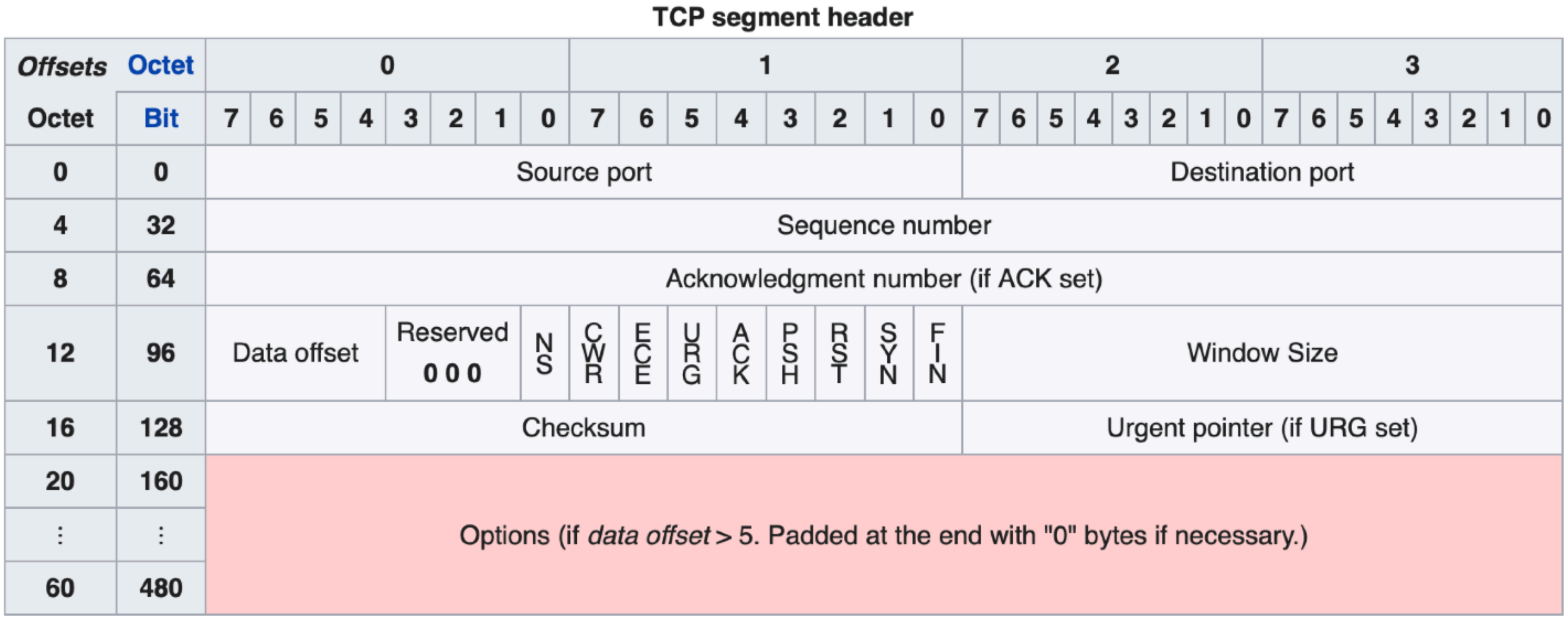
- Integrity check with checksum
- Packet retransmission upon losses with sequence number, timer, and sender buffer
- Packet reordering with sequence number and receiver buffer

Flow and congestion control

- Flow control: receive window advertised by the receiver
- Congestion control: congestion window set by the sender

TCP segment header format

~~RFC 793~~ RFC 9293



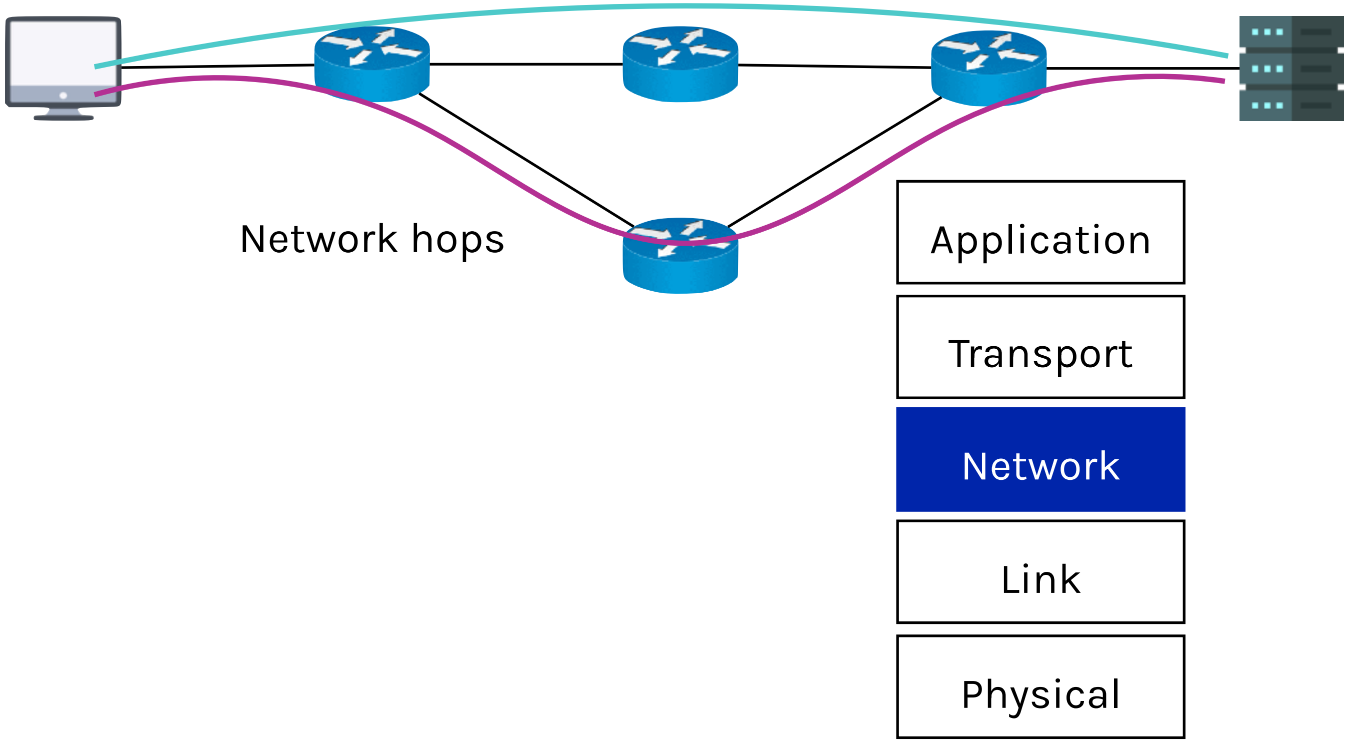
TCP segment header

Offsets	Octet	0								1								2								3							
Octet	Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset				Reserved 0 0 0			NS	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size															
16	128	Checksum																Urgent pointer (if URG set)															
20	160	Options (if *data offset* > 5. Padded at the end with "0" bytes if necessary.)																															
⋮	⋮																																
60	480																																

Network Routing

Network routing

Key question: how to identify computers and how to find a router between computers?



Network hops

Application

Transport

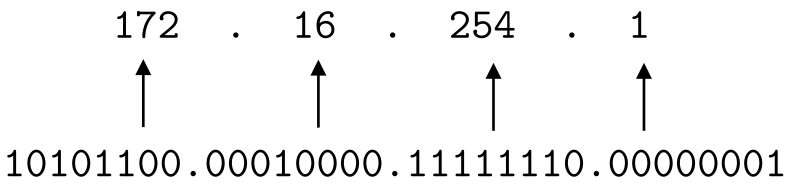
Network

Link

Physical

Network layer address: example IPv4

RFC7020



172 . 16 . 254 . 1

10101100.00010000.11111110.00000001

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

Network Identifier (IP prefix)	Host Identifier

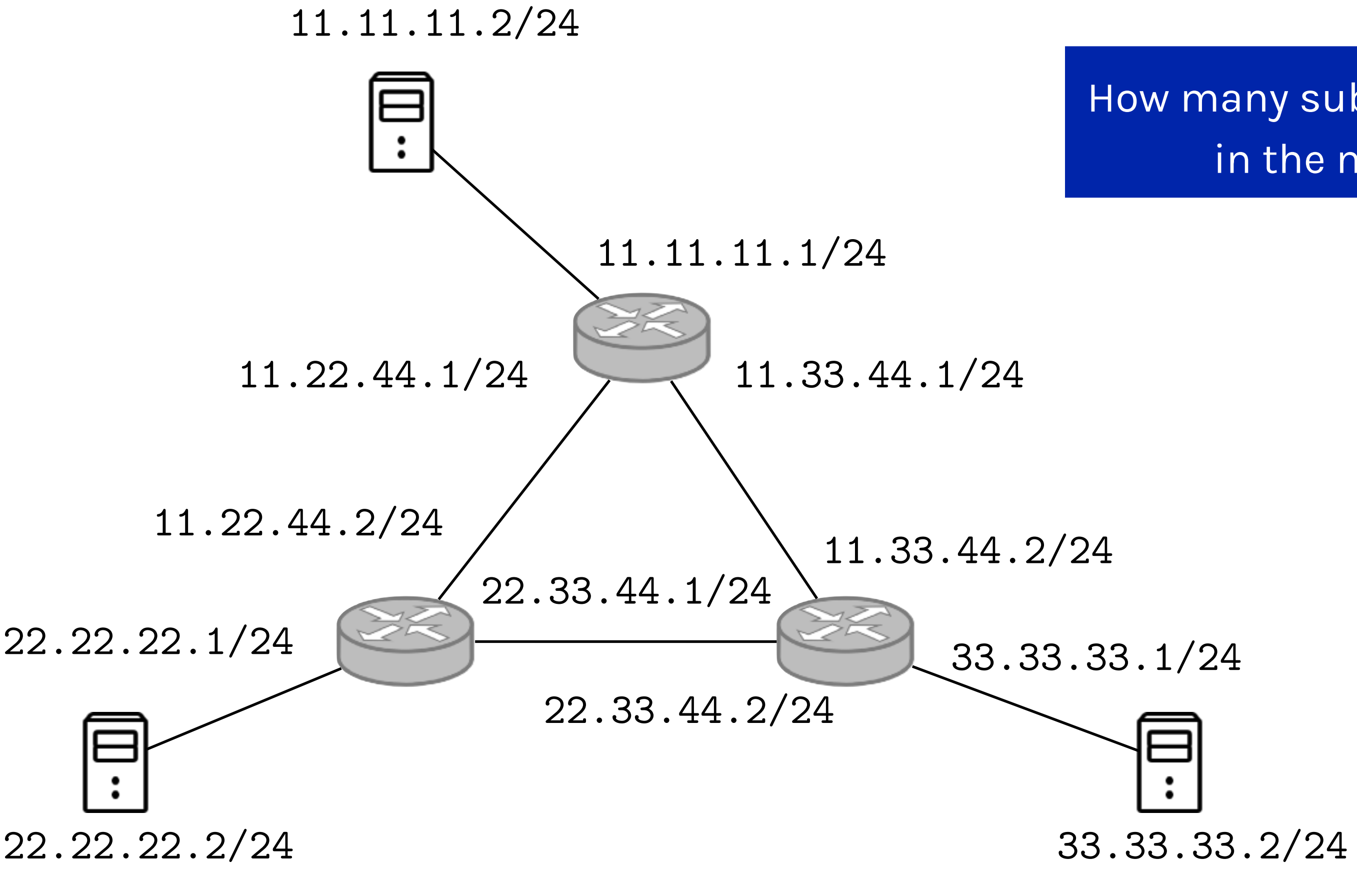
Network Identifier	Subnet Identifier	Host Identifier

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

Subnet mask notation: 255.255.255.0

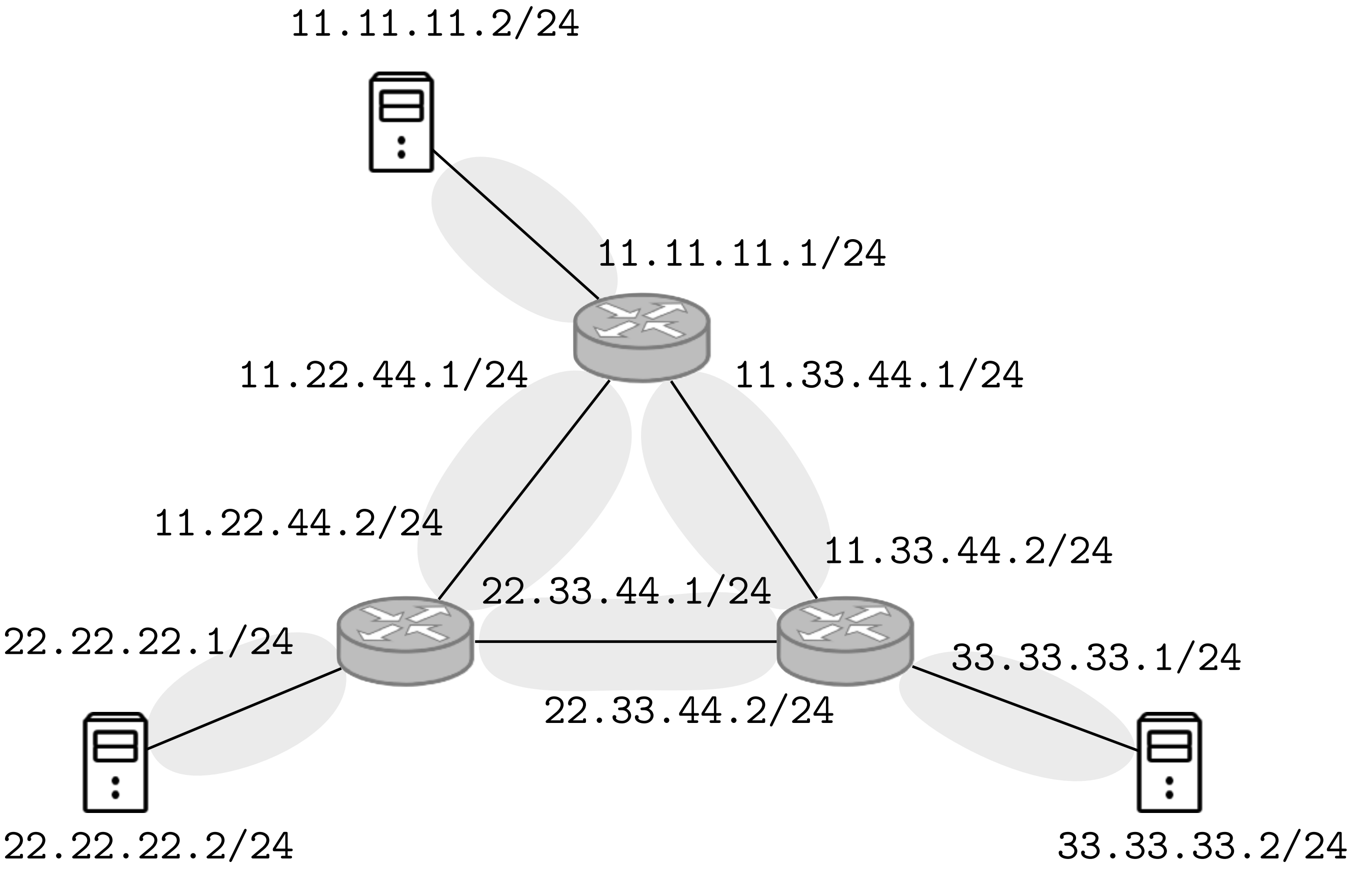
Who do we assign IP addresses to? A host? switch? router? or…

Routers interconnecting subnets

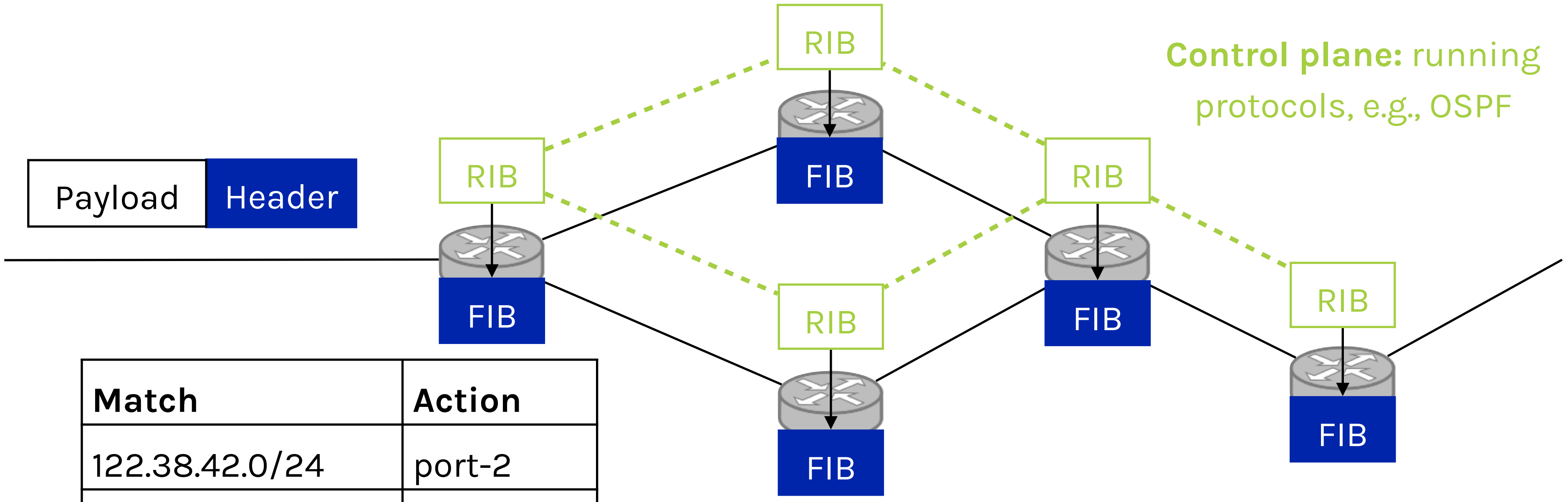


How many subnets are there in the network?

Routers interconnecting subnets



IP routing



Control plane: running protocols, e.g., OSPF

Match	Action
122.38.42.0/24	port-2
116.16.0.0/16	port-1
139.70.8.0/24	drop

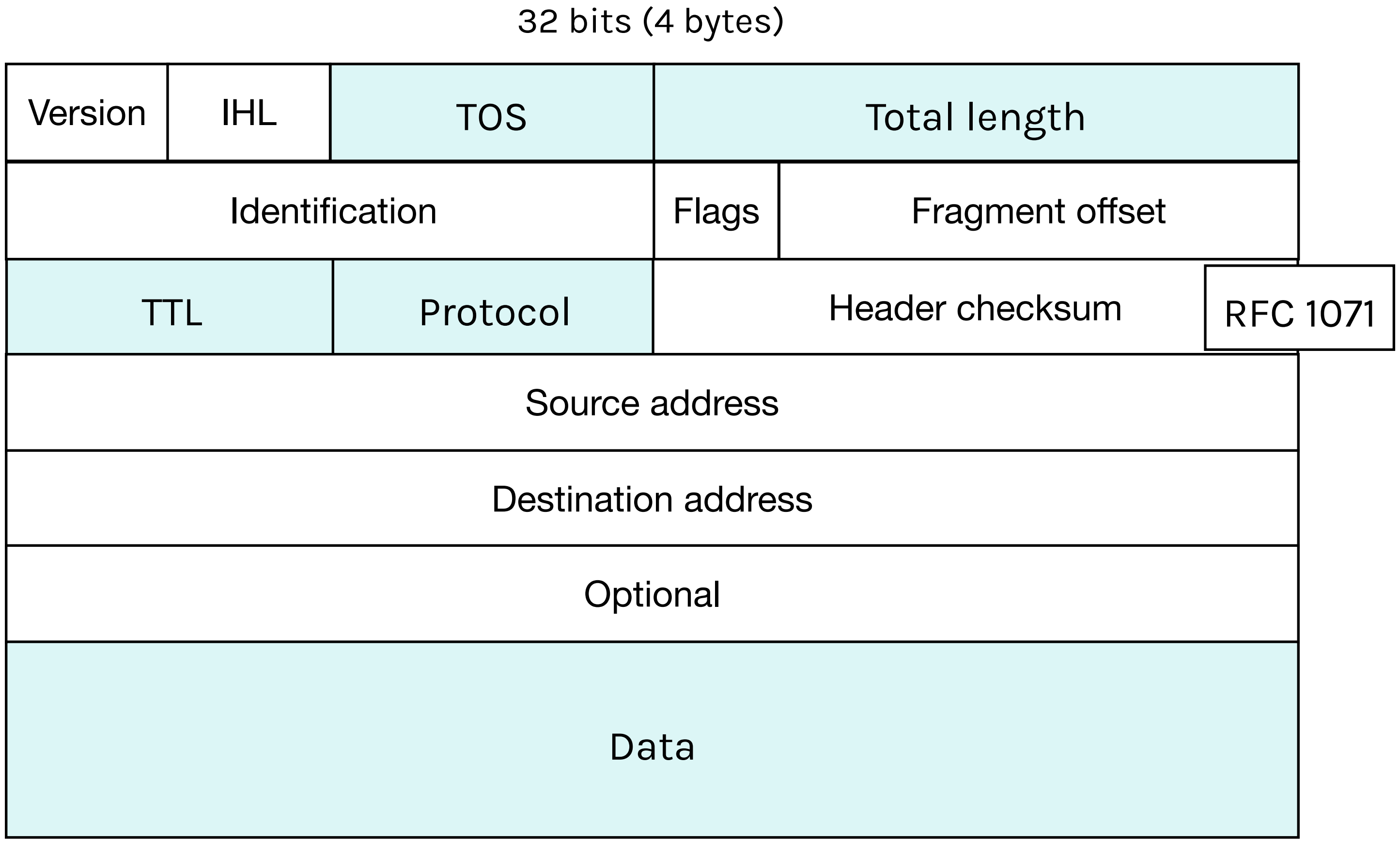
Data plane: packet forwarding with the match-action model

RIB: routing information base, or routing table

FIB: forwarding information base

IPv4 packet format

RFC 791



32 bits (4 bytes)

Version | IHL | TOS | Total length

Identification | Flags | Fragment offset

TTL | Protocol | Header checksum

RFC 1071

Source address

Destination address

Optional

Data

TOS: type of service, two bits used for Explicit Congestion Notification

RFC 3168

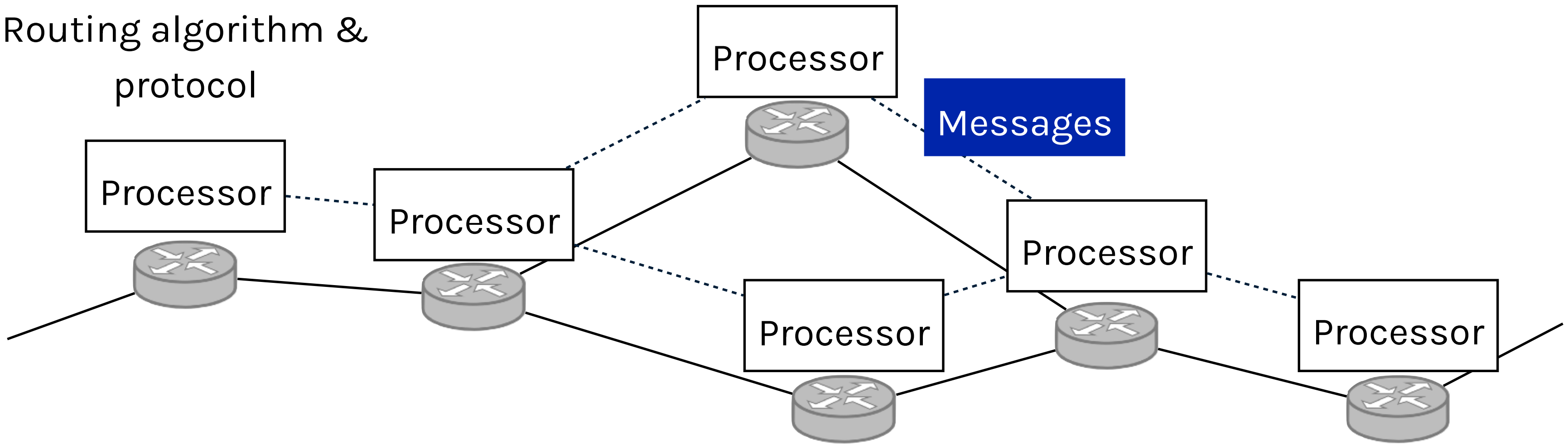
Total length: max. 65535 bytes, typically bounded by Ethernet MTU (1500 bytes)

TTL: decreased by one when passing a router, packet dropped by the router when it reaches 0

Protocol: transport layer protocol (6 for TCP, 17 for UDP)

How to generate forwarding tables?

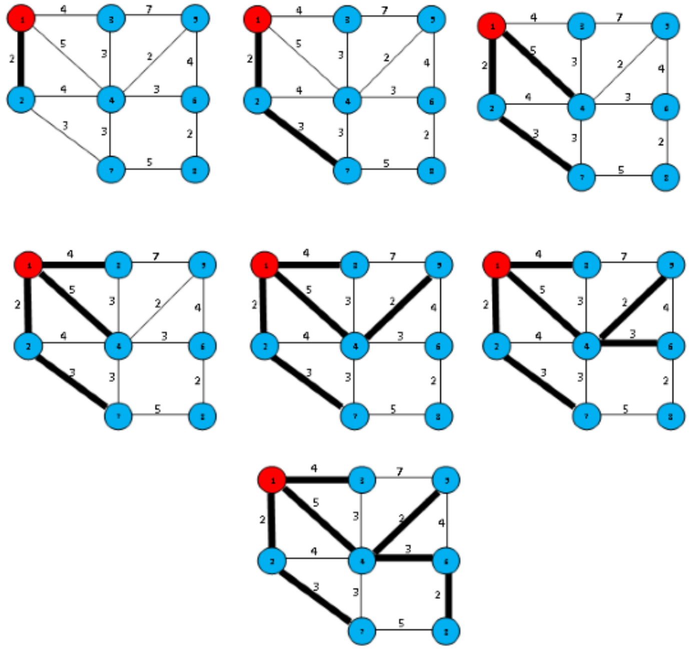
Control plane: distributed protocols based on some shortest-path algorithms (e.g., OSPF, BGP)



Routing protocol: intra-domain

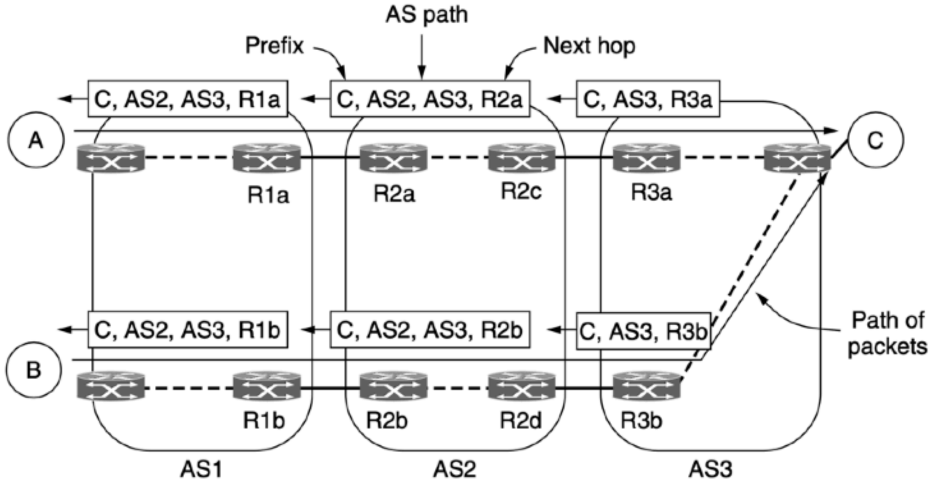
Open Shortest Path First (OSPF):

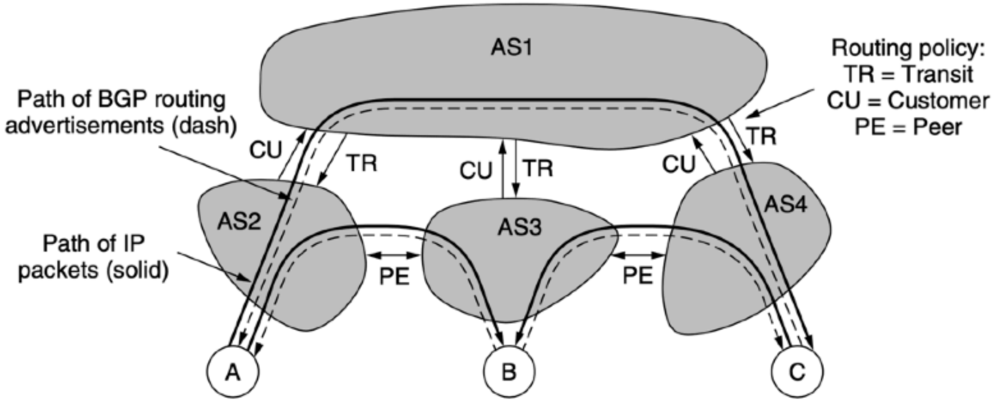
- Routers exchange link-state messages to learn the topology
- Each router runs the **Dijkstra's algorithm** to computer the shortest paths to other routers
- Each router generates the forwarding table entries based on the shortest paths



Routing protocol: inter-domain (BGP)

Routers exchange **path vectors** to form shortest paths between ASes





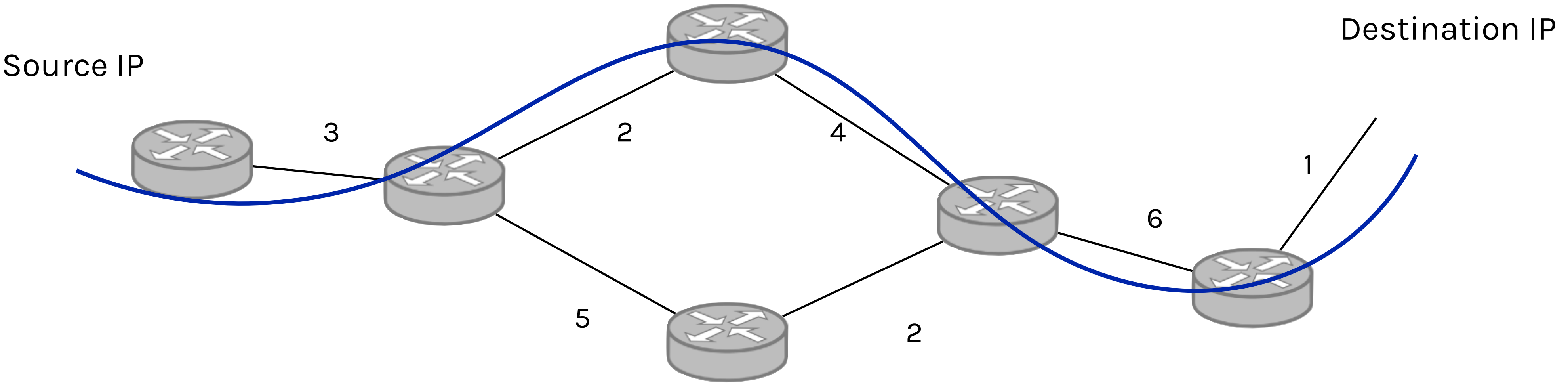
Andrew S. Tanenbaum, David Wetherall. Computer Networks (5th edition), Pearson Education, 2011.

Traffic engineering

RFC 3272

RFC 2702

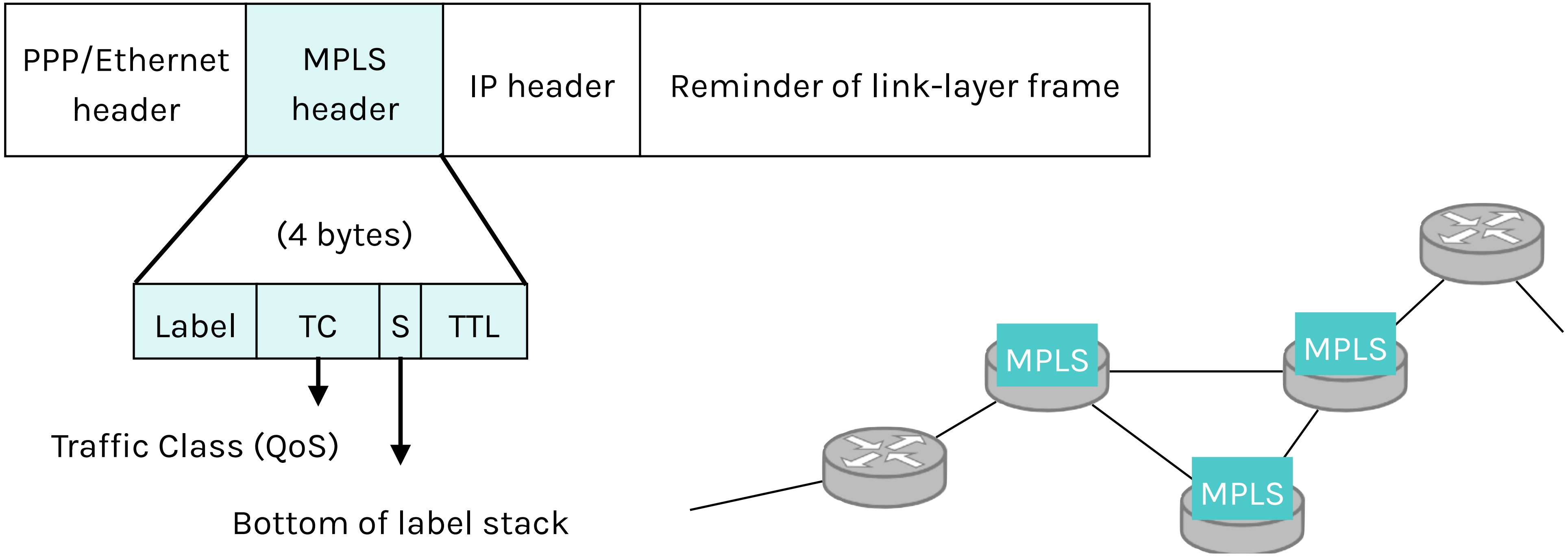
Performance evaluation and performance optimization: measurement, characterization, **modeling**, and **control** of Internet traffic



What limitations and associated performance issues can you see in network routing?

Multiprotocol label switching (MPLS)

RFC 3031



PPP/Ethernet header	MPLS header	IP header	Reminder of link-layer frame

(4 bytes)

Label	TC	S	TTL

Traffic Class (QoS)

Bottom of label stack

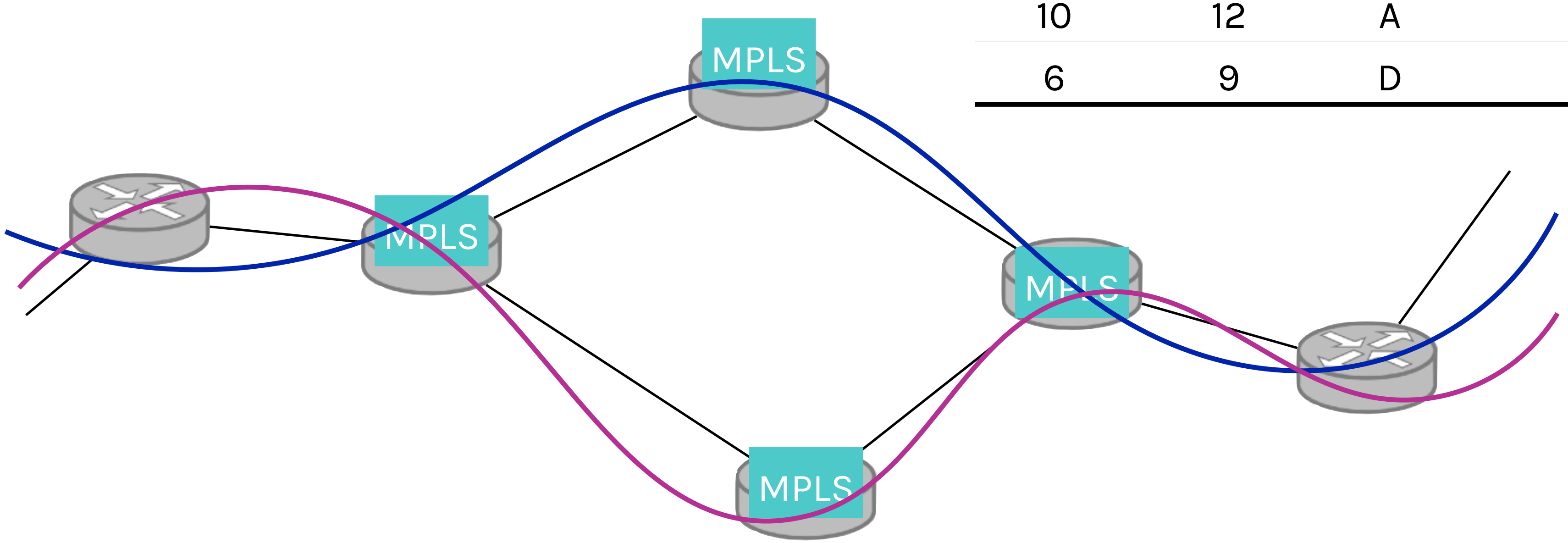
Label switched router

Traffic engineering with MPLS

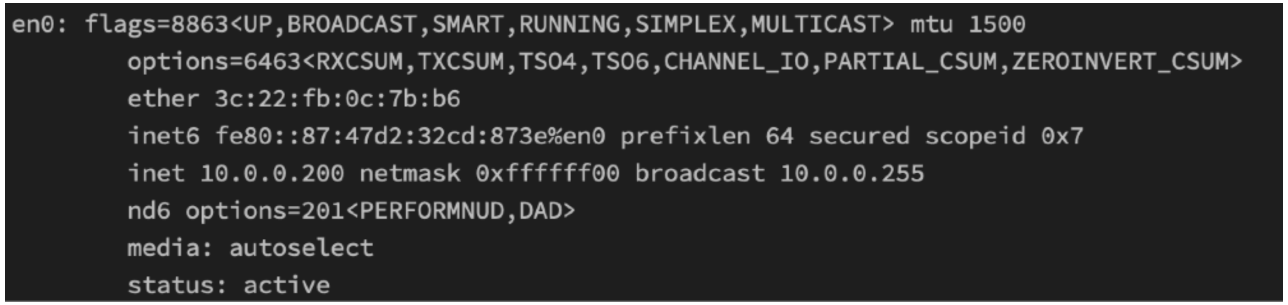
RFC 2702 RFC 3272 RFC 3346

Even for the same source-destination (IP) pair, multiple paths can be set up for forwarding the traffic. By carefully assigning the labels, we can control how the traffic is shipped on the network links - traffic engineering

In-label	Out-label	Dest	Out interface
10	12	A	1
6	9	D	0

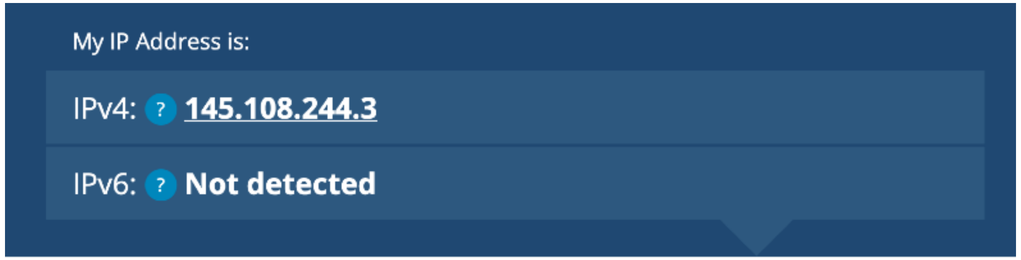


Network Addresss Translation



```
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=6463<RXCSUM,TXCSUM,TSO4,TSO6,CHANNEL_IO,PARTIAL_CSUM,ZEROINVERT_CSUM>
        ether 3c:22:fb:0c:7b:b6
        inet6 fe80::87:47d2:32cd:873e%en0 prefixlen 64 secured scopeid 0x7
        inet 10.0.0.200 netmask 0xffffff00 broadcast 10.0.0.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
```

IP as you see from your computer: 10.0.0.200



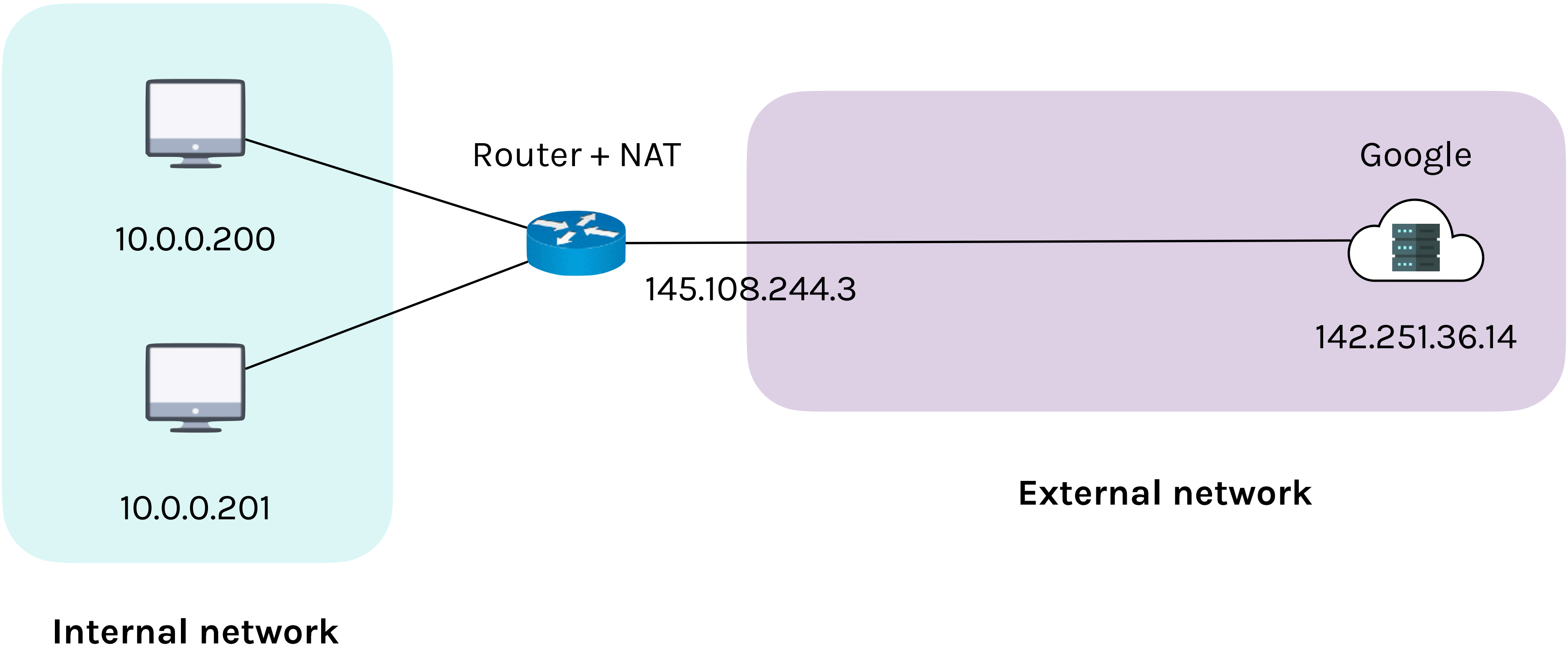
My IP Address is:

IPv4: ? 145.108.244.3

IPv6: ? Not detected

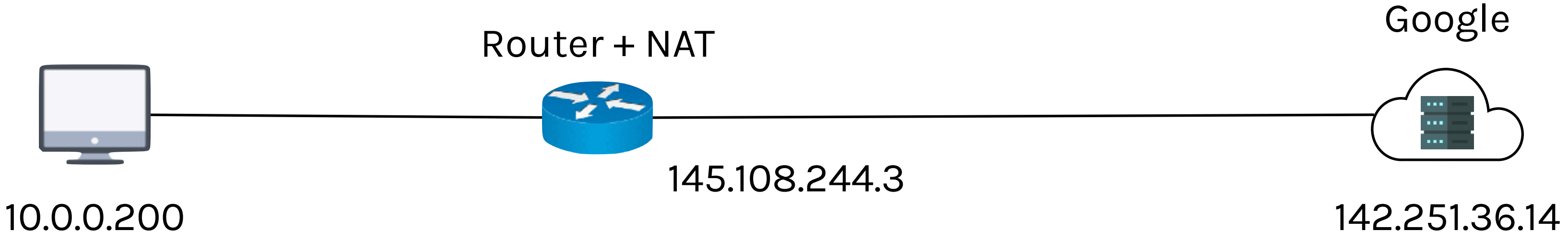
IP seen from outside: 145.108.244.3

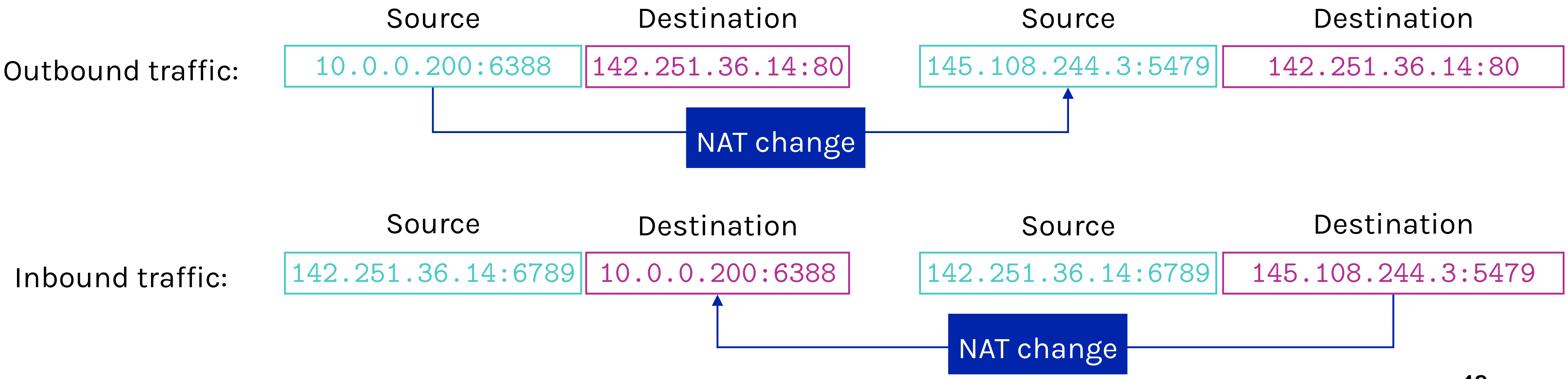
Network address translation (NAT)



NAT example

Source	Destination
10.0.0.200:6388	145.108.244.3:5479
10.0.0.300:7173	145.108.244.3:5480





	Source	Destination	Source	Destination
Outbound traffic:	10.0.0.200:6388	142.251.36.14:80	145.108.244.3:5479	142.251.36.14:80
Inbound traffic:	142.251.36.14:6789	10.0.0.200:6388	142.251.36.14:6789	145.108.244.3:5479

NAT pros and cons

Pros

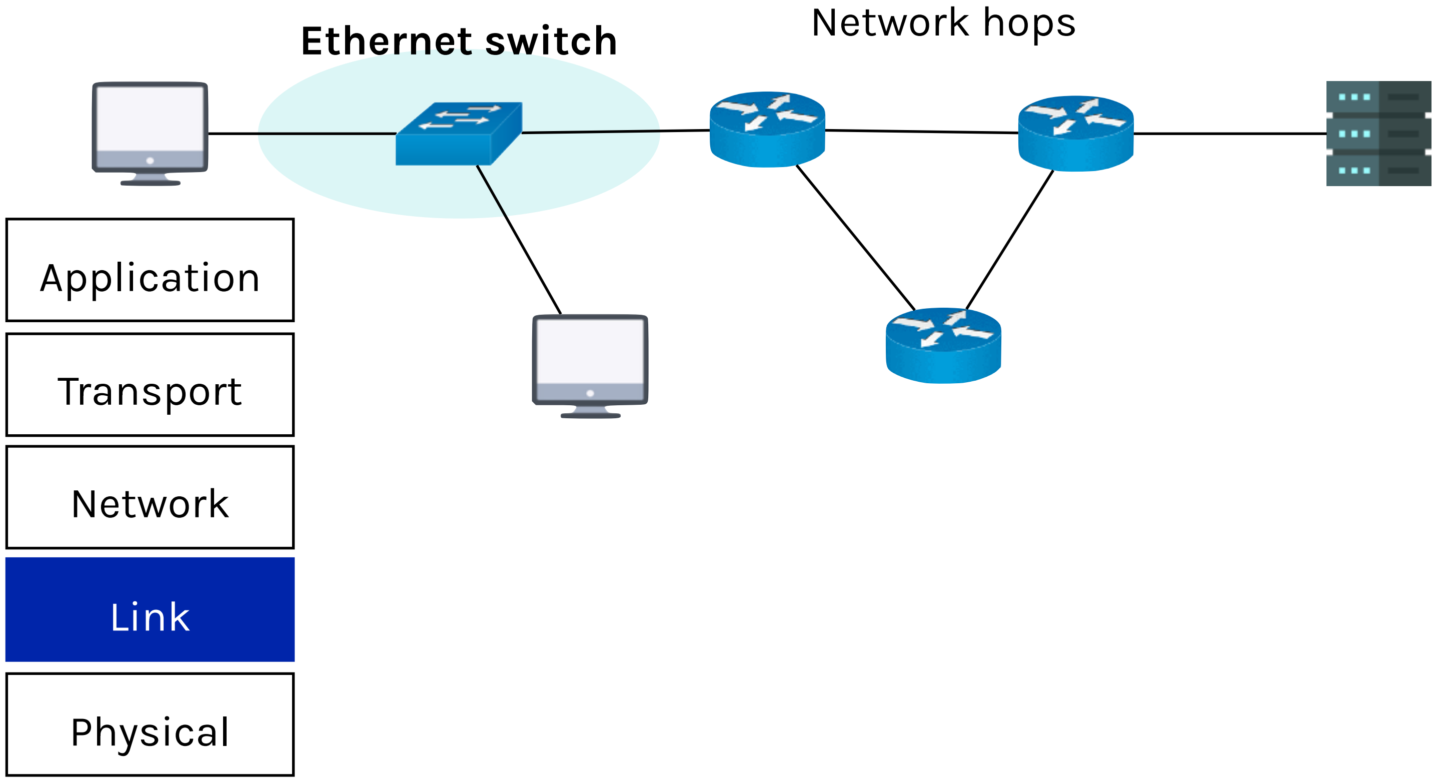
- Mitigates IPv4 address exhaustion problem: reuse IPv4 addresses in private networks
- Destination NAT for port forwarding: hiding internal servers, load balancing

Cons

- Hard to establish peer-to-peer connections
- Violates the end-to-end principle!

Switching

Link-layer forwarding



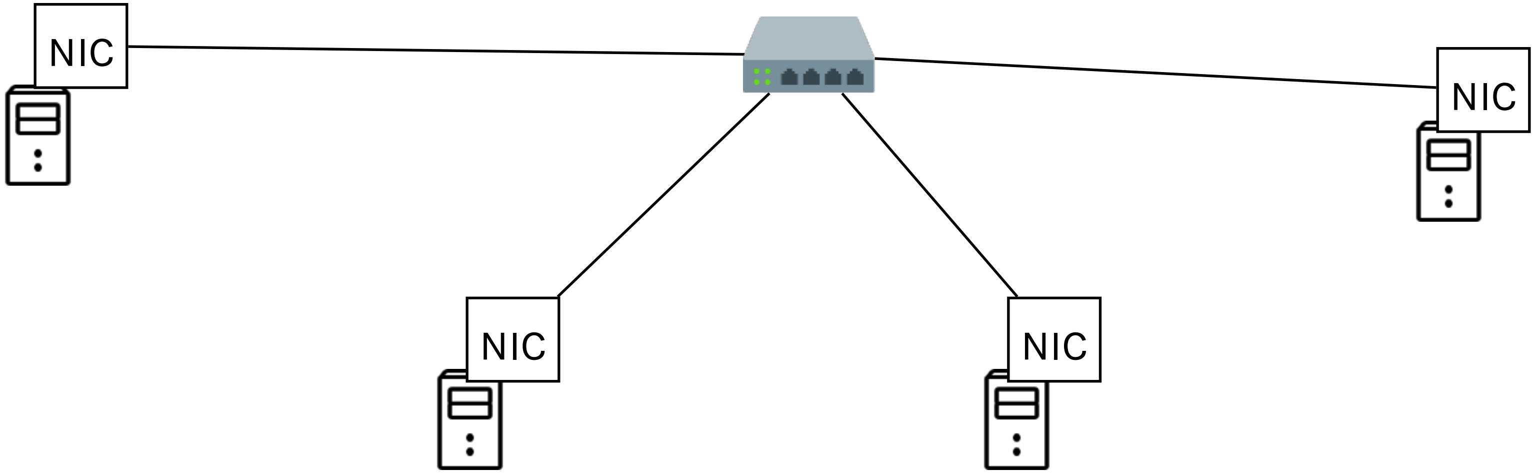
Ethernet

IEEE 802.3

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

Hub (repeater): replicates signals to all ports except the one that signals were received on

OBSOLETE



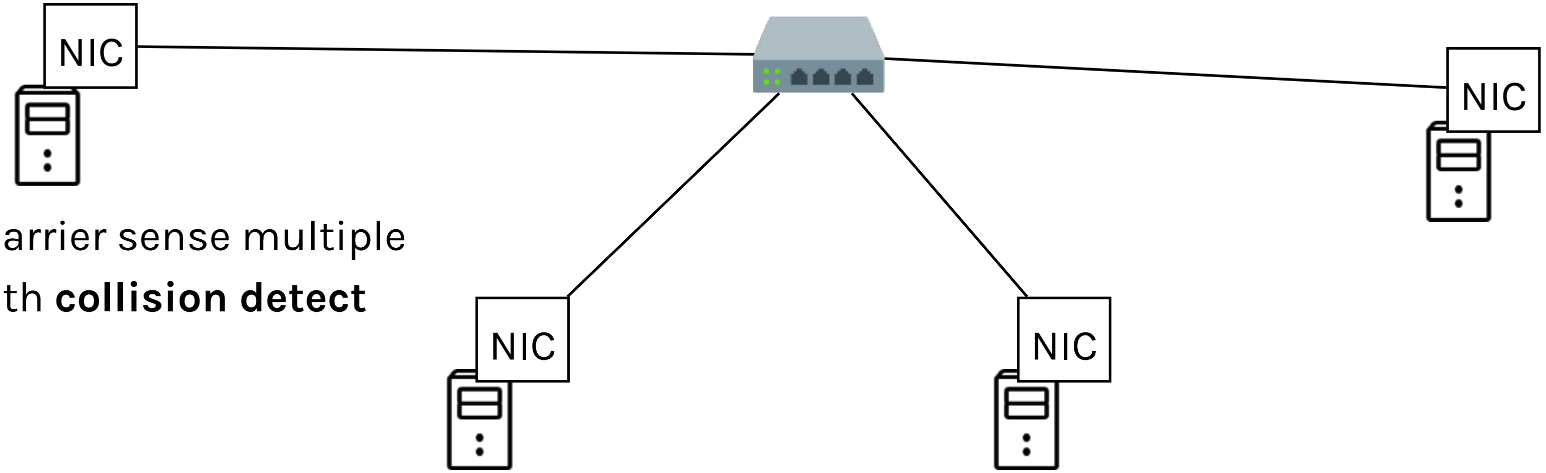
Ethernet

IEEE 802.3

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

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

OBSOLETE



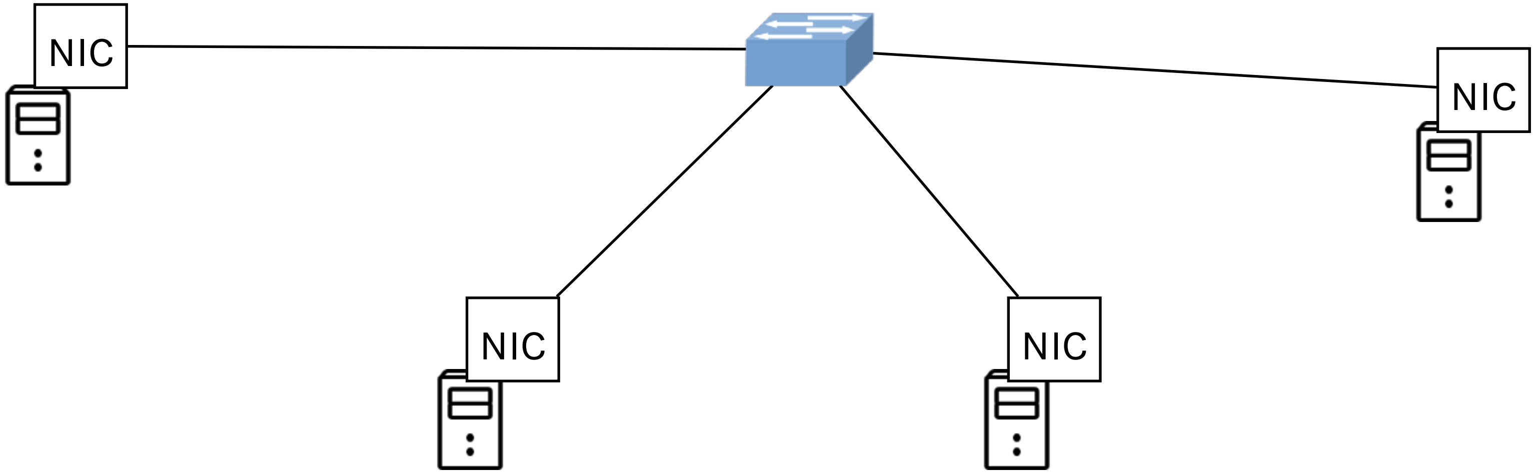
CSMA/CD: carrier sense multiple access with **collision detect**

Switched Ethernet

Different Ethernet segments are interconnected with switches

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

Switches typically do not need to run CSMA/CD, why?

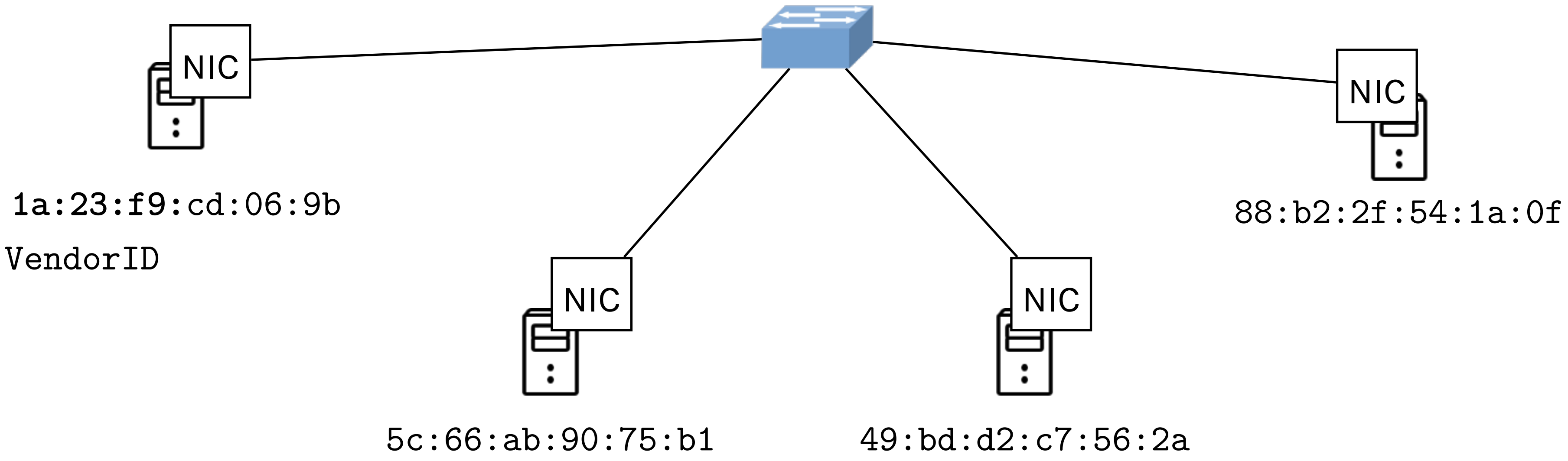


Ethernet MAC address

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

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

Do switches need MAC addresses? Why?



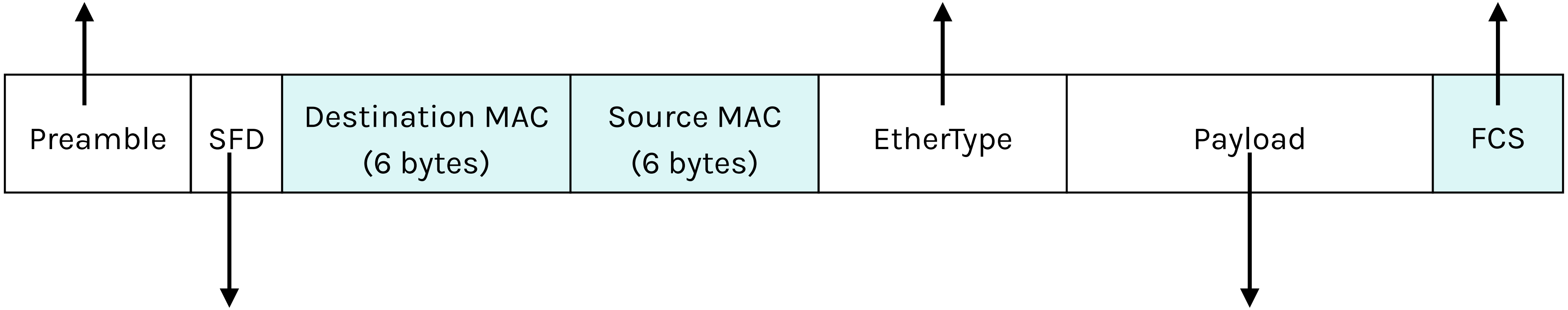
Ethernet frame structure

IEEE 802.3

Alternating 0/1s to allow for bit-level sync (7 bytes)

Specifies the upper-layer protocol (2 bytes), e.g., IPv4 (0800), ARP (0806)

Frame Check Sequence, i.e., CRC (4 bytes)



Preamble | SFD | Destination MAC (6 bytes) | Source MAC (6 bytes) | EtherType | Payload | FCS

Start Frame Delimiter (10101011) allows for frame-level sync (1 byte)

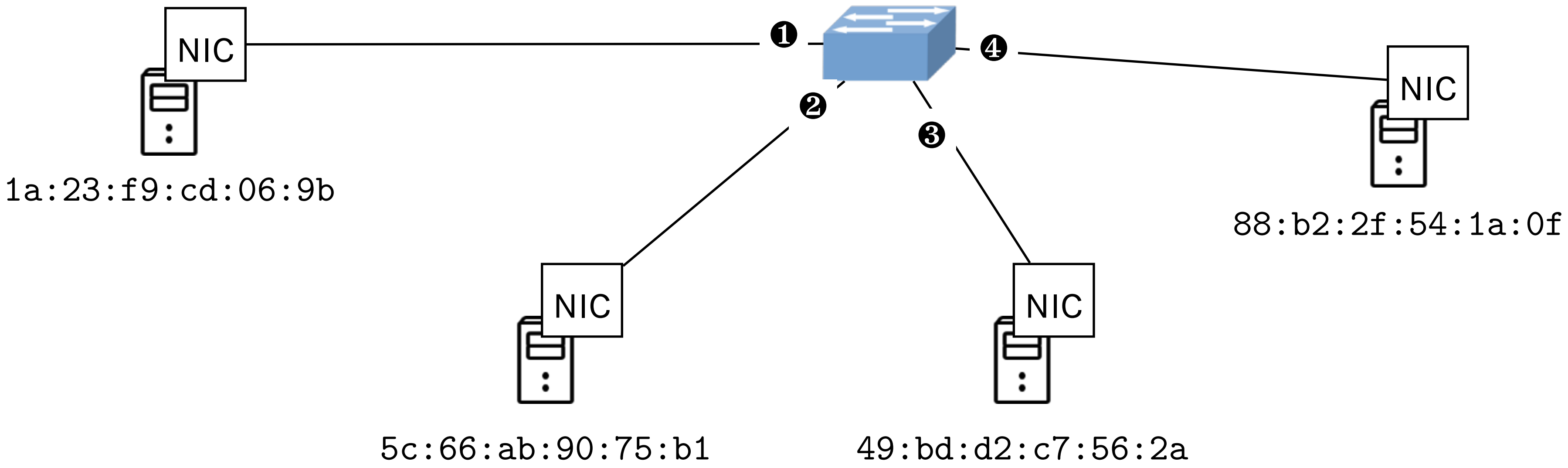
Carries the IP packet, max size decided by MTU (1500 bytes for Ethernet), stuffed if less than 46 bytes

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

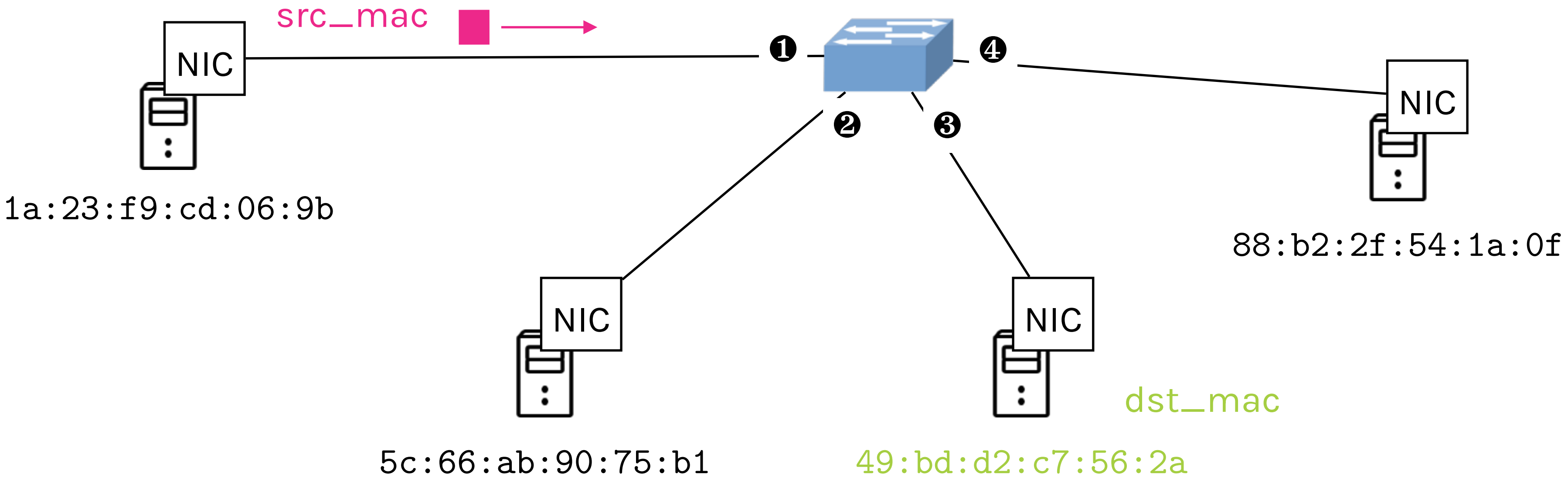
How to configure the forwarding table?



Self learning

Learn new MAC-interface mappings through incoming frames

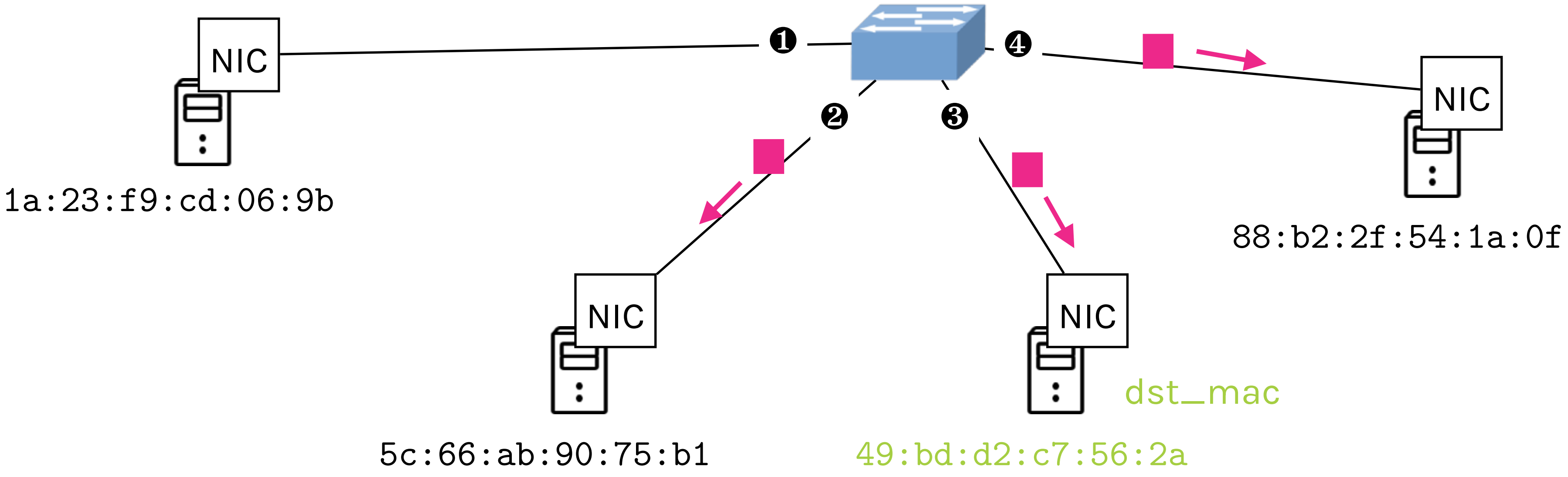
MAC	Interface	Time
88:b2:2f:54:1a:0f	4	9:32
5c:66:ab:90:75:b1	2	9:34
1a:23:f9:cd:06:9b	1	10:00



Self learning

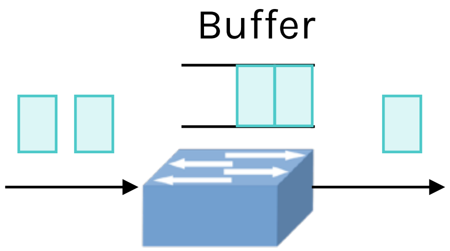
Broadcast the new frame with unknown destination MAC on all interfaces but the one that has received the frame

MAC	Interface	Time
88:b2:2f:54:1a:0f	4	9:32
5c:66:ab:90:75:b1	2	9:34
1a:23:f9:cd:06:9b	1	10:00



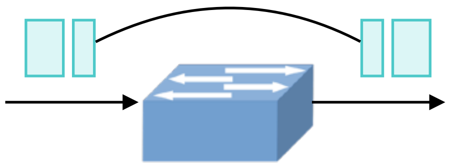
Store-and-forward vs. cut-through

Store-and-forward



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

Cut-through

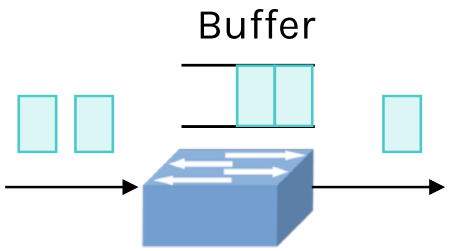


Once lookup is done, packet receiving and sending happen at the same time.

What are the pros and cons of each approach?

Store-and-forward vs. cut-through

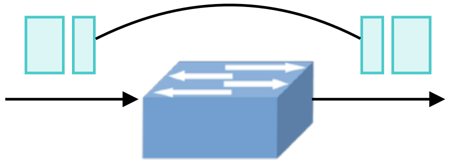
Store-and-forward



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

Integrity checks are possible, but the frame has to **wait** in the buffer before being sent out.

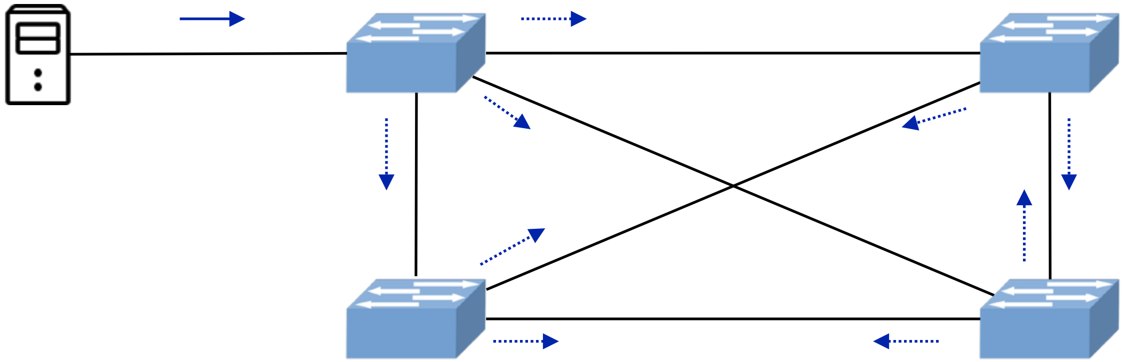
Cut-through



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

Frames are sent out with **low latency**, but **integrity checks** become impossible.

Problem #1: when flooding meets loops



Each frame leads to the creation of at least two new frames.

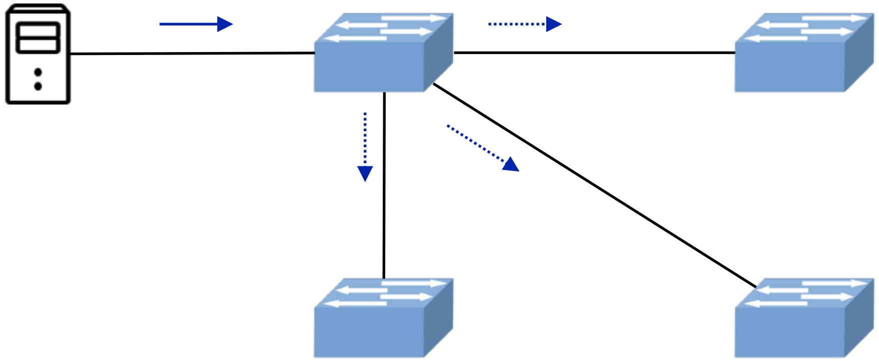
Exponential increase, with no TTL to remove looping frames...

Redundancy without loops

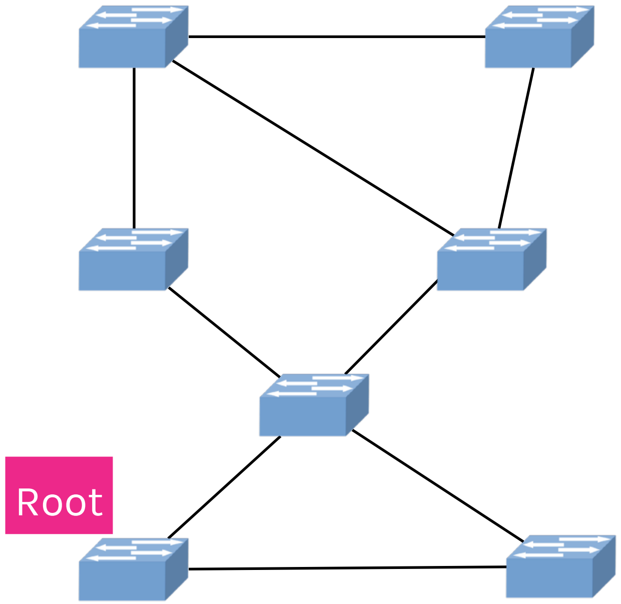
Solution

- Reduce the network to one logical spanning tree
- Upon failure, automatically rebuild a spanning tree

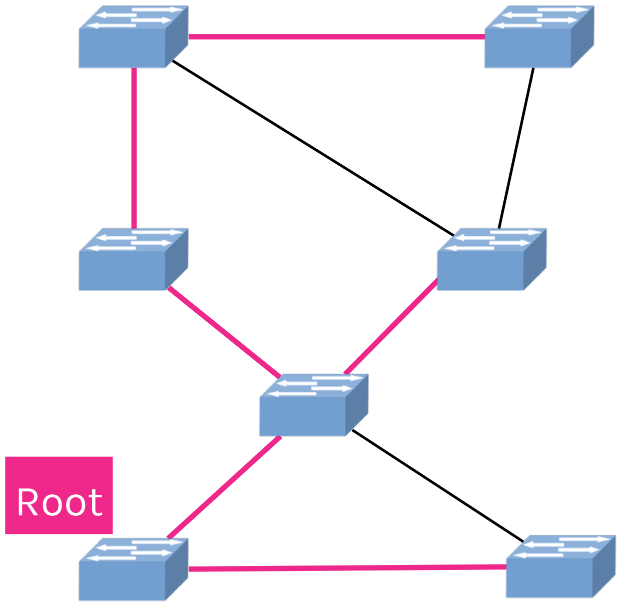
In practice, switches run a distributed spanning tree protocol (STP)



STP example



Select the root



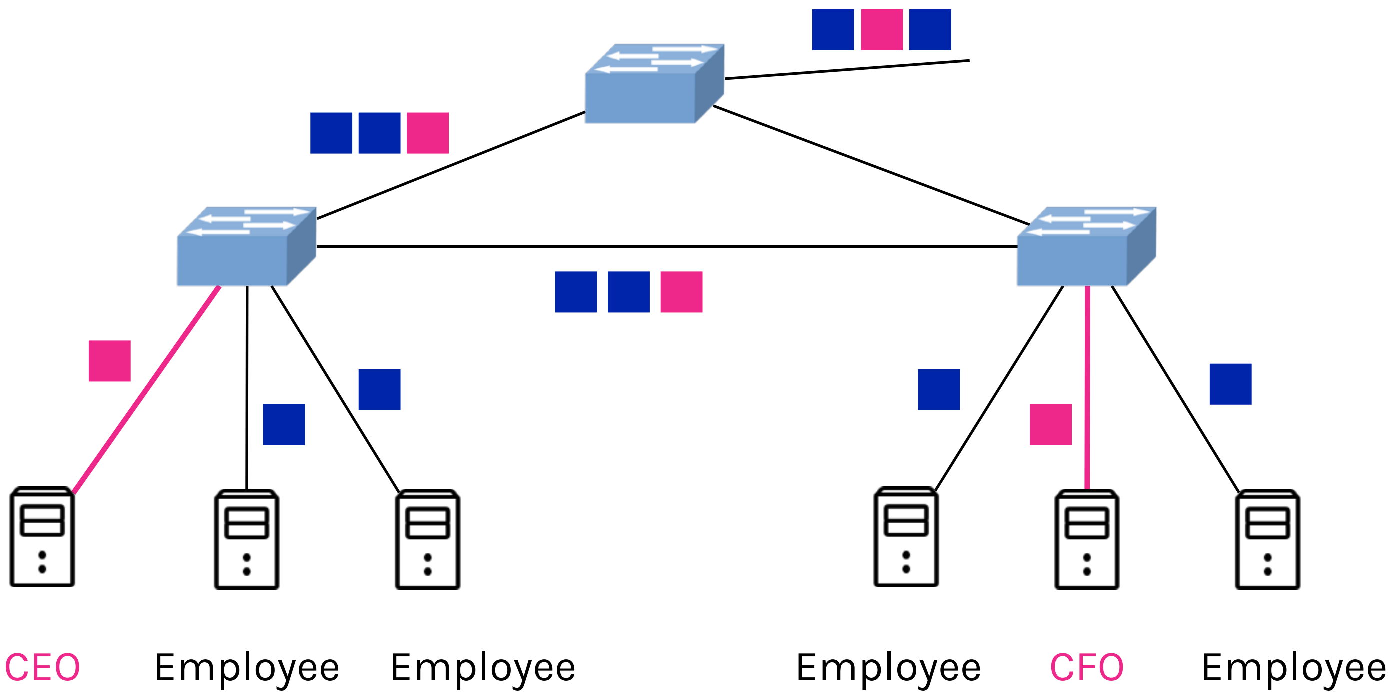
Keep shortest paths to root

To ensure robustness, the root switch keeps sending the messages.
If timeout, switches claim itself to be root.

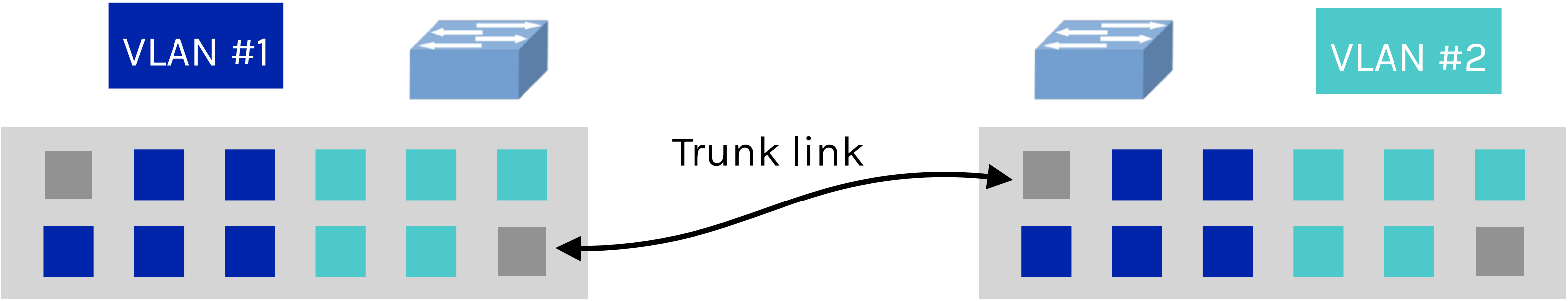
Problem #2: traffic isolation

Broadcast packets cannot be localized and can cause broadcast storm in the network

Hard user management: A user has to be connected to the a particular switch in order to isolate its traffic



VLAN

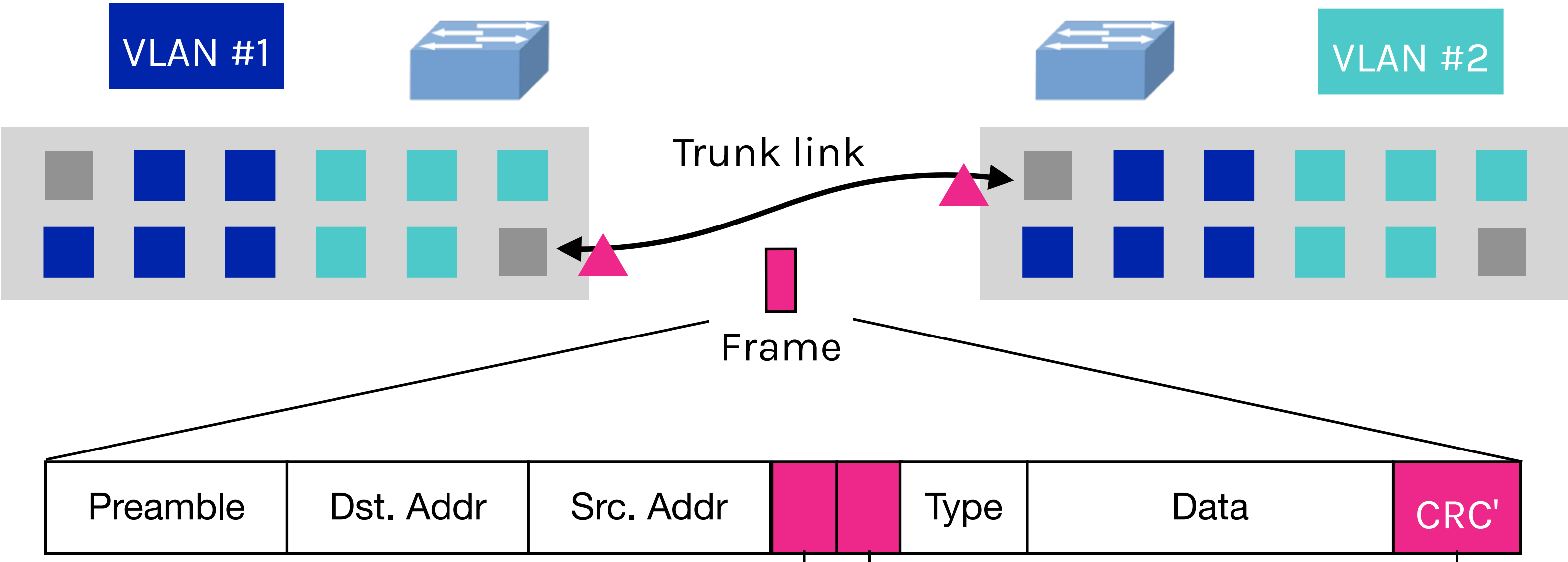


1. Network manager can **partition the ports** into subsets and assign them to VLANs
2. Ports in the same VLAN form a broadcast domain, while ports on different VLANs are routed through an internal router within the switch
3. Switches are connected on trunk ports that belong to all VLANs

How does a receiving switch know which VLAN a frame belongs to?

VLAN tag

IEEE 802.1Q



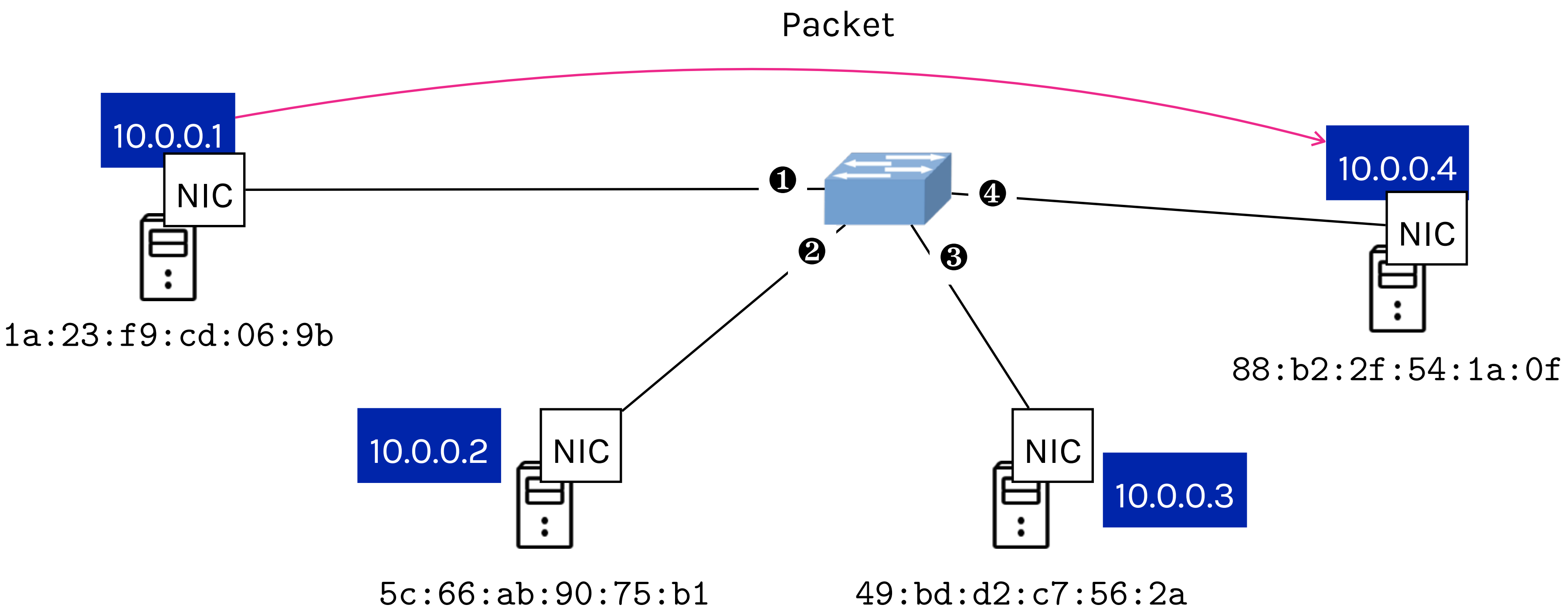
Preamble	Dst. Addr	Src. Addr			Type	Data	CRC'

Tag Protocol Identifier
(2 bytes, fixed at 8100)

Tag Control Information
(2 bytes, **12bits VID**, 3 bits priority code point, 1 bit drop eligible indicator)

Recomputed frame check sequence (FCS)

How to obtain the destination MAC address?

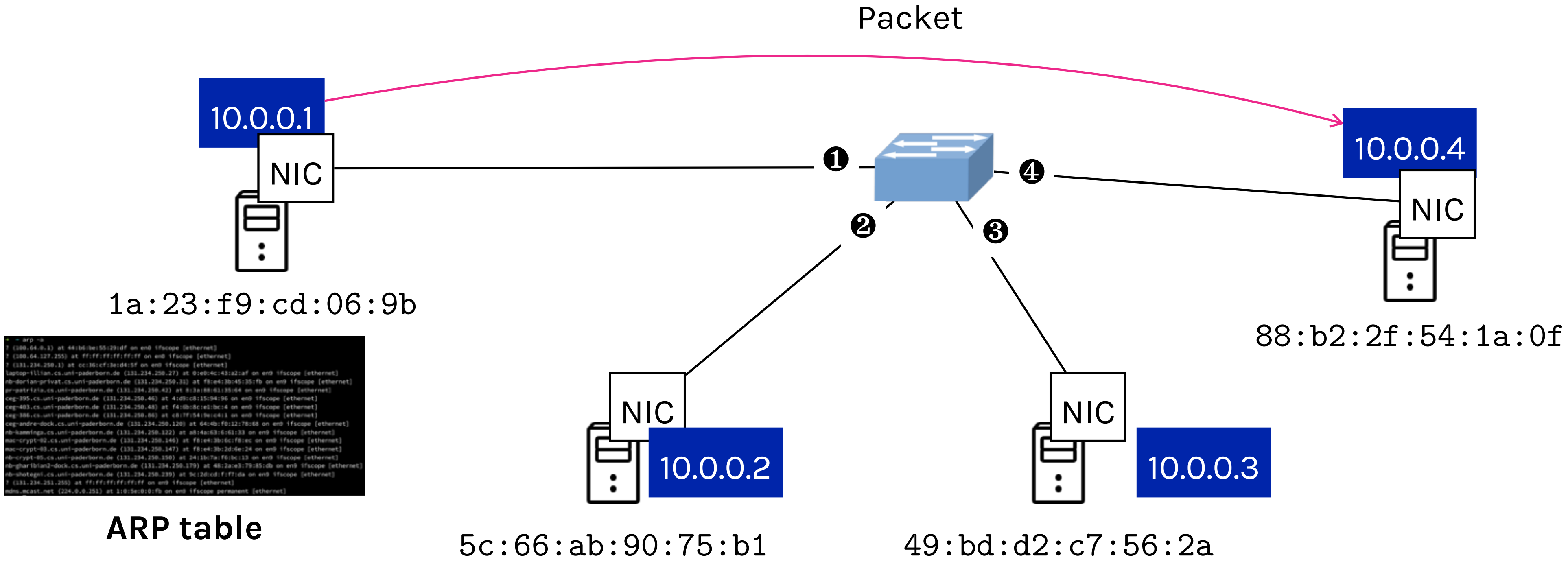


How to obtain the destination MAC address?

RFC 826

ARP query: Whoever has the IP address `10.0.0.4`, please tell me your MAC address

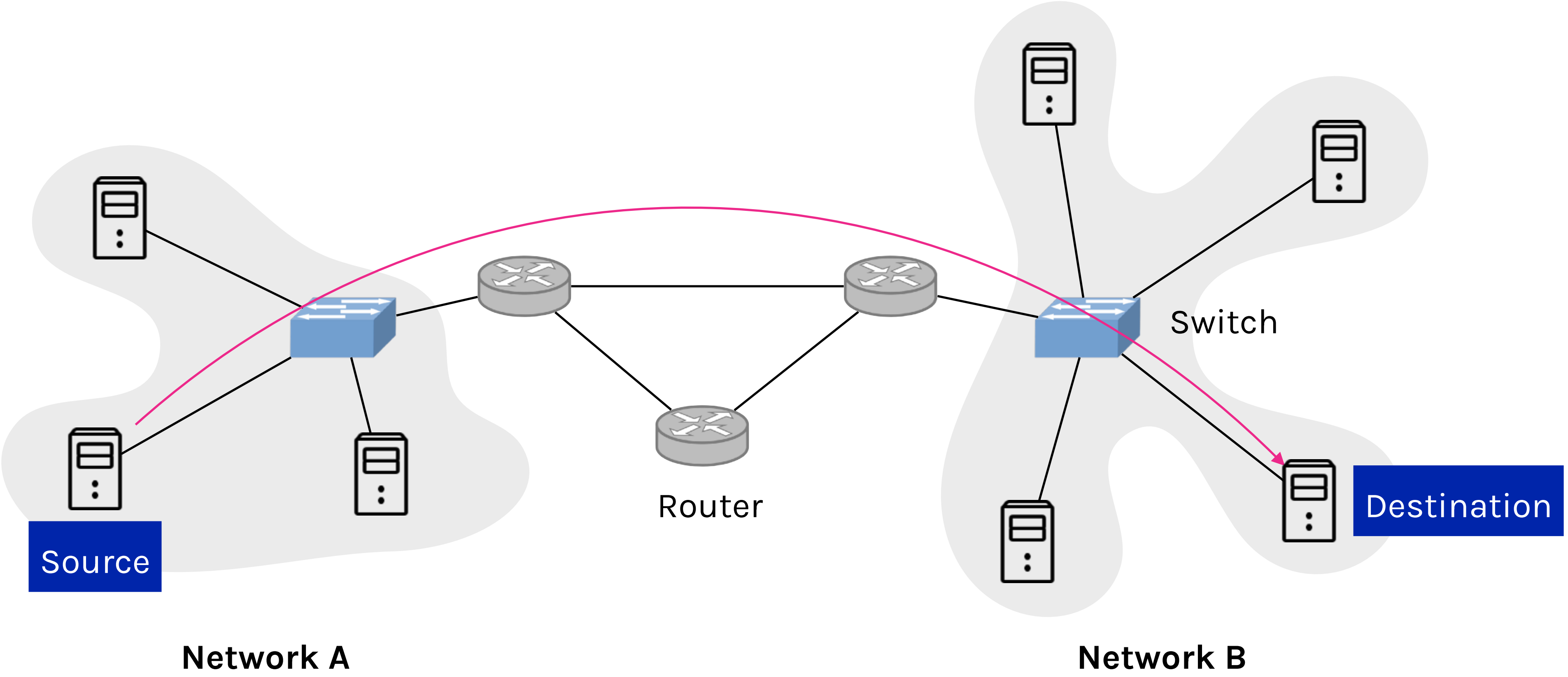
ARP reply: that is me, my MAC address is `88:b2:2f:54:1a:0f`



```
→ ~ arp -a
? (100.64.0.1) at 44:b6:be:55:29:df on en0 ifscope [ethernet]
? (100.64.127.255) at ff:ff:ff:ff:ff:ff on en0 ifscope [ethernet]
? (131.234.250.1) at cc:36:cf:3e:d4:5f on en9 ifscope [ethernet]
laptop-illian.cs.uni-paderborn.de (131.234.250.27) at 0:e0:4c:43:a2:af on en9 ifscope [ethernet]
nb-dorian-privat.cs.uni-paderborn.de (131.234.250.31) at f8:e4:3b:45:35:fb on en9 ifscope [ethernet]
pr-patrizia.cs.uni-paderborn.de (131.234.250.42) at 8:3a:88:61:35:64 on en9 ifscope [ethernet]
ceg-395.cs.uni-paderborn.de (131.234.250.46) at 4:d9:c8:15:94:96 on en9 ifscope [ethernet]
ceg-403.cs.uni-paderborn.de (131.234.250.48) at f4:6b:8c:e1:bc:4 on en9 ifscope [ethernet]
ceg-386.cs.uni-paderborn.de (131.234.250.86) at c8:7f:54:9e:c4:1 on en9 ifscope [ethernet]
ceg-andre-dock.cs.uni-paderborn.de (131.234.250.120) at 64:4b:f0:12:78:68 on en9 ifscope [ethernet]
nb-kamminga.cs.uni-paderborn.de (131.234.250.122) at a8:4a:63:6:61:33 on en9 ifscope [ethernet]
mac-crypt-02.cs.uni-paderborn.de (131.234.250.146) at f8:e4:3b:6c:f8:ec on en9 ifscope [ethernet]
mac-crypt-03.cs.uni-paderborn.de (131.234.250.147) at f8:e4:3b:2d:6e:24 on en9 ifscope [ethernet]
nb-crypt-05.cs.uni-paderborn.de (131.234.250.150) at 24:1b:7a:f6:bc:13 on en9 ifscope [ethernet]
nb-gharibian2-dock.cs.uni-paderborn.de (131.234.250.179) at 48:2a:e3:79:85:db on en9 ifscope [ethernet]
nb-shotegni.cs.uni-paderborn.de (131.234.250.239) at 9c:2d:cd:f:f7:da on en9 ifscope [ethernet]
? (131.234.251.255) at ff:ff:ff:ff:ff:ff on en9 ifscope [ethernet]
mdns.mcast.net (224.0.0.251) at 1:0:5e:0:0:fb on en9 ifscope permanent [ethernet]
```

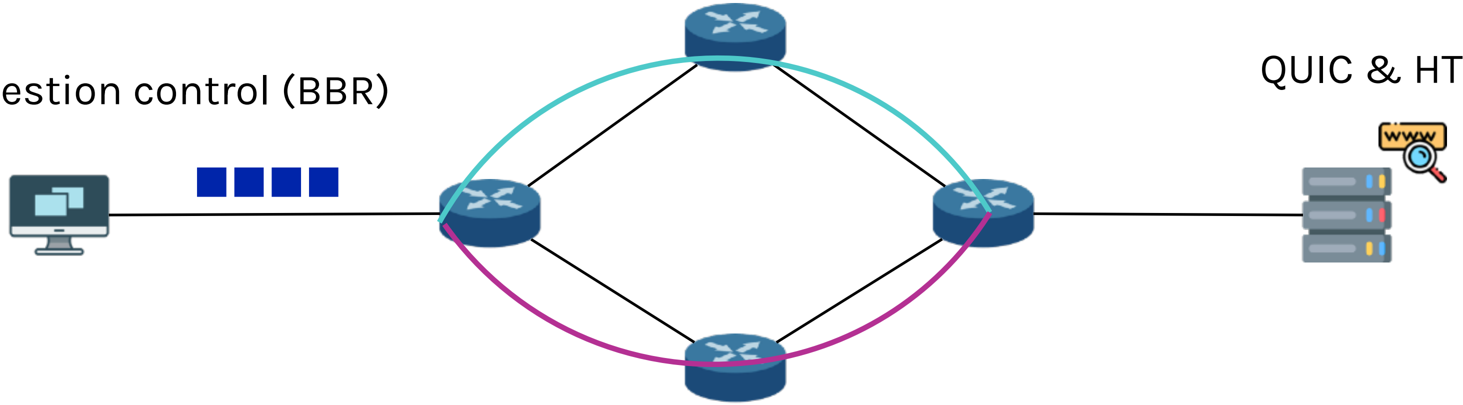
ARP table

Exercise



What steps are involved?

Next lecture: network transport



Congestion control (BBR)

QUIC & HTTP3.0

Multi-path TCP