Advanced Network Programming

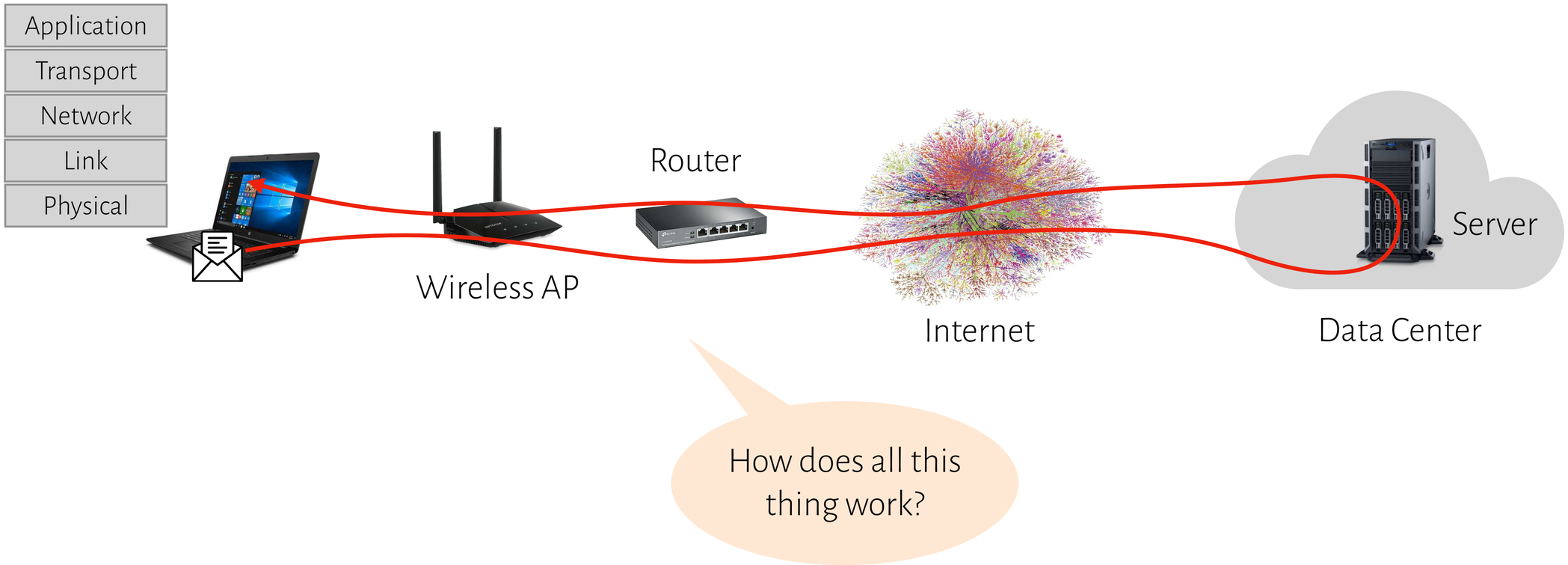
Network Forwarding and Routing

Lin Wang

Fall 2020, Period 1

Part of the content is adapted from Kurose&Ross, Computer Networking: A Top-Down Approach, Pearson

What happens when a packet leaves your computer?

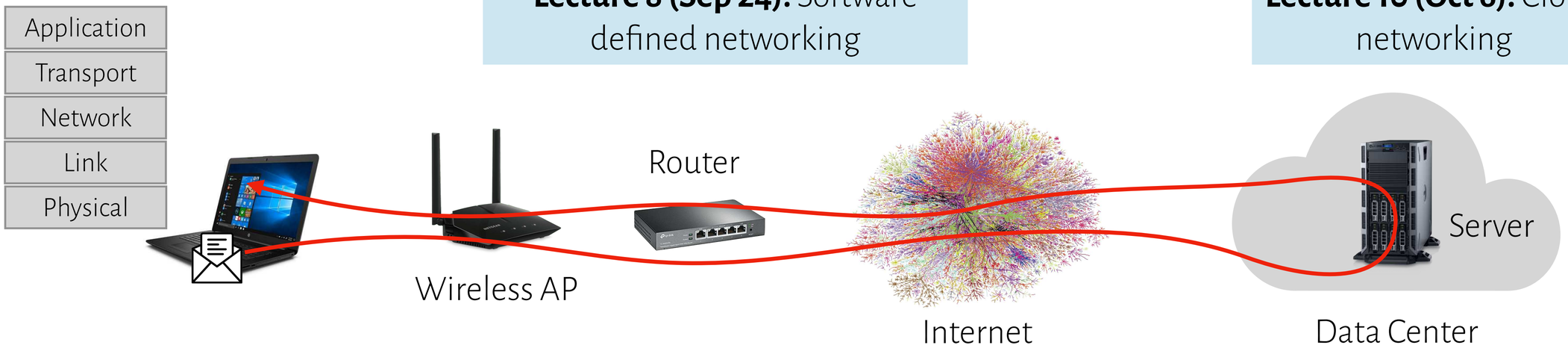


Part 2: network infrastructure

Lecture 11 (Oct 8): Beyond networking

Lecture 8 (Sep 24): Software defined networking

Lecture 10 (Oct 6): Cloud networking



Lecture 7 (Sep 22): Network forwarding and routing

Lecture 9 (Oct 1): Programmable data plane

Part 2: network infrastructure

Lecture 7: Network forwarding and routing

- The link layer and Ethernet
- The network layer: data and control planes
- Router architecture

Lecture 8: Software defined networking

Lecture 9: Programmable data plane

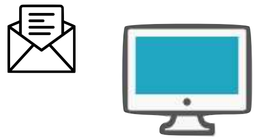
Lecture 10: Cloud networking

Lecture 11: Beyond networking

Let us start from simple

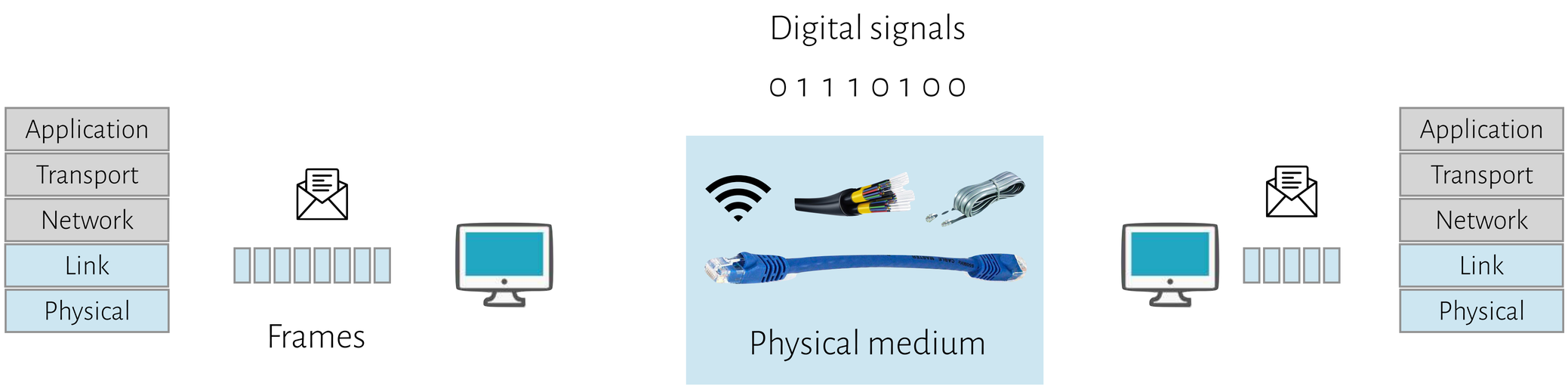
"It is a mistake to look too far ahead. Only one link in the chain of destiny can be handled at a time."

—*Winston Churchill*



How do we connect two computers?

Connecting two computers



The sending host encapsulates the packet from the network layer into frames.

The receiving host unpacks the frames and send the network packets to the upper layers.

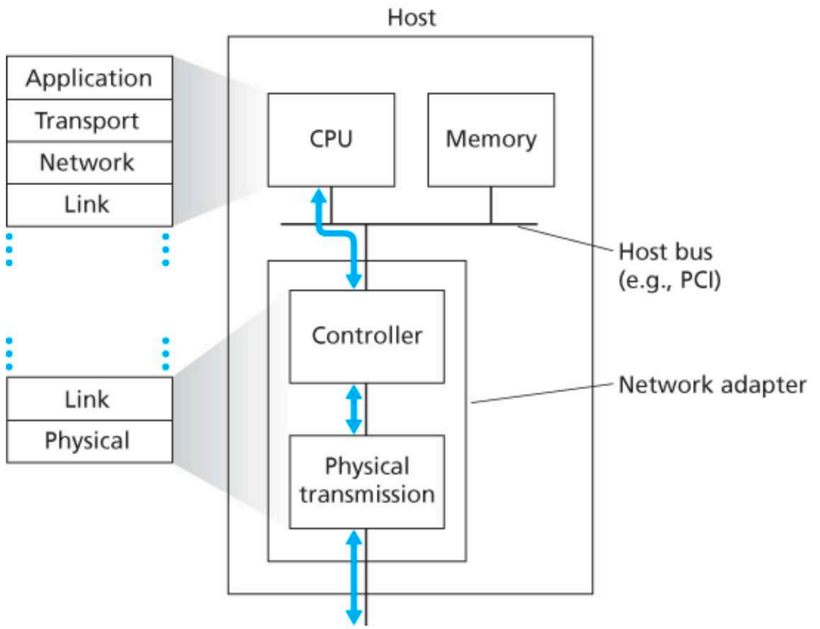
The link layer

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 (i.e., layer 2) protocol, including hosts, routers, switches, and WiFi access points.

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

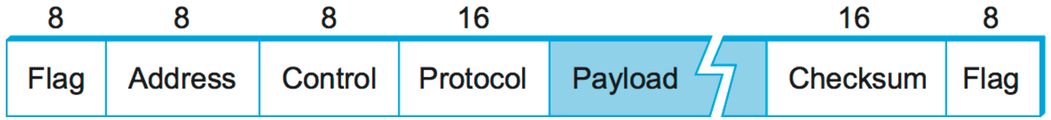


Kurose & Ross, Computer Networks:
A Top-Down Approach.

Framing: determine where the frame starts and ends

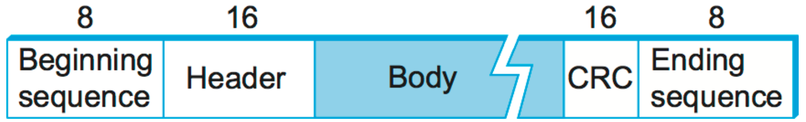
Byte-oriented protocols

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



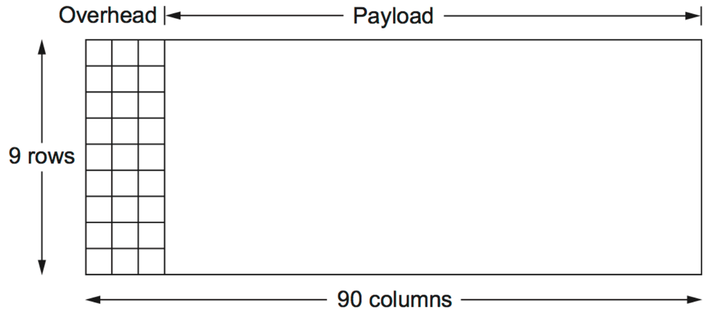
Bit-oriented protocols

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

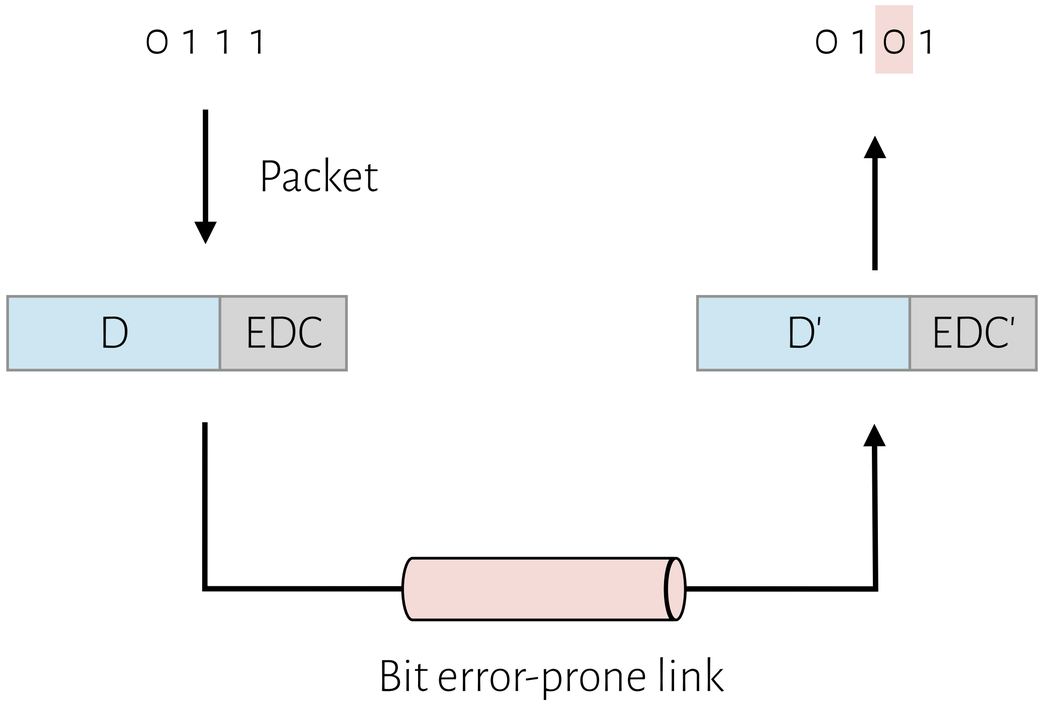


Clock-based framing

- Synchronous optical network (SONET)
- Sync use the first 2 bytes (with a pattern) in the overhead, look for pattern every 810 bytes



Error detection



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

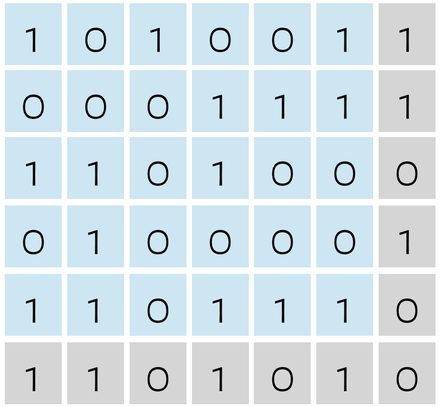
There are different ways for detecting errors:

- 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

Single bit parity checks



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

1001 1011 0100 0111
1010 1011 1101 0111
0100 0111 0001 1110
1
0100 0111 0001 1111
Checksum =
1011 1000 1110 0000

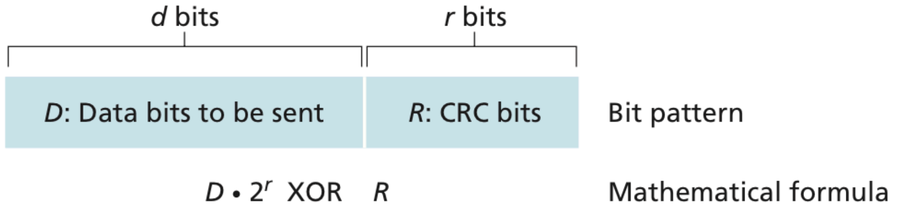
Receiver

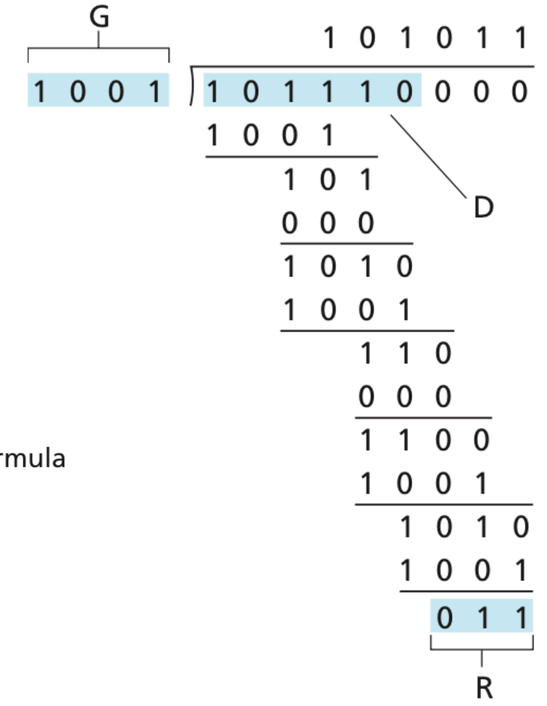
1001 1011 0100 0111
1010 1011 1101 0111
1011 1000 1110 0000
1111 1111 1111 1110
1
1111 1111 1111 1111
Checksum OK

Cyclic redundancy check (CRC)

Applies polynomial arithmetic operations on the input bit string

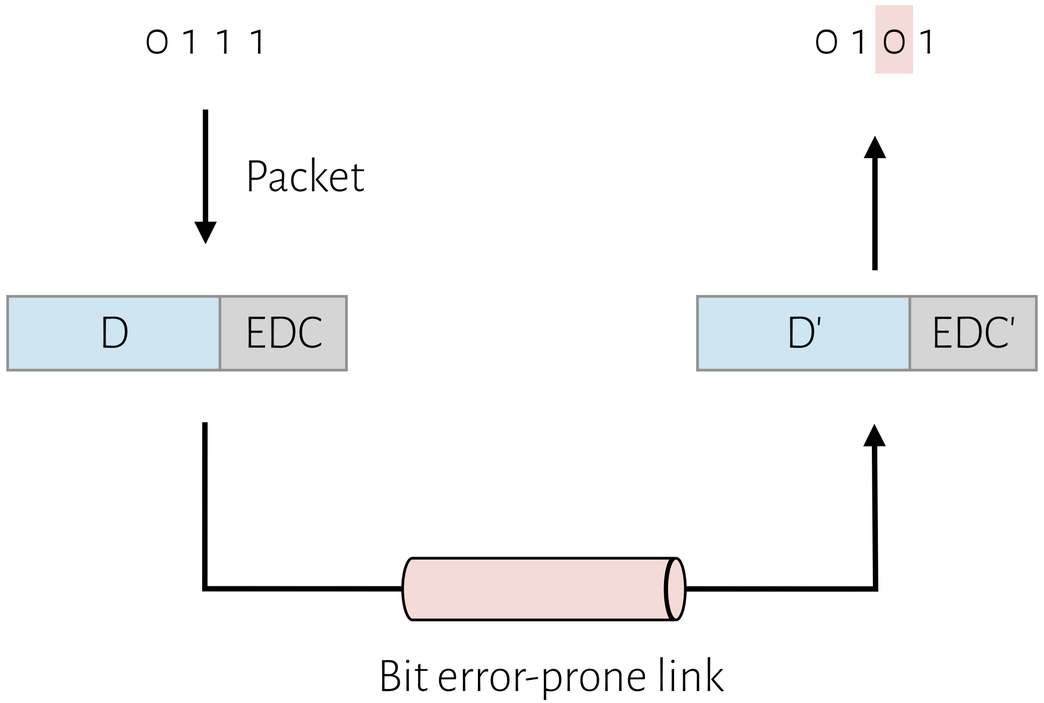
- Smaller chance of collisions, but more computation-intensive
- Adopted by the link layer and implemented in hardware NIC





Why not using MD5, SHA256, etc.?

Error detection summary



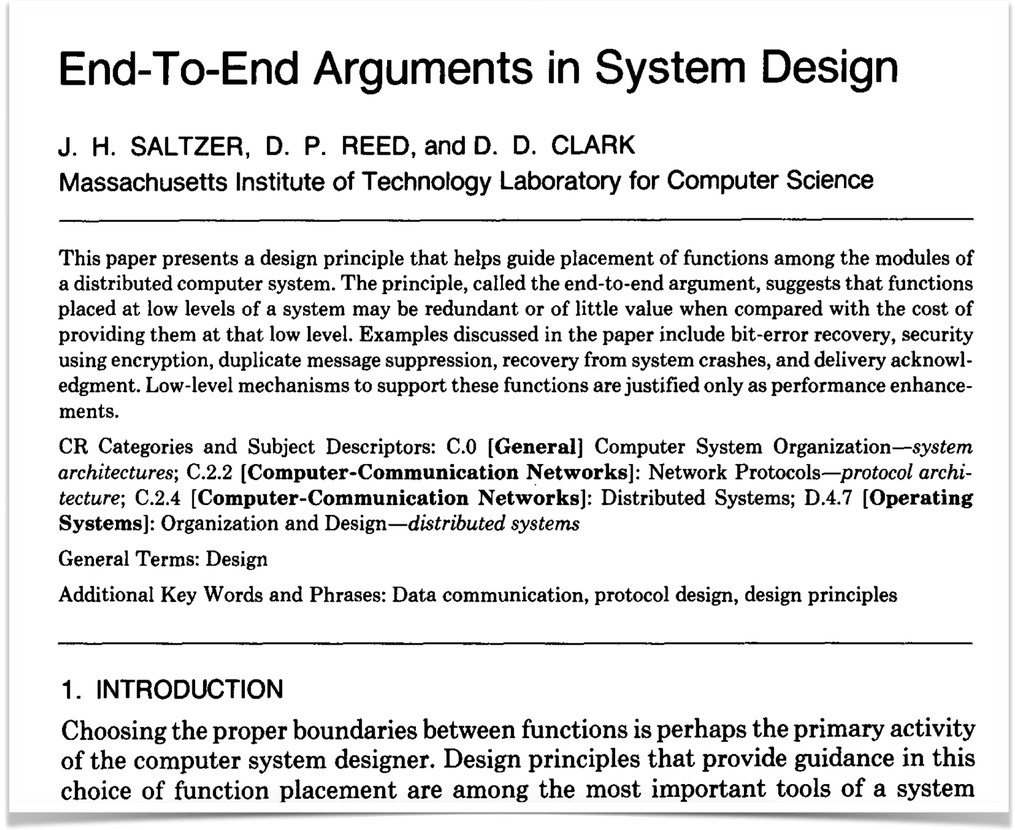
Detecting bit flips in the frame and discard the frame if errors are found

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

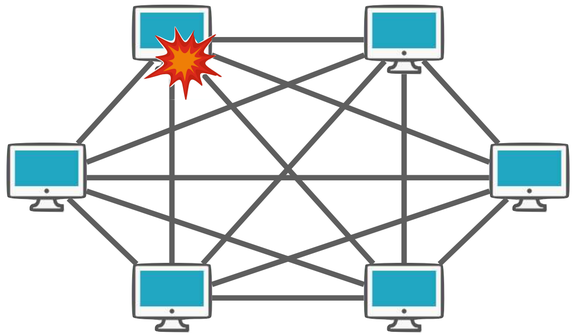
1. INTRODUCTION

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

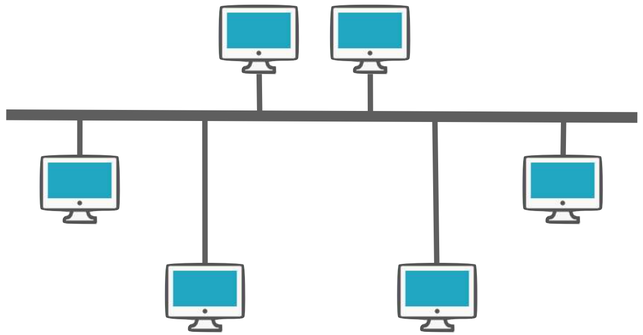
"The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the endpoints of the communication system. Therefore, providing that questioned function as a feature of the communication system itself is not possible. (Sometimes an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

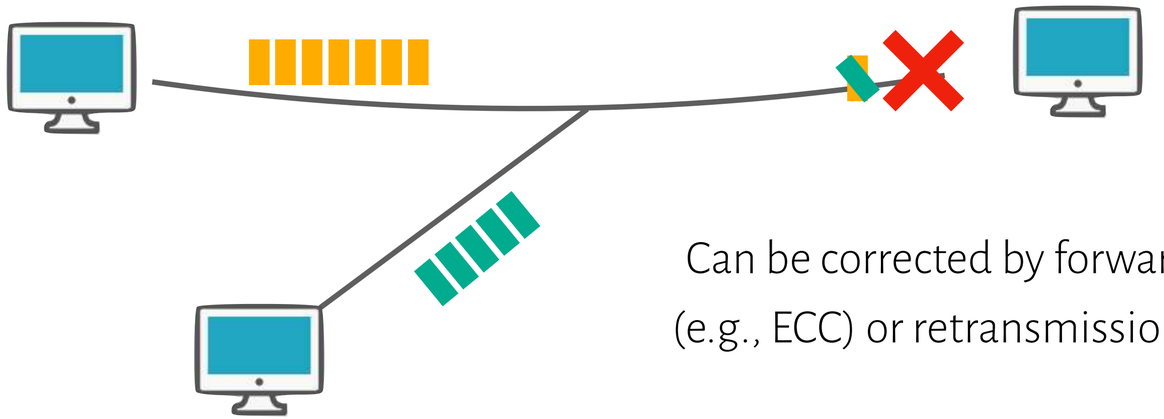


A slightly better approach: shared medium

What could be the problem?

Shared broadcast medium

Signals may collide with each other,
leading to frame corruptions

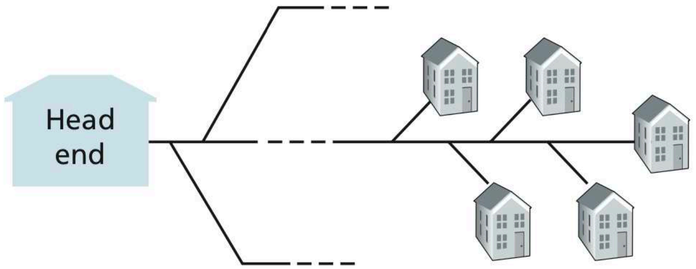


Can be corrected by forward error correction
(e.g., ECC) or retransmission, but it takes time

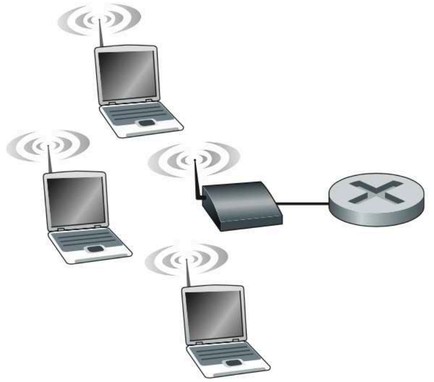
Examples: Ethernet and Wireless LAN

Shared medium examples

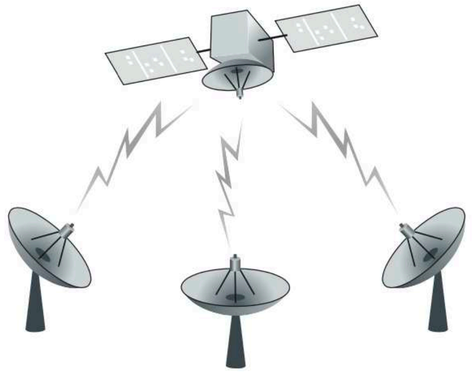
Shared wire
(for example, cable access network)



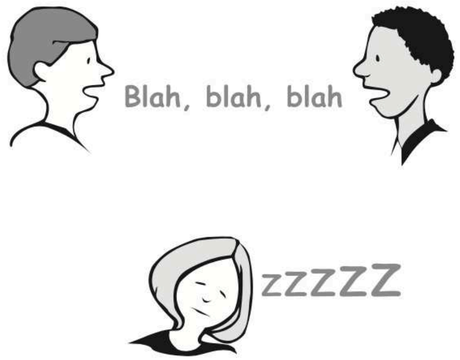
Shared wireless
(for example, WiFi)



Satellite



Cocktail party



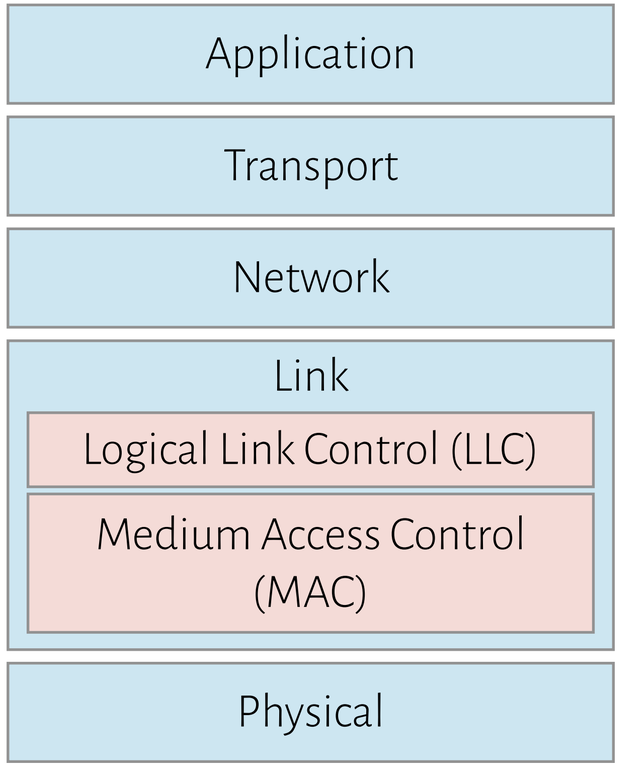
Multiple access protocol

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



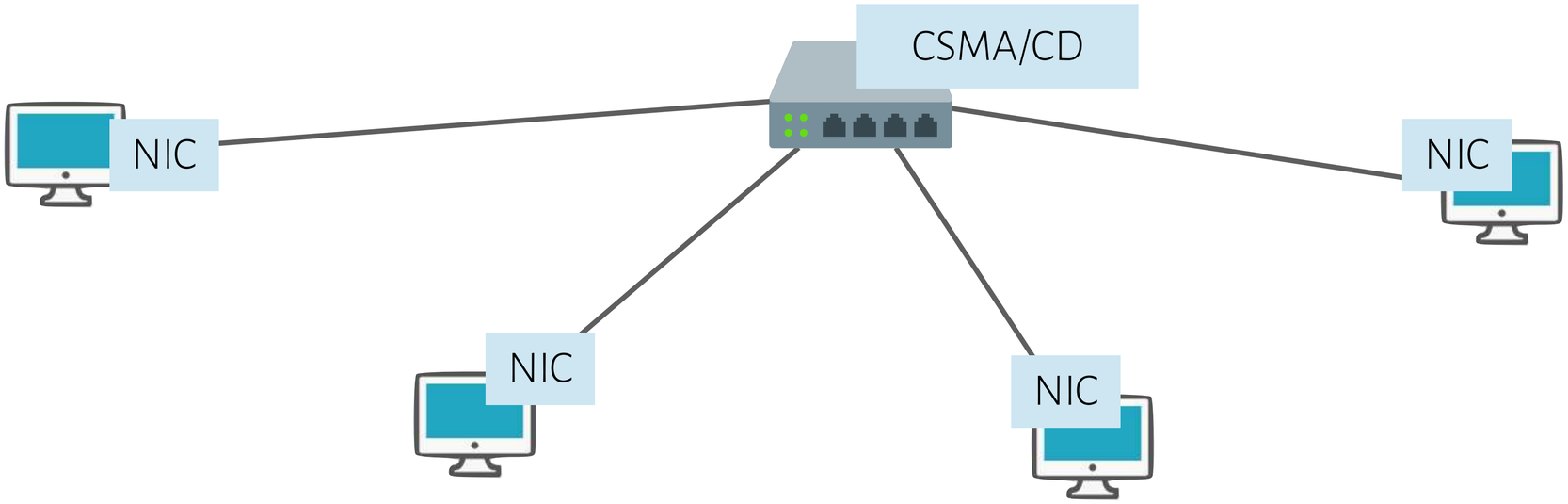
Heavy adoptions in wireless networks (WLAN, cellular, LoRa, etc.) due to the shared-medium nature

Ethernet

IEEE 802.3

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

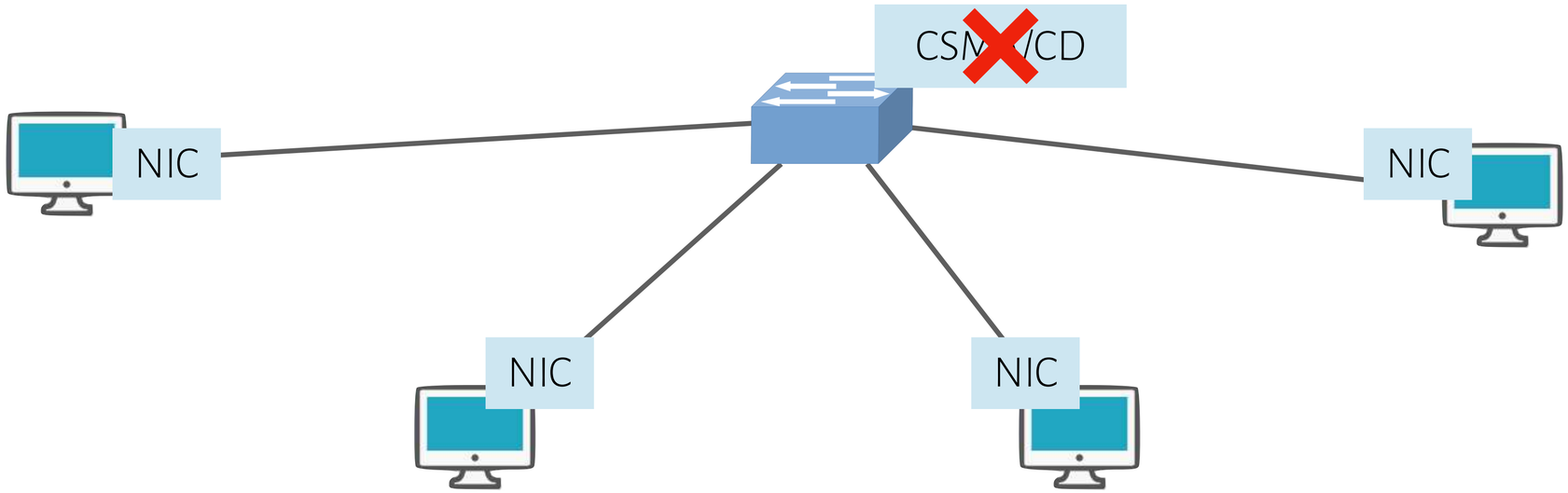
Hub: replicates signals to all ports except the one that signals are received on



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

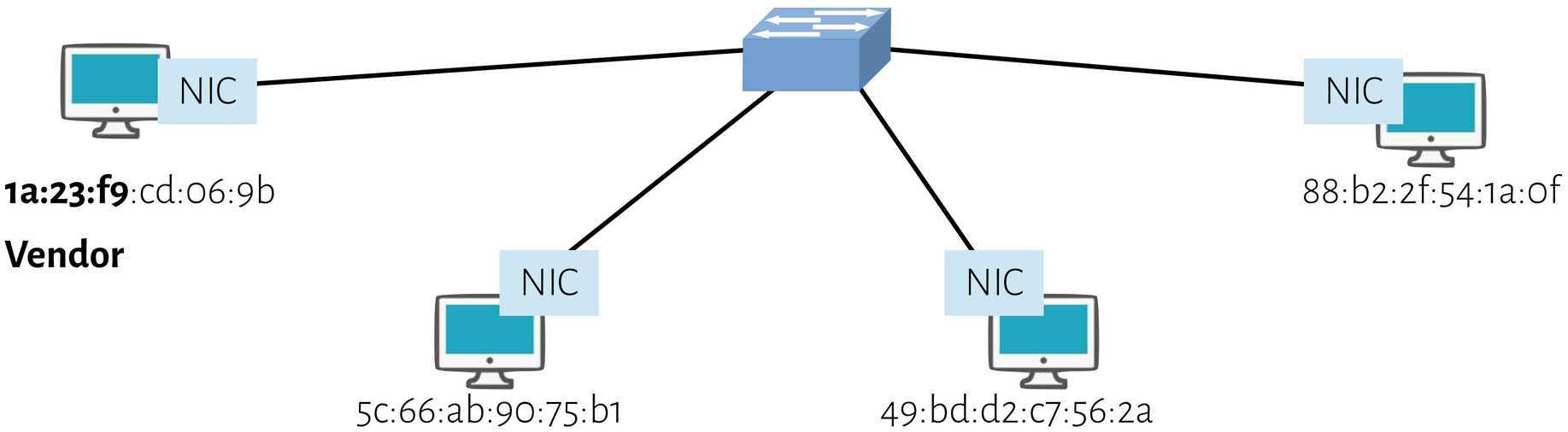


Ethernet MAC address

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

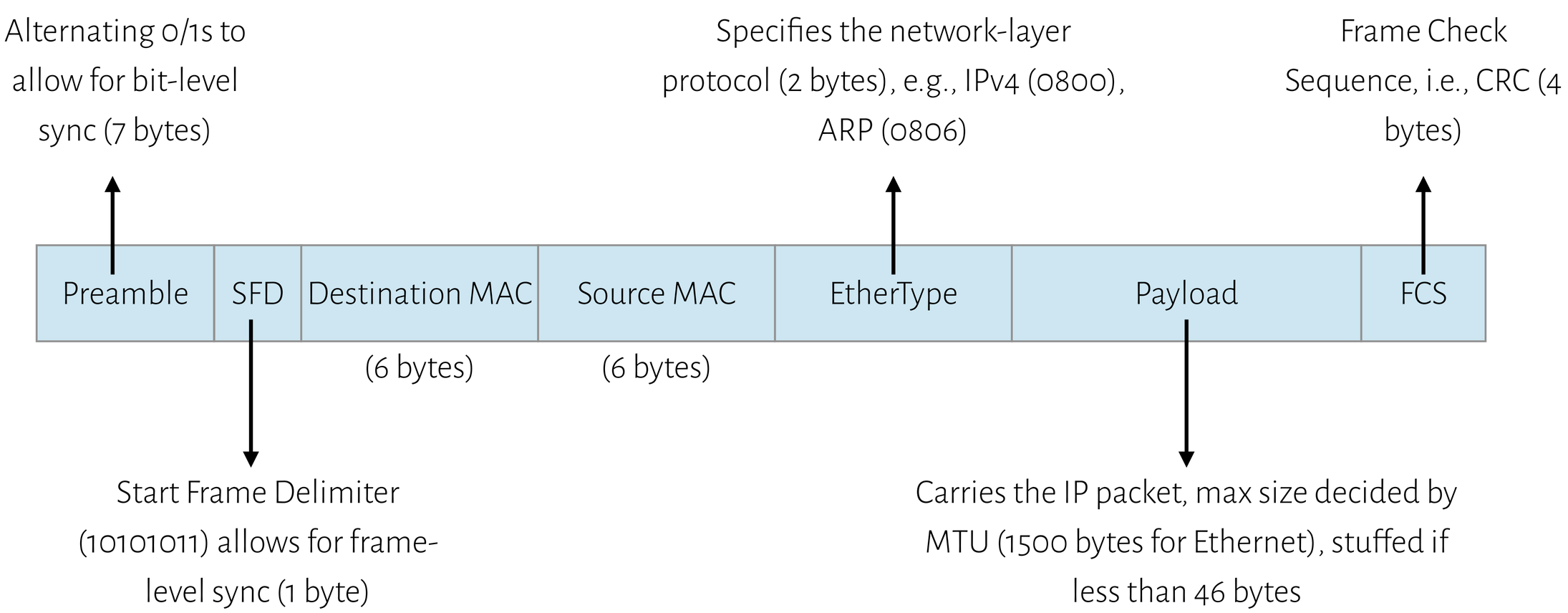
Broadcast MAC address
ff:ff:ff:ff:ff:ff

Do switches have MAC addresses? Why?



Ethernet frame structure

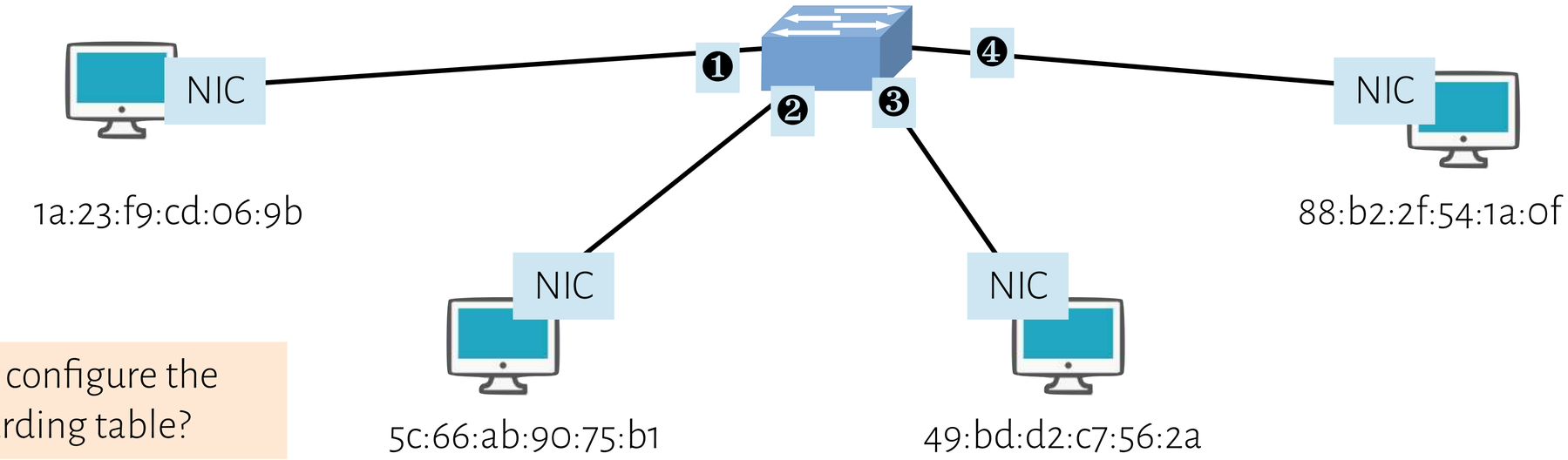
IEEE 802.3



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



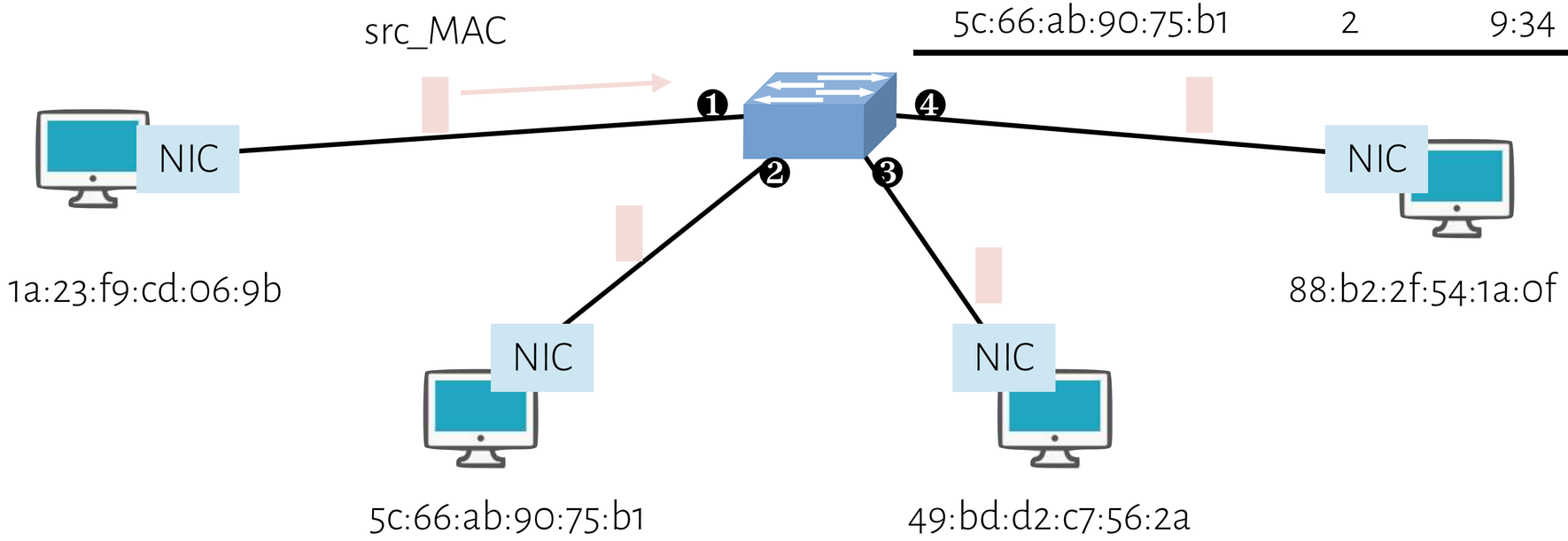
How to configure the forwarding table?

Switches are self-learning

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

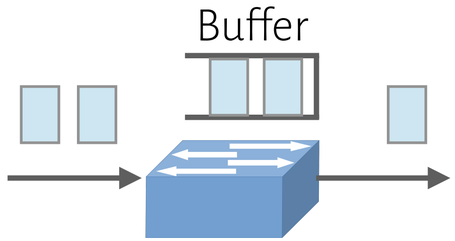
- Initially empty forwarding table
- For each incoming frame received on an interface, store the **source MAC** of the frame and map it to the **receiving interface**, with the current time
- An entry is deleted if the aging time has elapsed

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



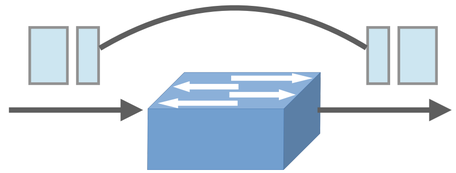
Store-and-forward vs. cut-through

Store-and-forward



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

Cut-through

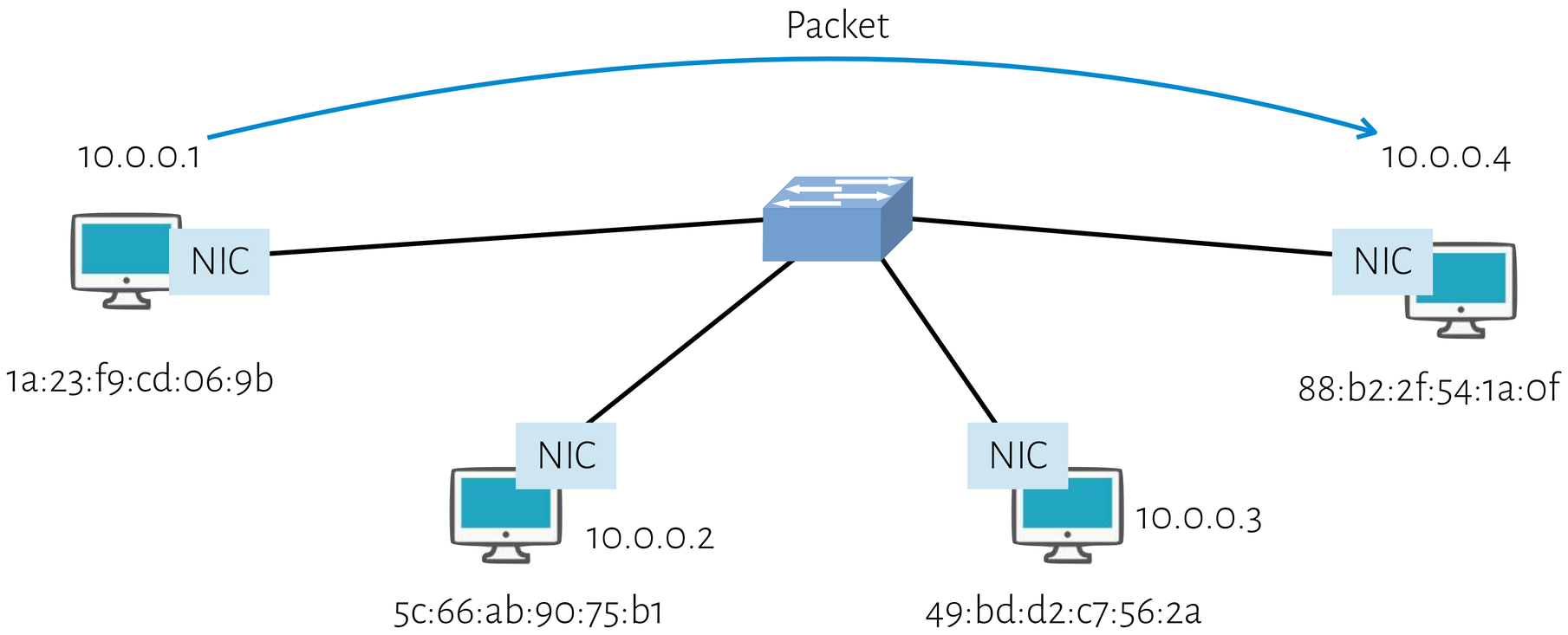


Once lookup is done, packet receiving and sending happen at the same time.

What are the pros and cons of each approach?

How to obtain destination MAC addresses?

Assume we want to send a packet from 10.0.0.1 to 10.0.0.4 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.

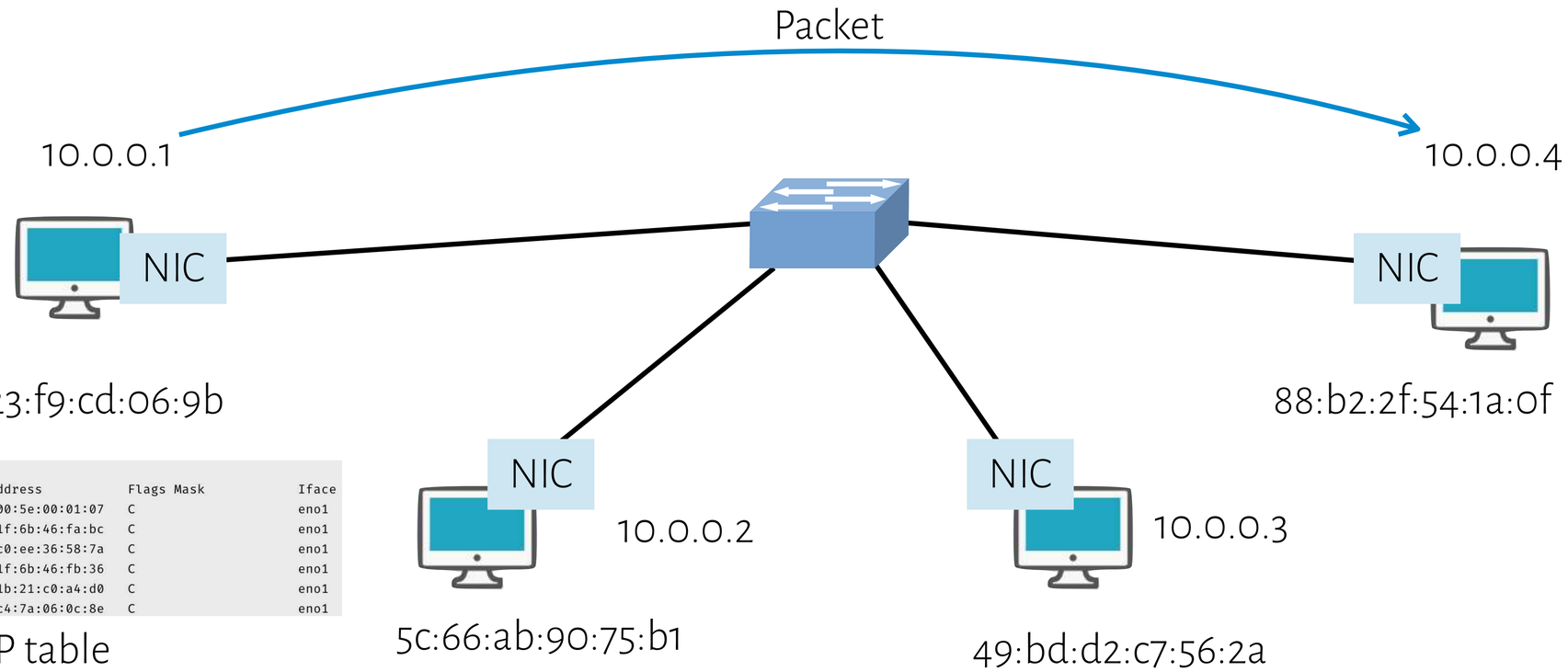


ARP protocol

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

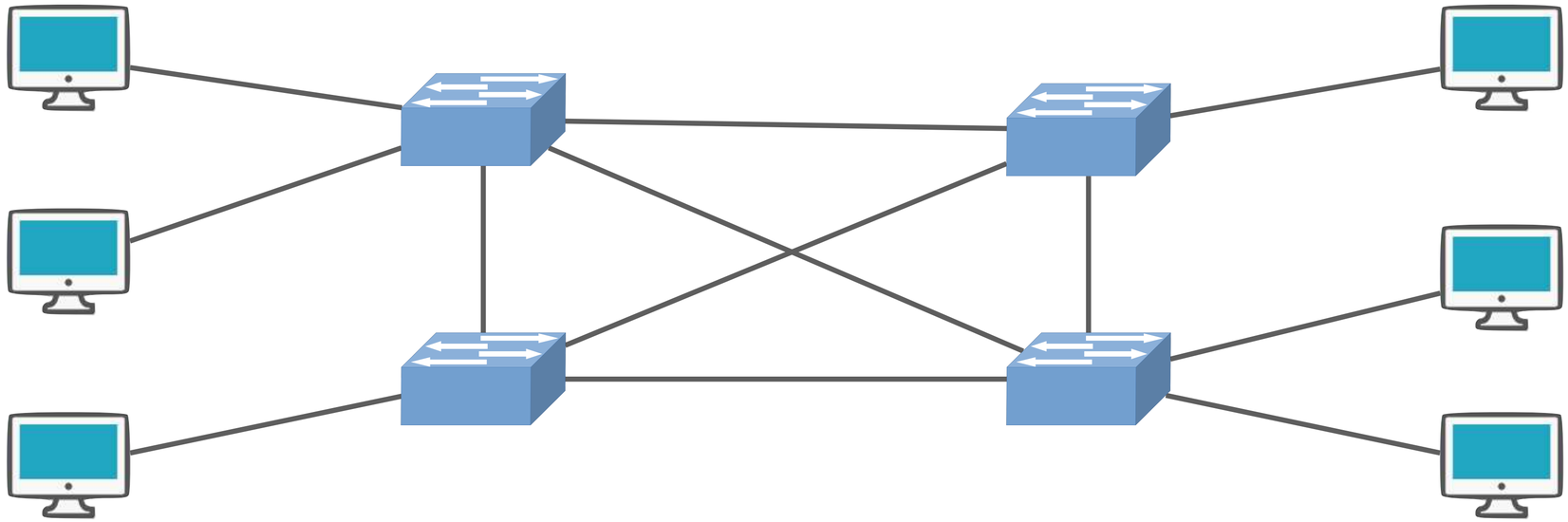



```
wang@erdos:~$ arp -n -i eno1
Address                  HWtype  HWaddress           Flags Mask            Iface
130.83.163.254           ether   00:00:5e:00:01:07   C                     eno1
130.83.163.133           ether   ac:1f:6b:46:fa:bc   C                     eno1
130.83.163.200           ether   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

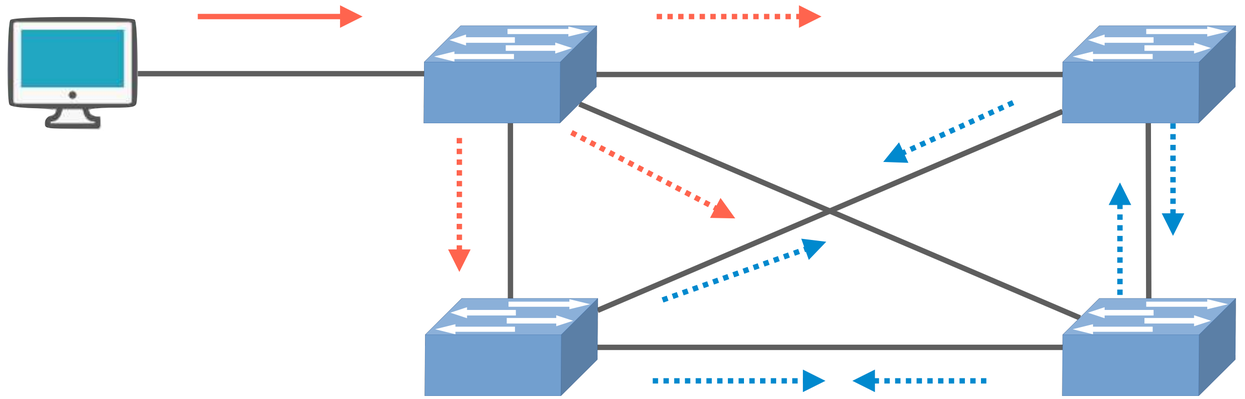
A network of switches

To connect a large number of nodes, we can simply put in more switches and connect them



What are the problems with a switched Ethernet?

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)**

Algorhyme

I think that I shall never see a graph
more lovely than a tree.
A tree whose crucial property is loop-
free connectivity.
A tree that must be sure to span so
packets can reach every LAN.
First, the root must be selected.
By ID, it is elected.
Least-cost paths from root are traced.
In the tree, these paths are placed.
A mesh is made by folks like me, then
bridges find a spanning tree.

— *Radia Perlman*

STP in a nutshell

Switches

- elect a root switch, the one with the smallest identifier
- determine if each interface is on the shortest-path from the root and disable it if not

Bridge Protocol Data Unit (BPDU) messages

Considered root switch ID	#hops to reach it	Switch ID

STP step by step

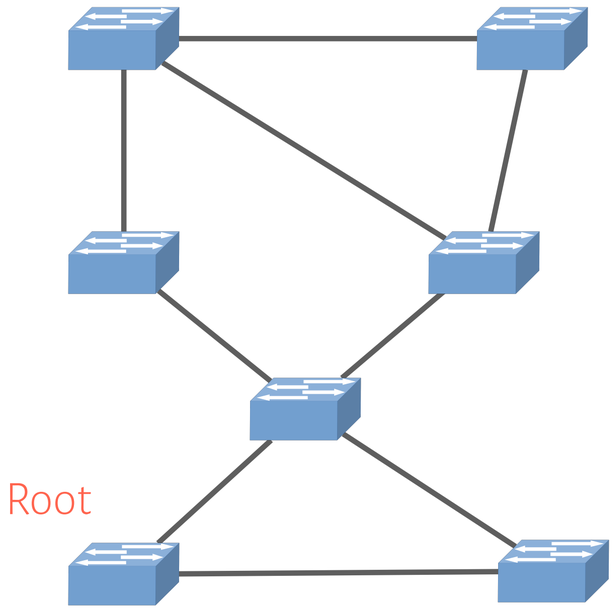
Initially

- Each switch proposes itself as root, i.e., sends (X, 0, X) on all its interface
- Upon receiving (Y, d, X), each switch checks if Y is a better root. If so, it considers Y as the new root, and floods updated message
- Switches compute their distance to the root, for each port: simply add 1 to the distance received; if shorter, flood
- Switches disable interfaces not on shortest path

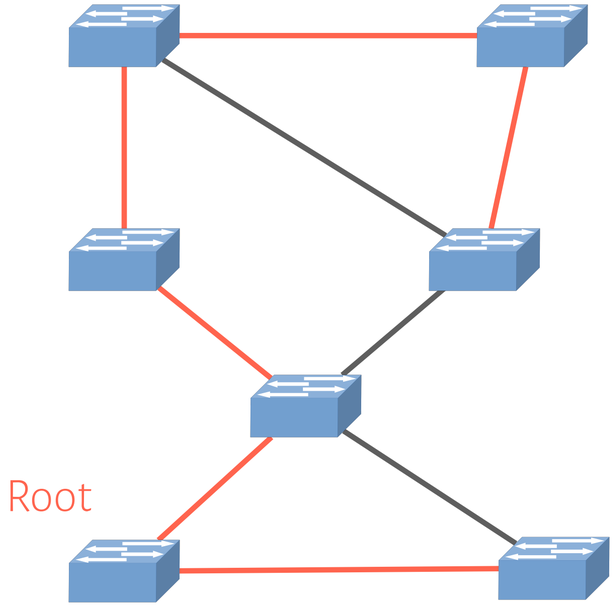
Tie-breaking

- Upon receiving different BPDUs from different switches with equal cost, pick the BPDU with the lower switch sender ID
- Upon receiving different BPDUs from a neighboring switch, pick the one with lowest port ID

STP example



Select the root



Keep shortest paths to root

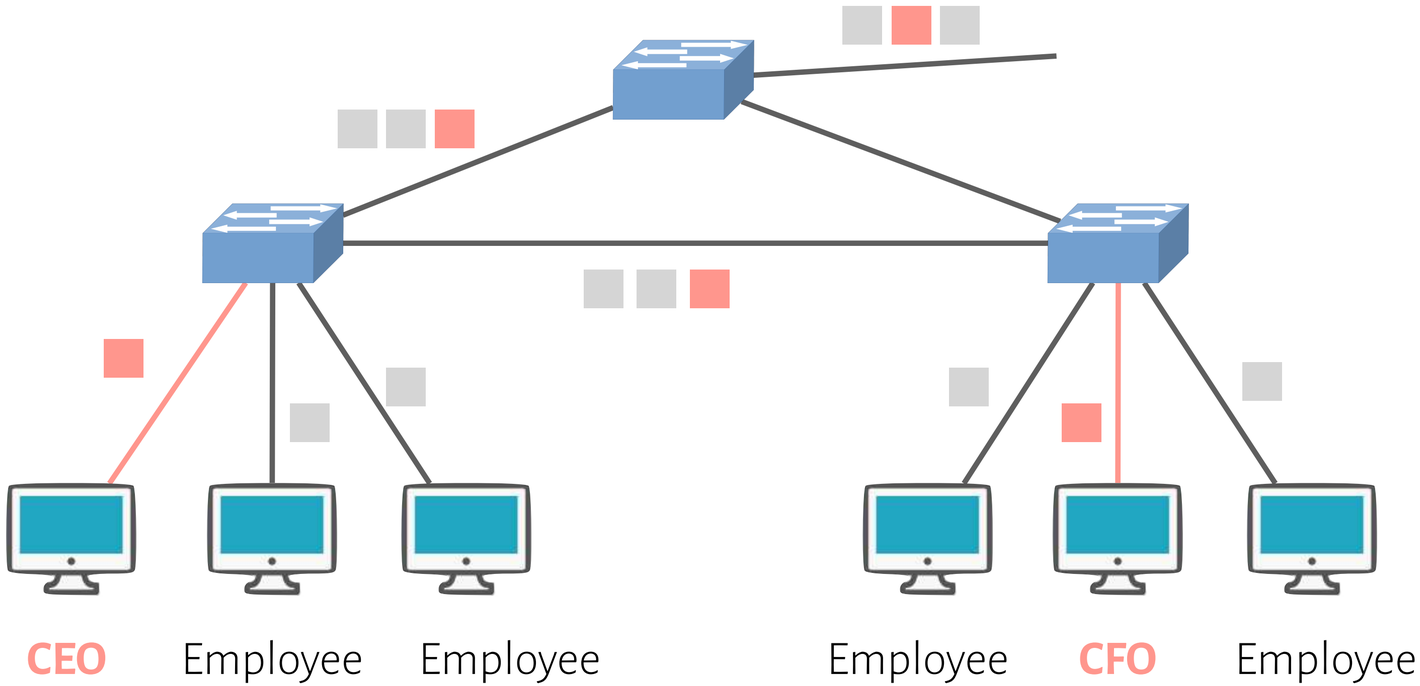
To ensure robustness, the root switch keeps sending the messages.

If timeout, claim itself to be root.

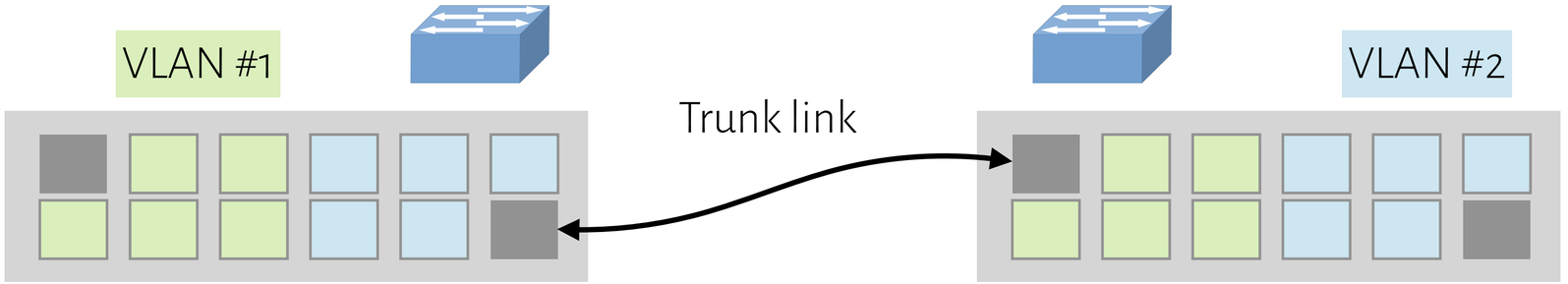
Problem #2: traffic isolation

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



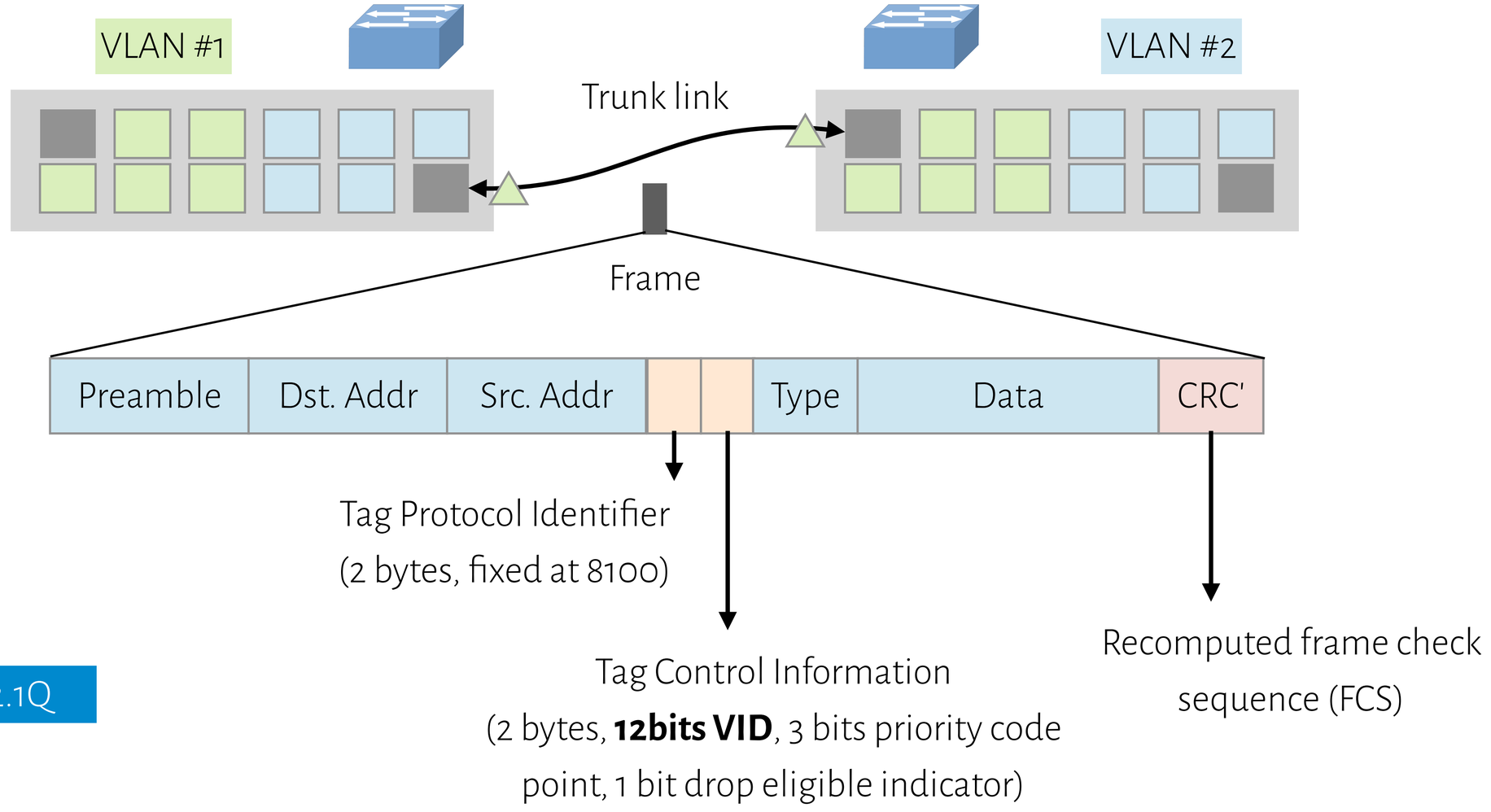
Network manager can partition the ports into subsets and assign them to VLANs

Ports in the same VLAN form a broadcast domain, while ports on different VLANs are routed through an internal router within the switch

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

Problems with switched Ethernet summary

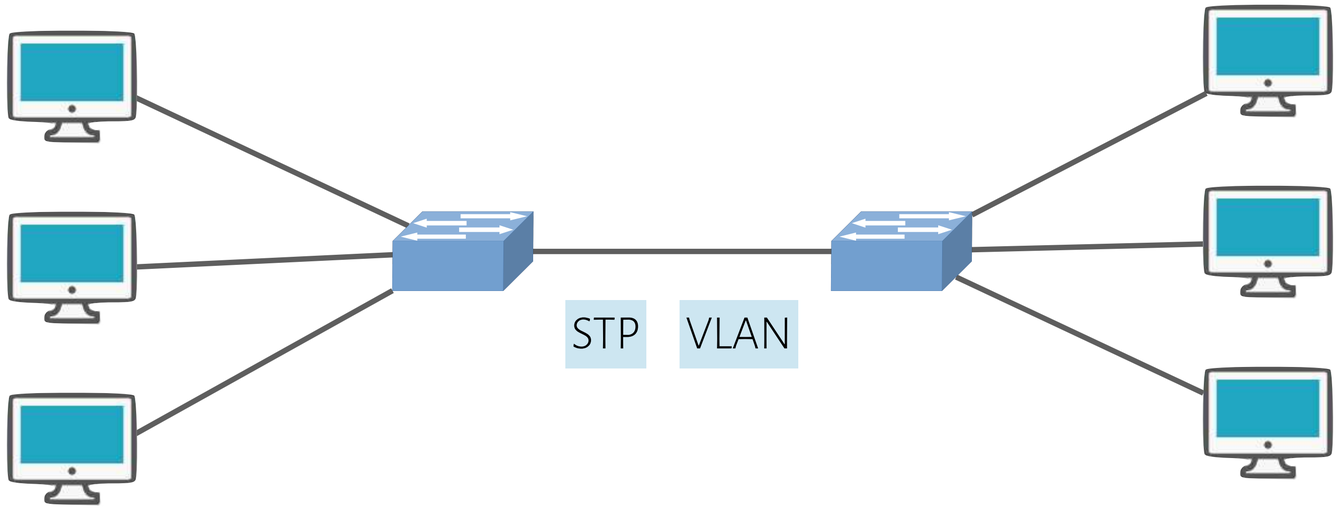
There could be forwarding **loops** in the network

- Packets that do not match on the forwarding table will be broadcasted. If there are cycles in the network, then the packet will loop and will never stop.
- Solution: **Spanning Tree Protocol (STP)**

Lack of 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
- Solution: **Virtual LAN (VLAN)**

How about a "switched Internet"?



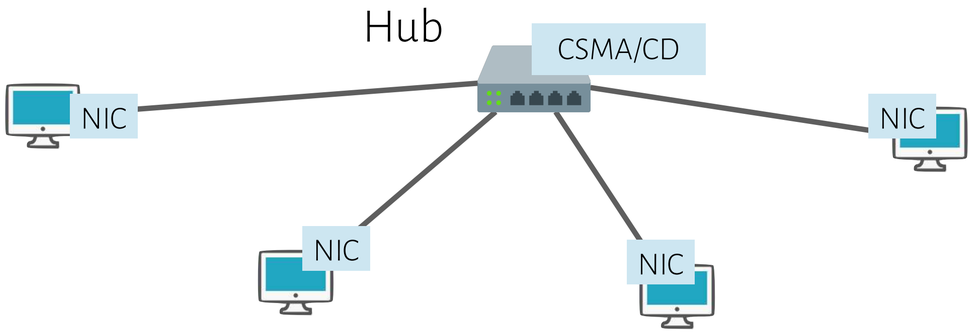
Can we simply build the global Internet with tons of interconnected switches? Why?

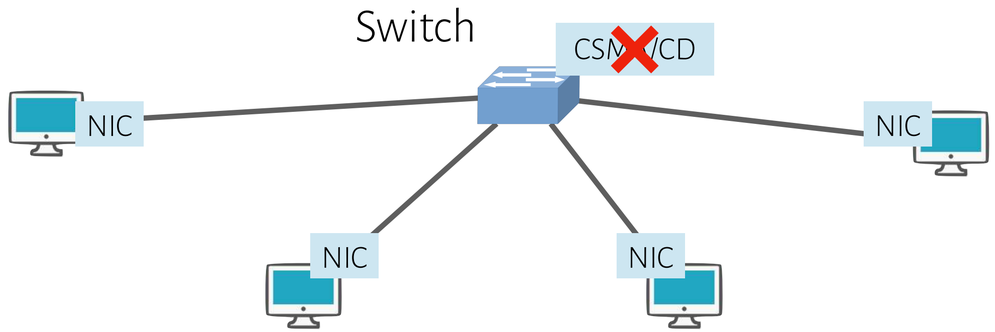
Switched Internet has many problems

We will discuss more in the context of cloud networking

Why?

- Broadcast storm: ARP requests, MAC addresses that have not been learned
- Limited switch forwarding table: a limited number of entries can be cached
- Limited isolation with VLAN: max. 4094 VLANs are possible (12 bits)
- Security issues: packet sniffing



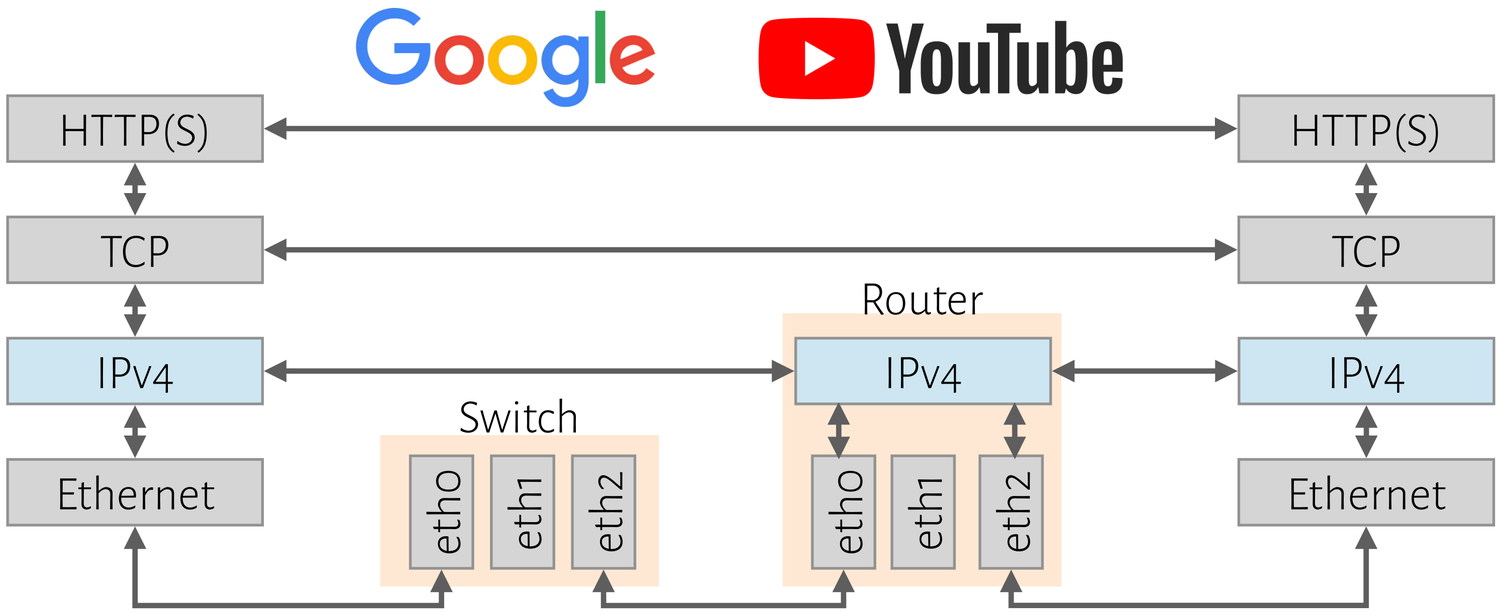


Anyone on the subnet can **sniff** the packets that are intended for others.

Can we also sniff? Why?

Questions

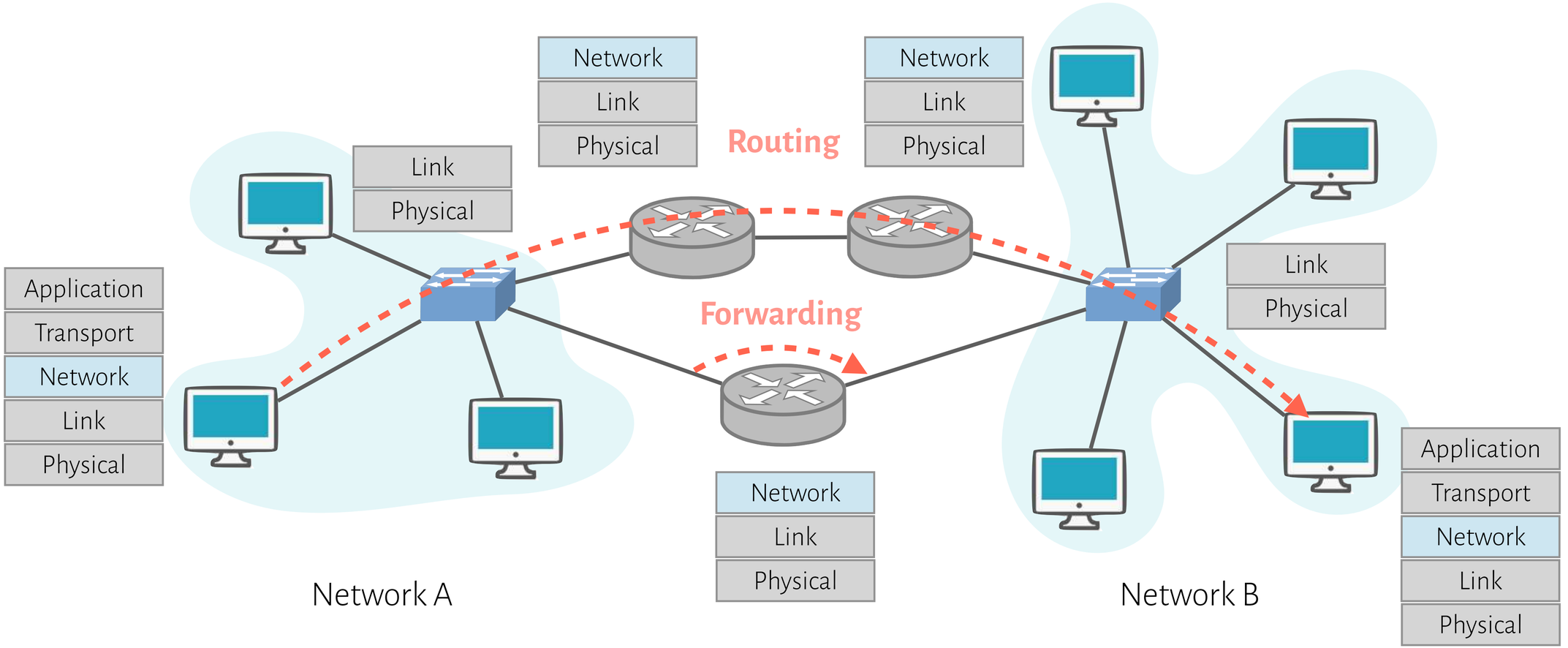
The network layer



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

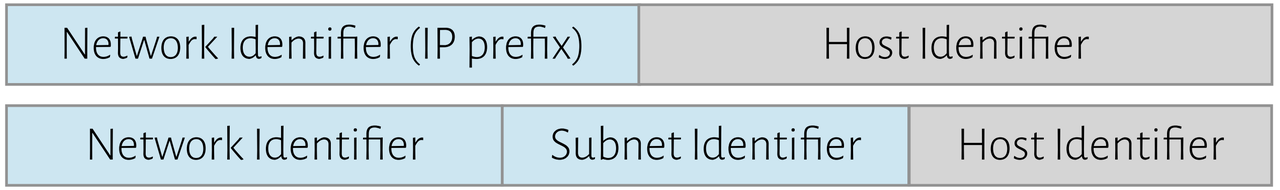
Forwarding and routing



Network layer address: example IPv4

RFC7020

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

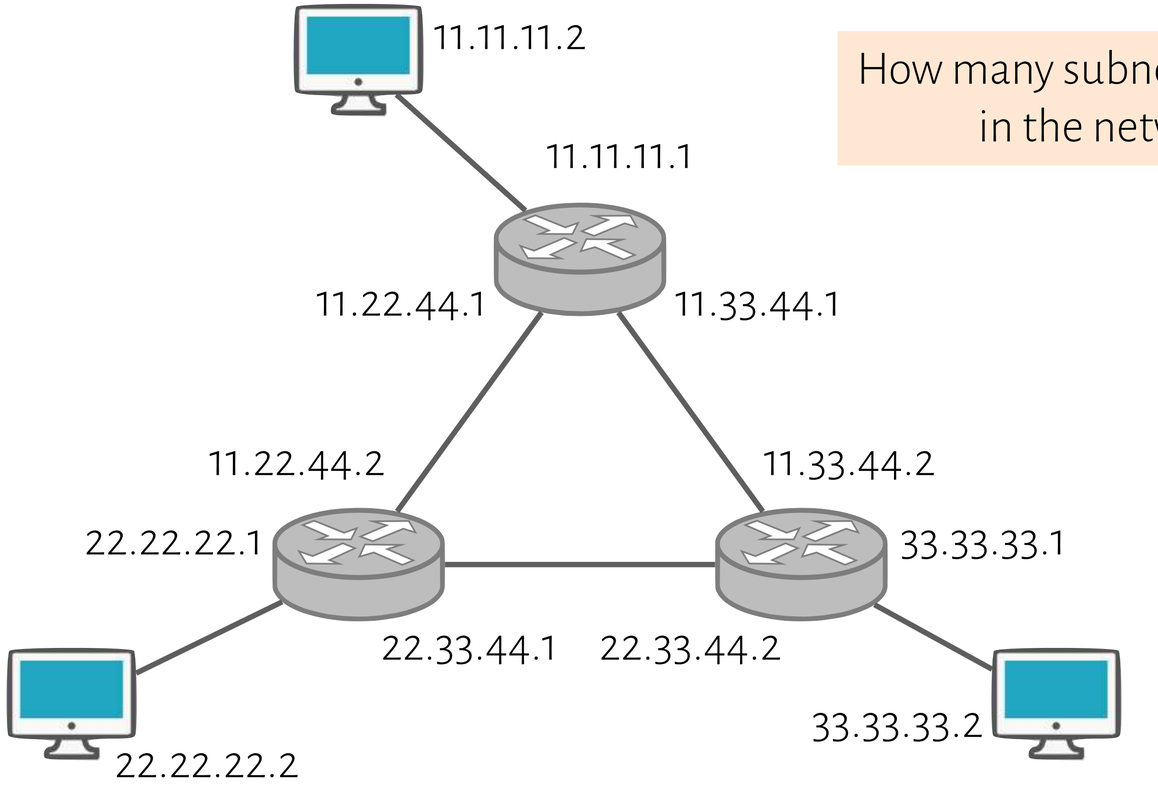


Classless Inter-Domain Routing (CIDR) notation: 10.0.0.1/24

Subnet mask notation: 255.255.255.0

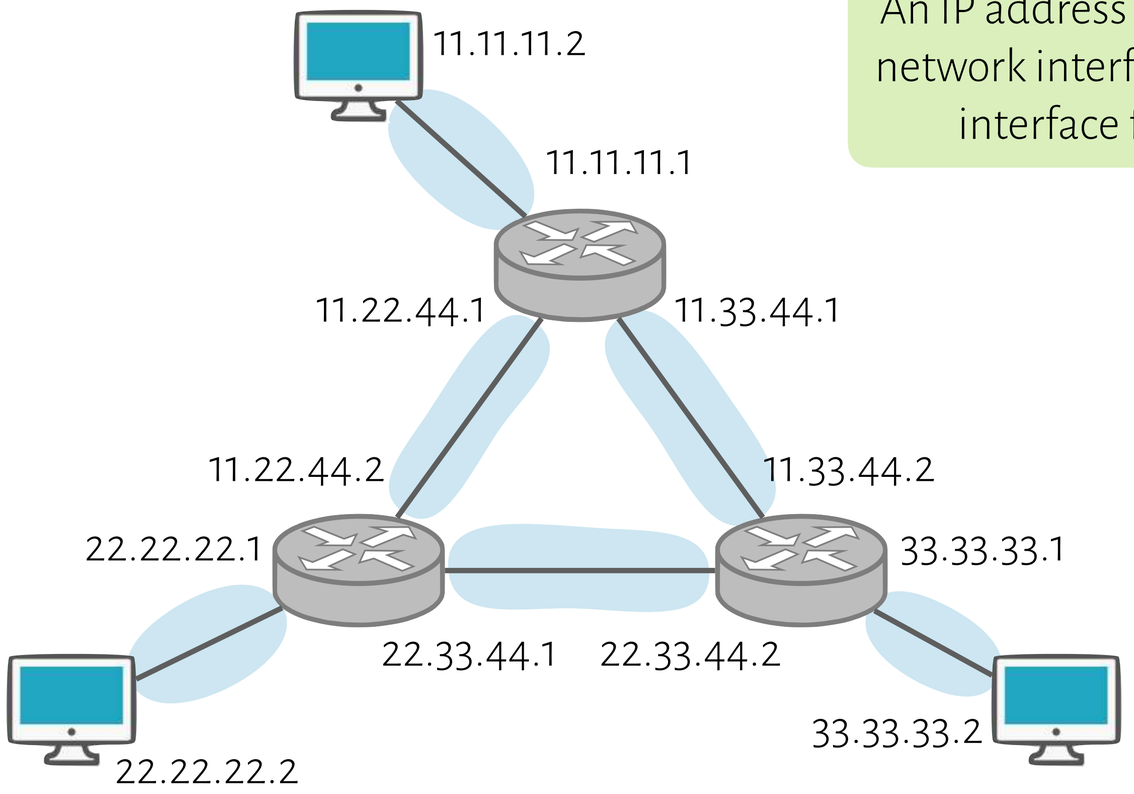
Who do we assign IP addresses to? A host? switch? router? or...

Routers interconnecting subnets



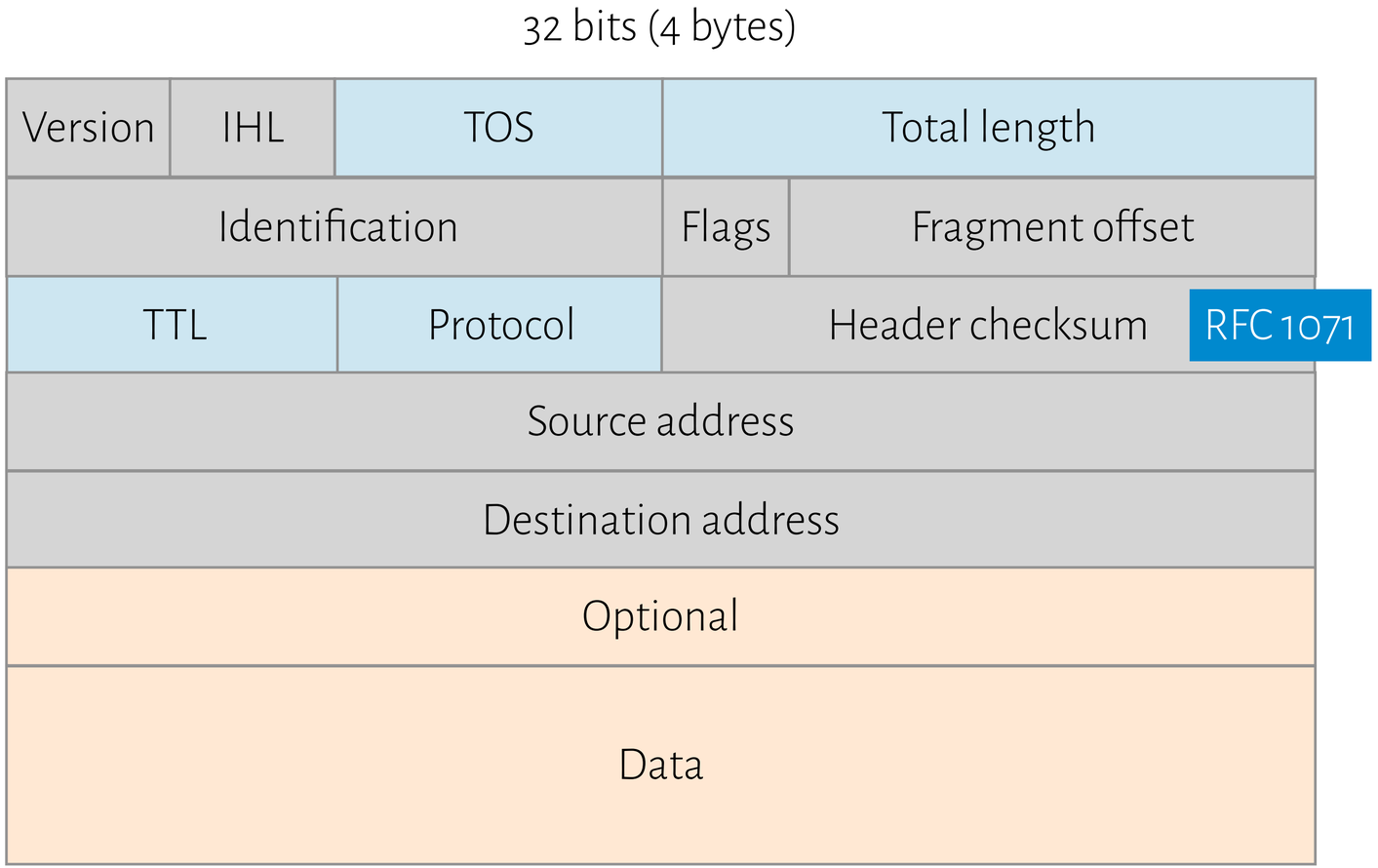
How many subnets are there in the network?

Routers interconnecting subnets



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

IPv4 packet format



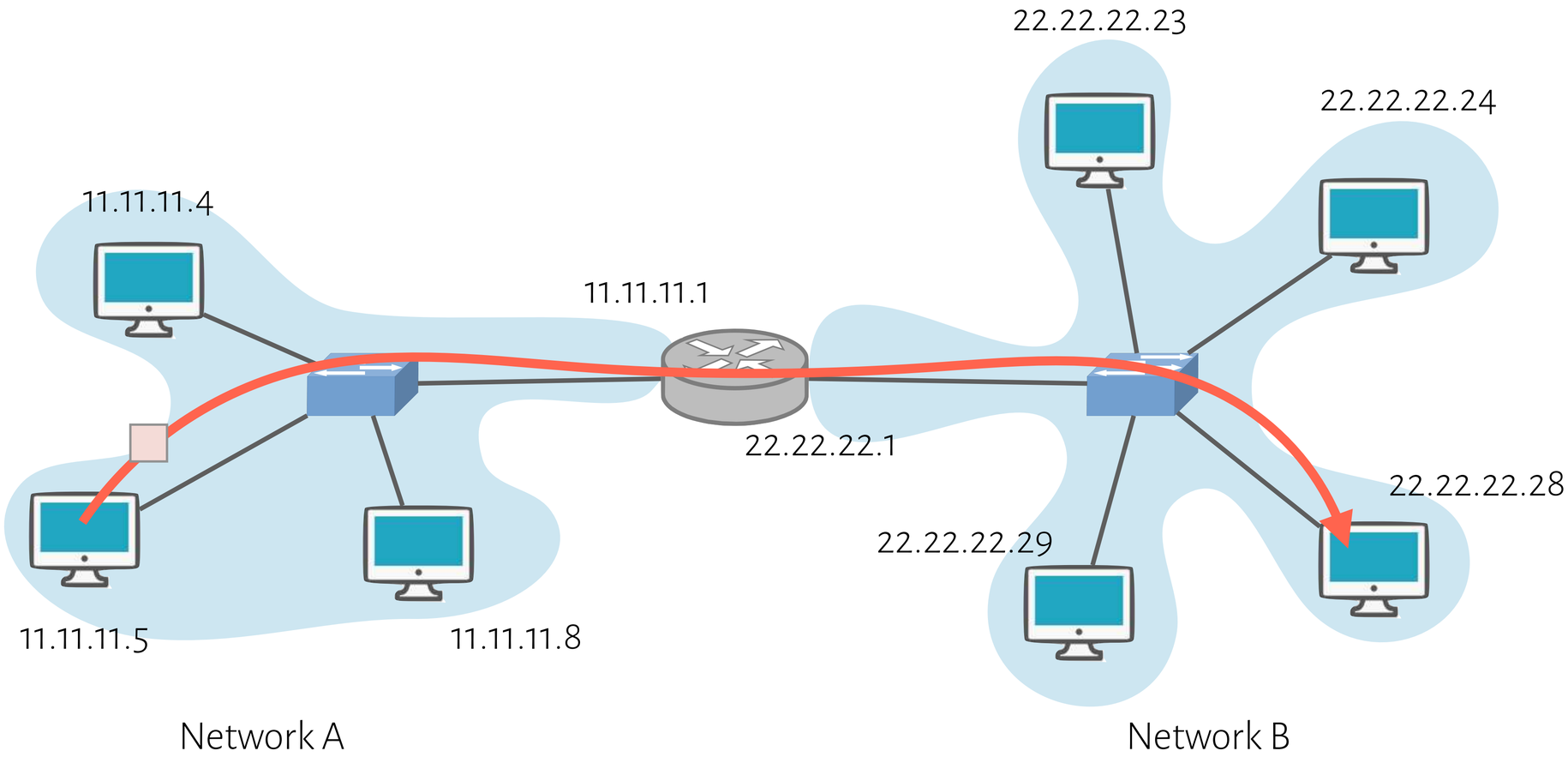
TOS: type of service, two bits used for Explicit Congestion Notification RFC 3168

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

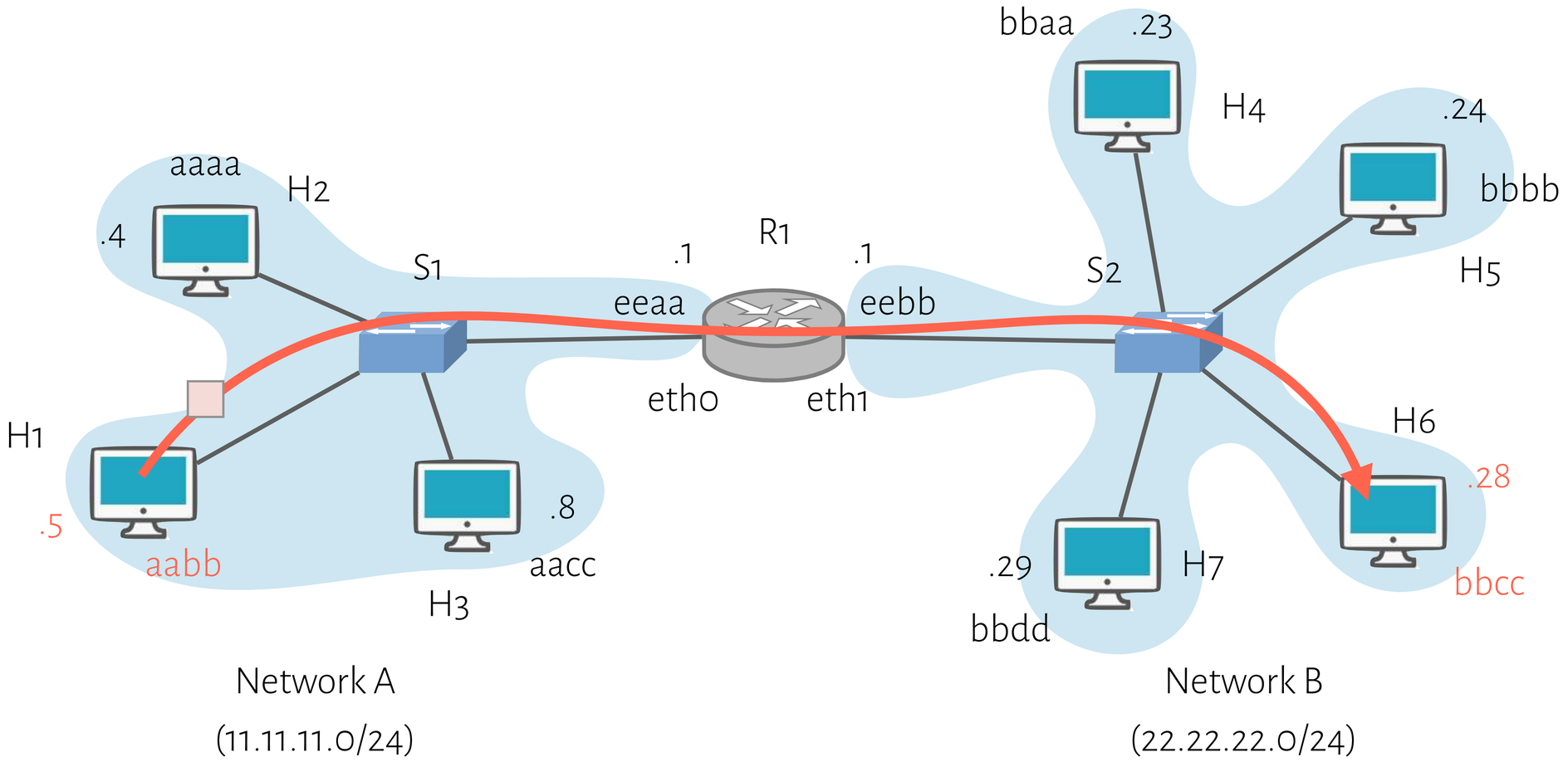
TTL: decreased by one when passing a router, packet dropped by the router when it reaches 0

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

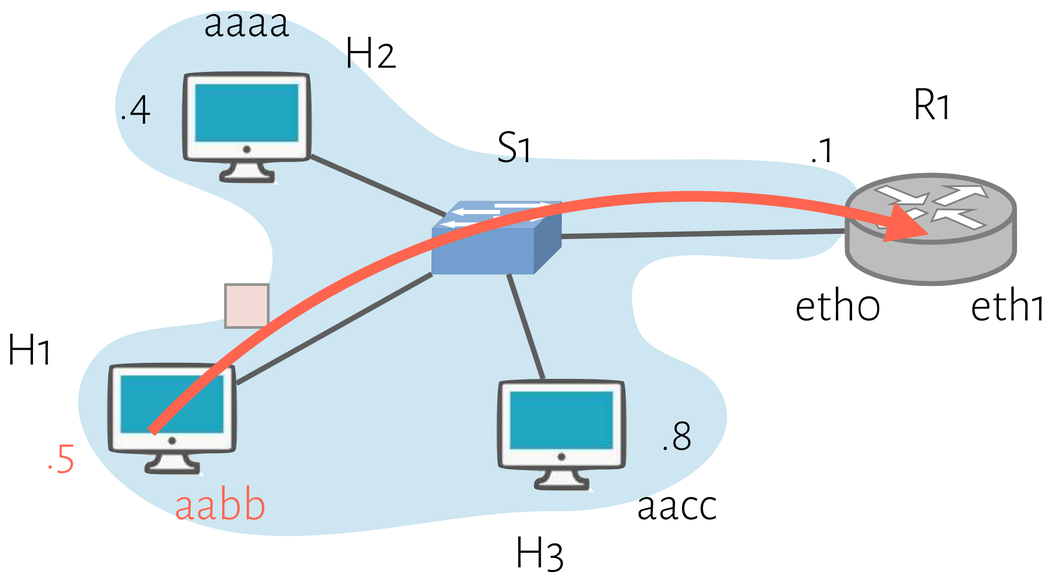
Routing between (IP) networks/subnets



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

What protocol could have been involved?

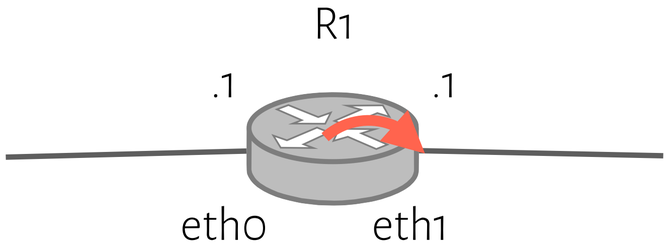
Journey of a packet

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

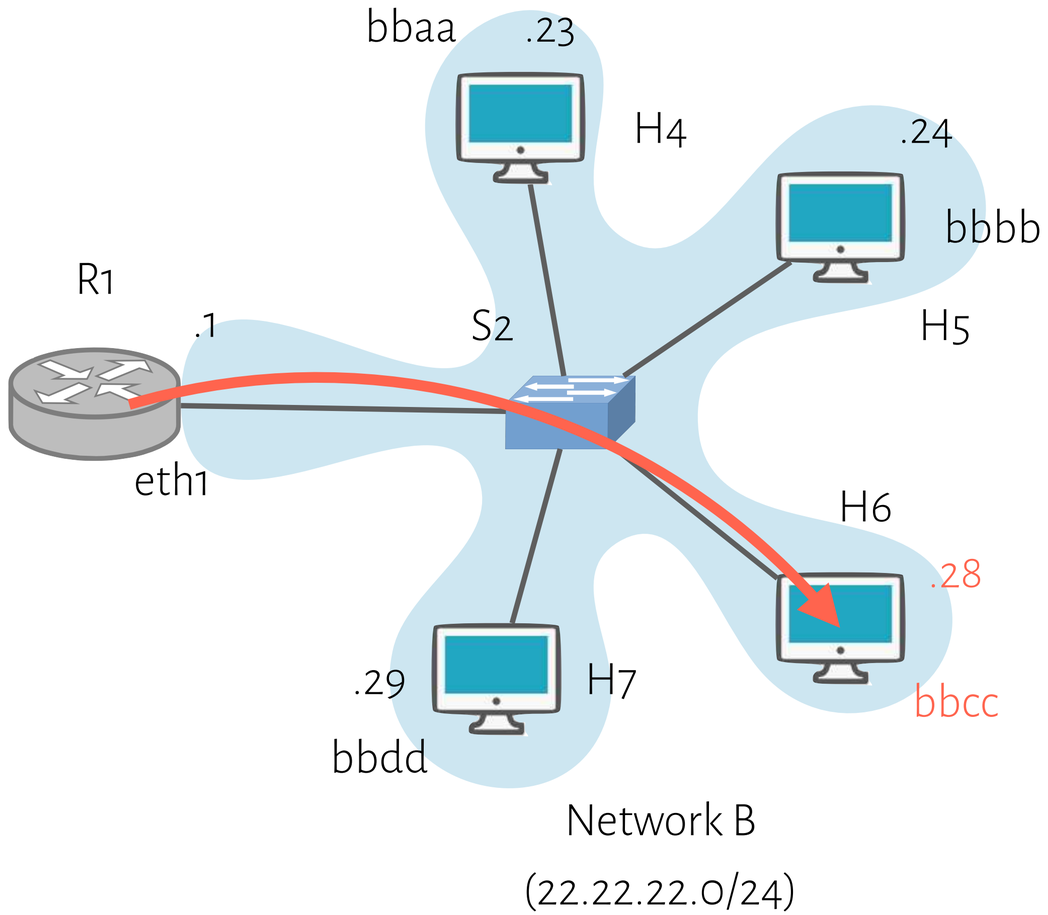
Forwarding table

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



We will discuss the internals of a router soon.

Journey of a packet



On R1

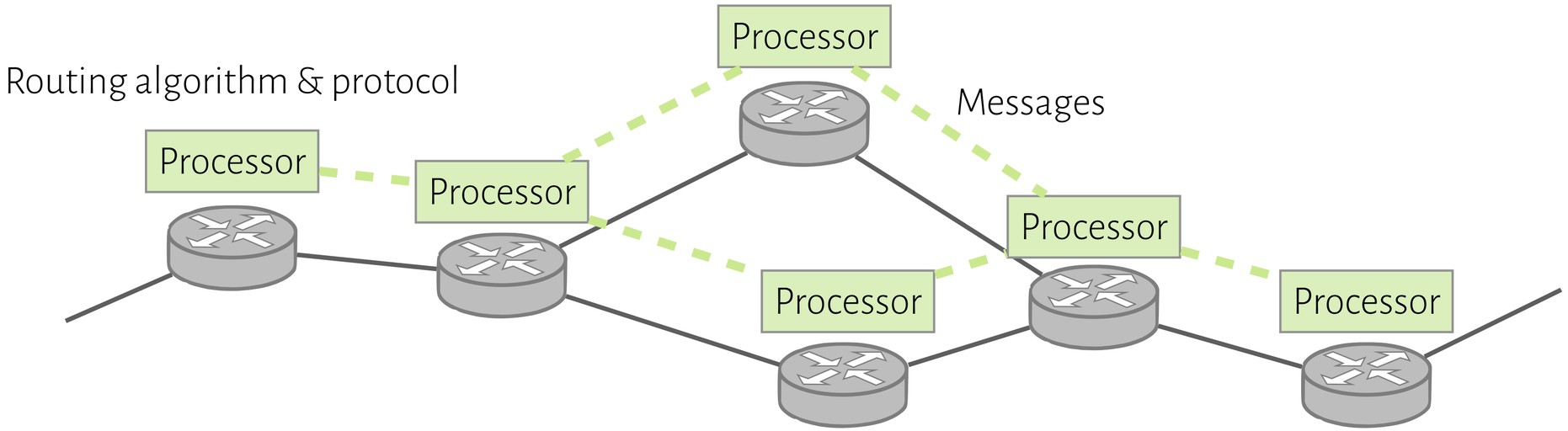
- Obtain the destination MAC of host H6
- Send a frame containing the packet with the MAC of H6 as destination MAC

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)

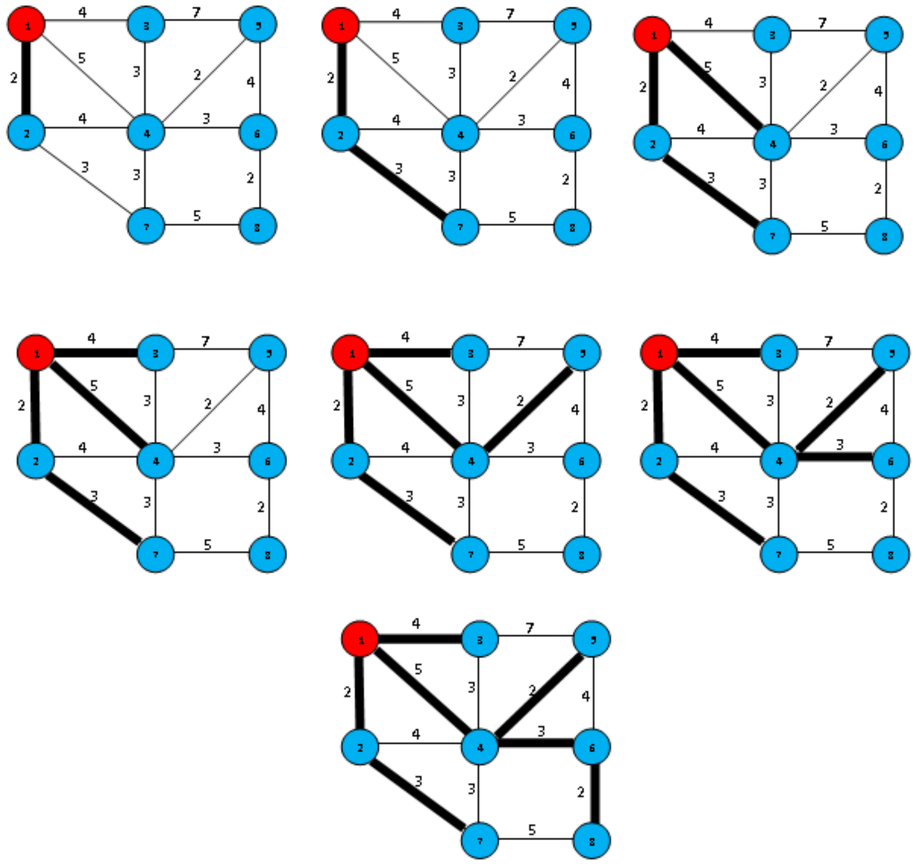


We will discuss a completely different way next time!

Routing protocol

Open Shortest Path First (OSPF):

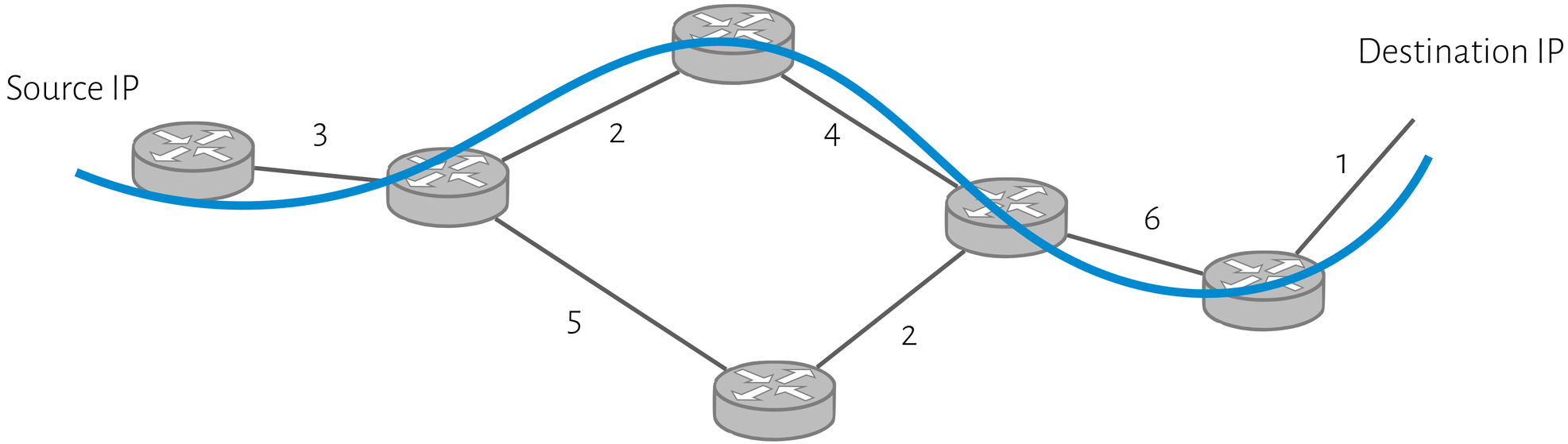
- Routers exchange link-state messages to learn the topology
- Each router runs the **Dijkstra's algorithm** to computer the shortest paths to other routers
- Each router generates the forwarding table entries based on the shortest paths



Traffic engineering

RFC 3272 RFC 2702

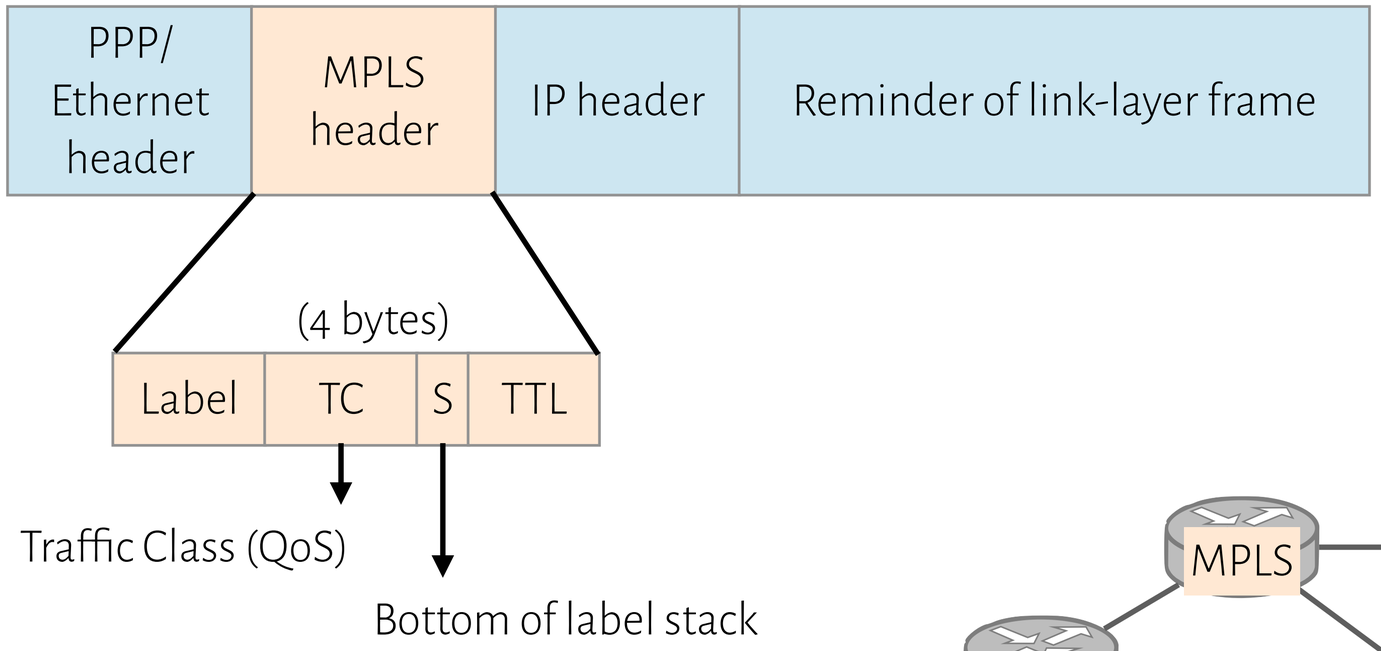
The aspect of network engineering that deals with the issue of performance evaluation and performance optimization of operational IP networks. Traffic engineering encompasses the application of technology and scientific principles to the **measurement**, **characterization**, **modeling**, and **control** of Internet traffic.

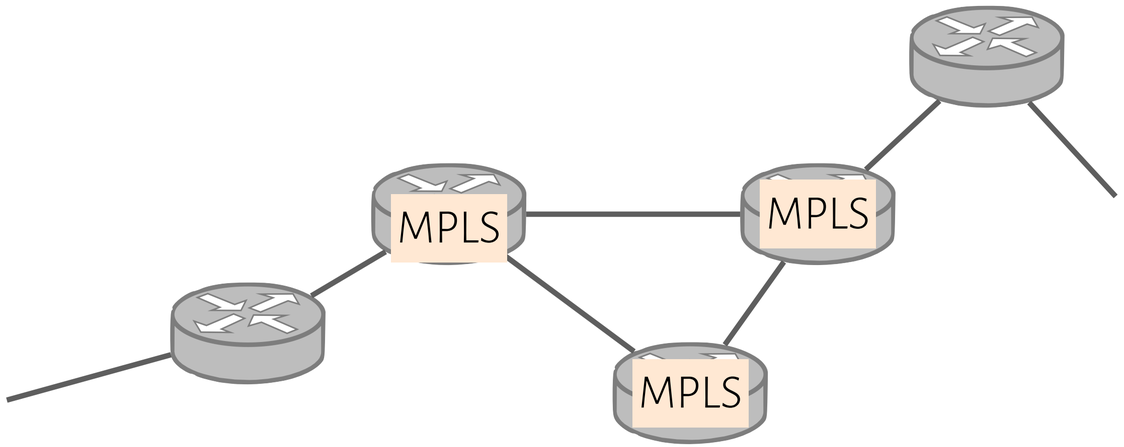


What performance issues can you foresee in network routing?

Multiprotocol label switching (MPLS)

RFC 3031 RFC 3032





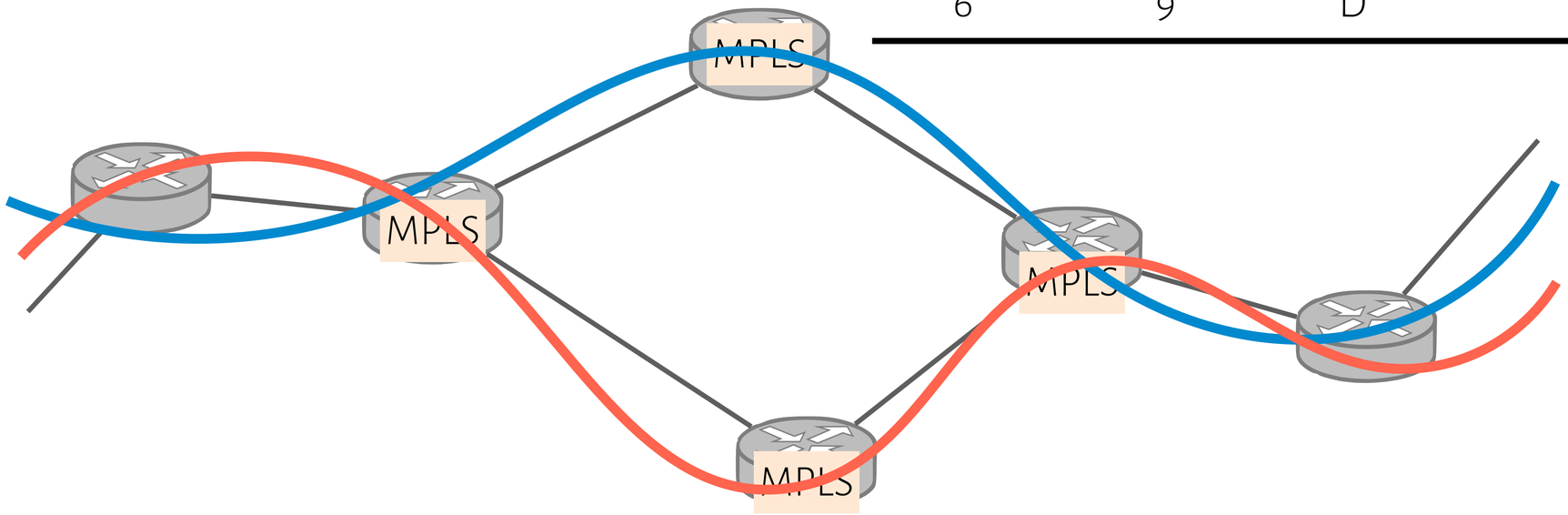
Label switched router

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

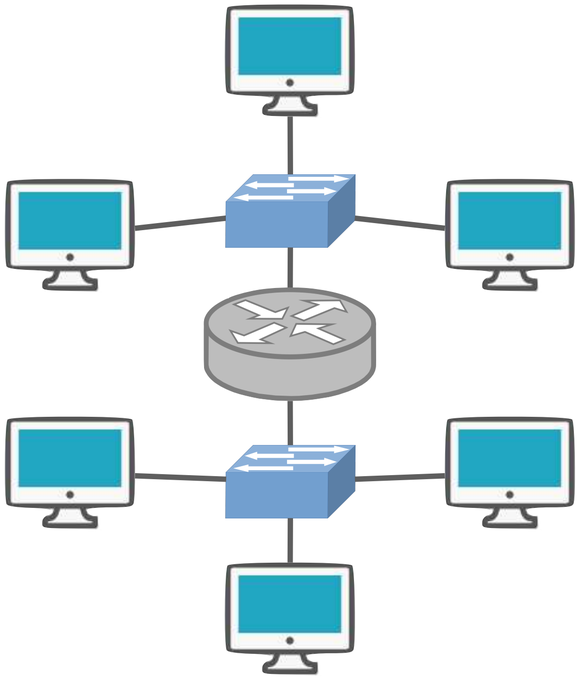


Questions

Recap

Lecture 7: Network forwarding and routing

- What does the link layer do?
- Switching in Ethernet
- STP, VLAN
- What does the network layer do?
- IP Forwarding and routing
- Traffic engineering, MPLS

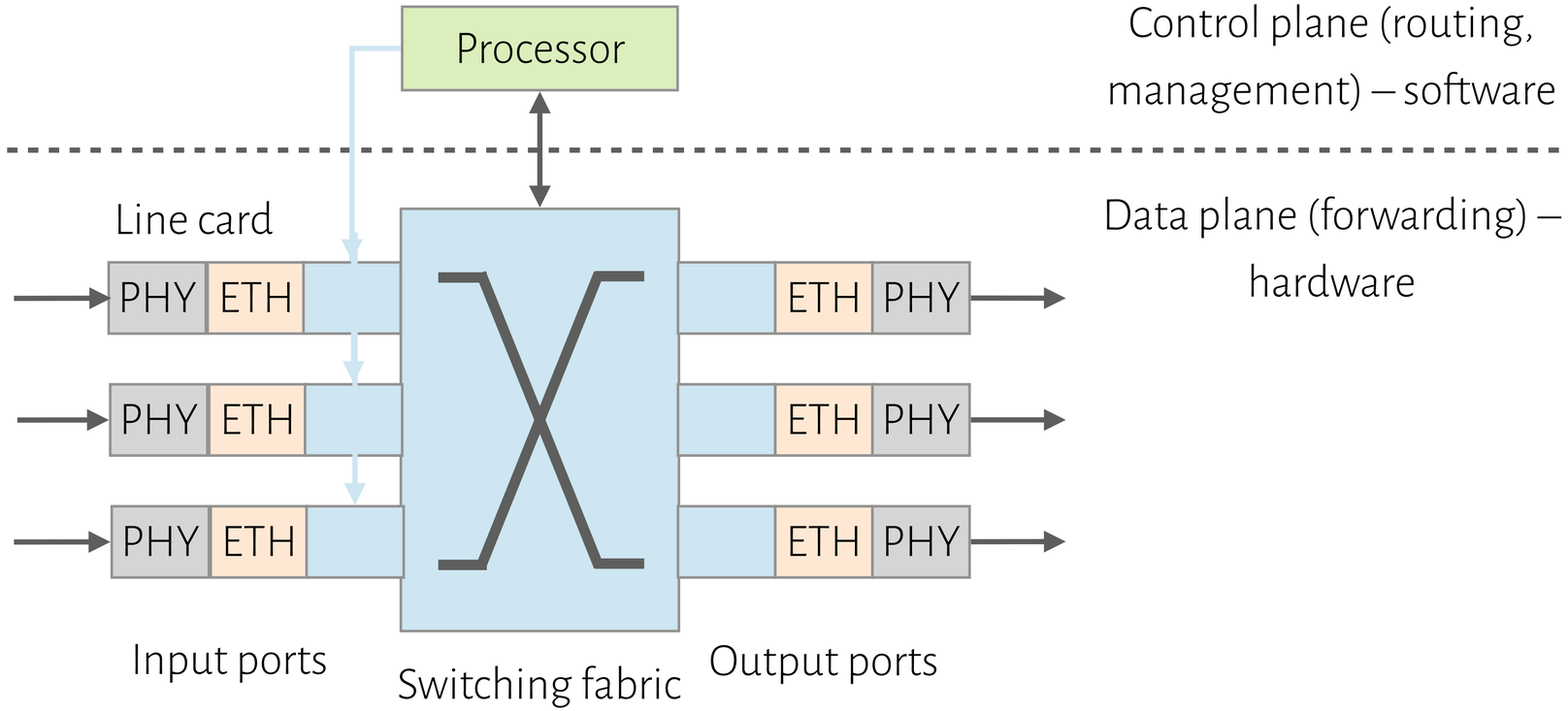


What are inside these boxes?

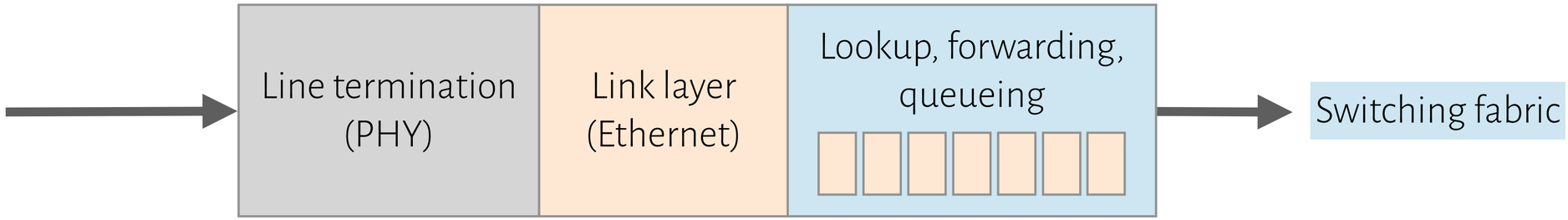
Router architecture

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

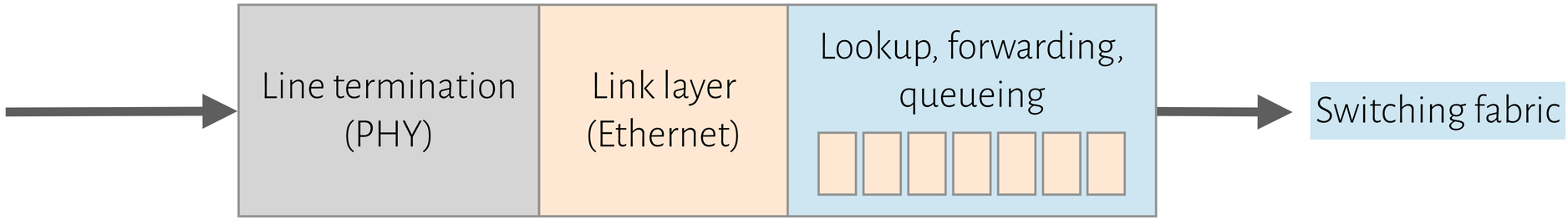


Decentralized switching:

- **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

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

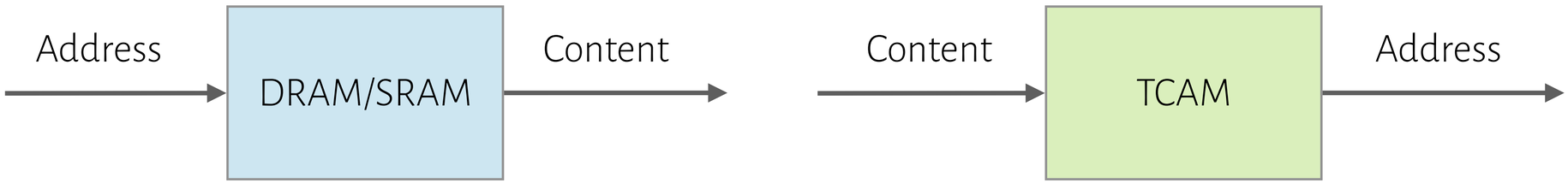
IP lookup

Match and action:

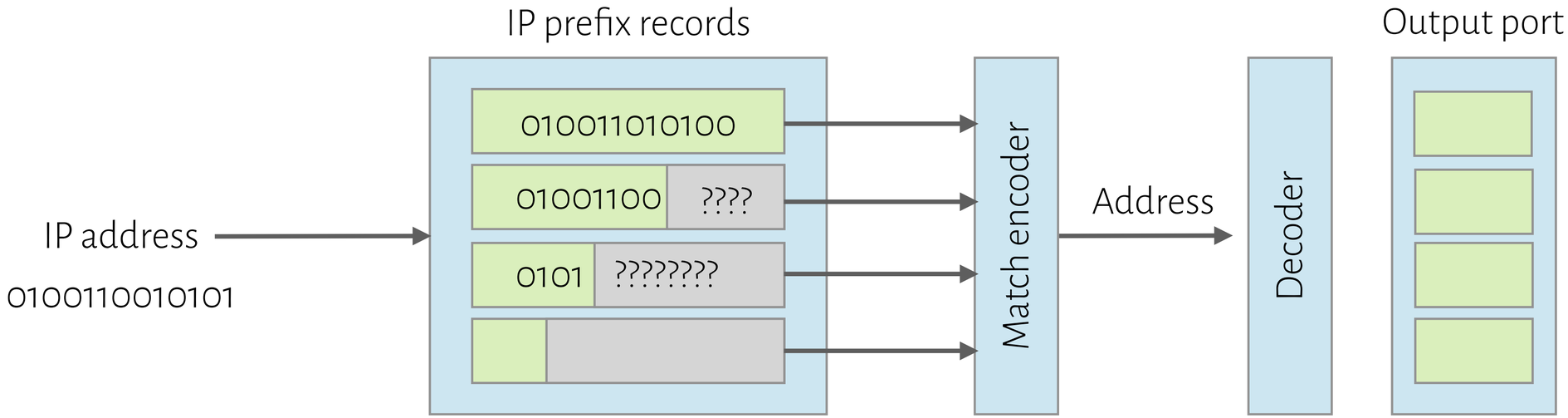
- Match on **IP prefix**: longest prefix matching rule
- Based on the match results, take an action: **forward** to an output port, **drop**, **replicate**, etc.

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 Context Addressable Memory (TCAM) for IP prefix matching



TCAM



Why IP prefix, not IP addresses?

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

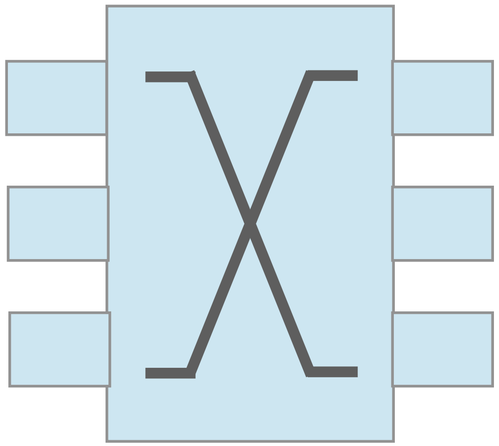
Transfer packet from input port to appropriate output port

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)

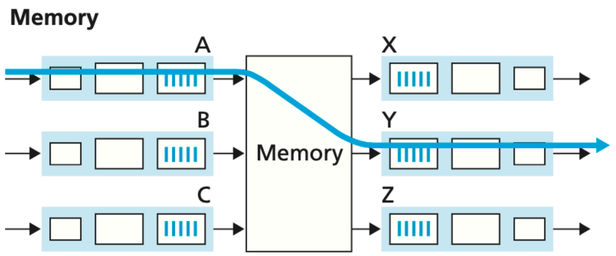


The Cisco 8000 Series Routers are the first routers in the industry that have the ability to redefine the economics of the Internet. They provide breakthrough density and massive scale, building the foundation of a new network for the next decade.

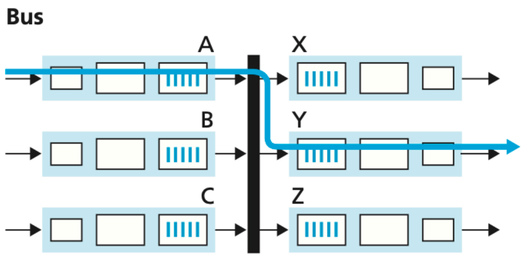
- 400G optimized platforms that scale from 10.8 Tbps to 260 Tbps.
- Design flexibility with up to 648 port configurations that support 100G or 400G throughput.
- Distinguished from System-on-Chip (SoC) designs by supporting full routing functionality on a single ASIC.
- Fully featured carrier-grade routing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

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

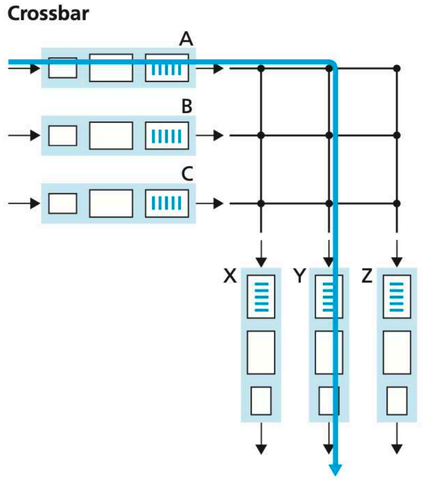
Switching fabric types



Throughput limited by memory bandwidth/2 (this is pretty like the CPU-based processing)



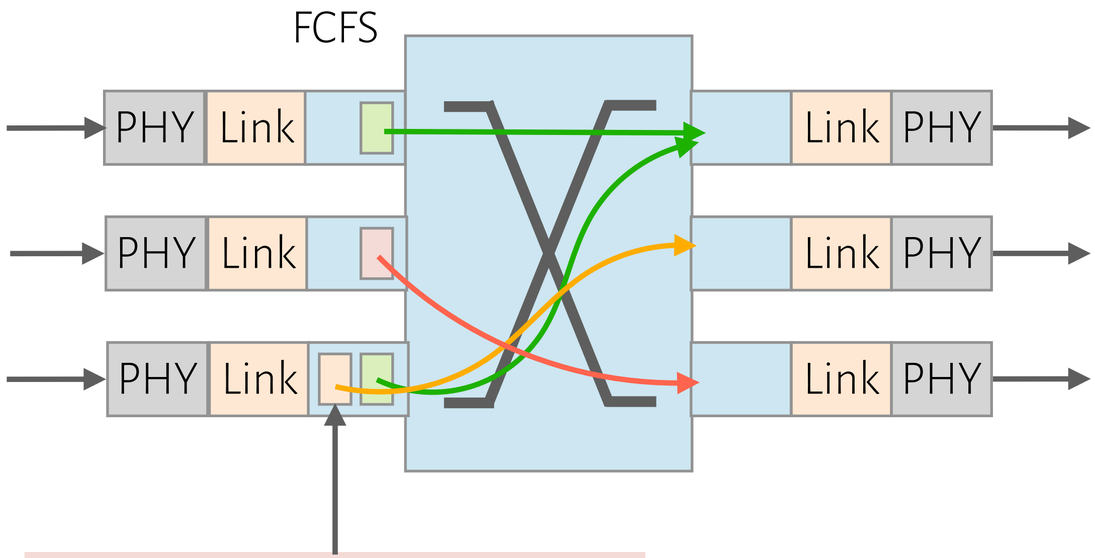
Throughput limited by the bus speed (only one packet at a time), sufficient for small local area/enterprise



Non-blocking (a packet to an output will not be blocked if there are no other packets to the same output)

Switching capacity can be scaled up by running switching fabrics in parallel.

Head of line (HOL) blocking



The beige packet is blocked and has to wait until the green one before it is transmitted.

Throughout is degraded to 58.6%!!

IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. COM-35, NO. 12, DECEMBER 1987

Input Versus Output Queueing on a Space-Division Packet Switch

MARK J. KAROL, MEMBER, IEEE, MICHAEL G. HLUCHYJ, MEMBER, IEEE, AND SAMUEL P. MORGAN, FELLOW, IEEE

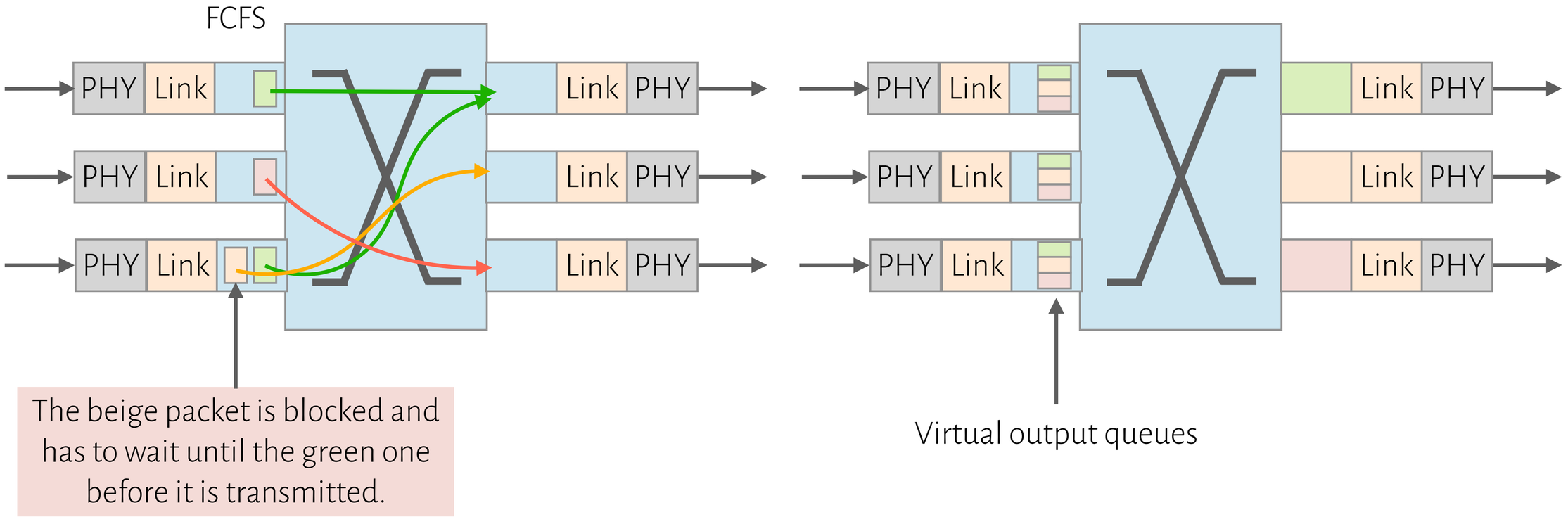
***Abstract*—Two simple models of queueing on an $N \times N$ space-division packet switch are examined. The switch operates synchronously with fixed-length packets; during each time slot, packets may arrive on any inputs addressed to any outputs. Because packet arrivals to the switch are unscheduled, more than one packet may arrive for the same output during the same time slot, making queueing unavoidable. Mean queue lengths are always greater for queueing on inputs than for queueing on outputs, and the output queues saturate only as the utilization approaches unity. Input queues, on the other hand, saturate at a utilization that depends on N, but is approximately $(2 - \sqrt{2}) = 0.586$ when N is large. If output trunk utilization is the primary consideration, it is possible to slightly increase utilization of the output trunks—up to $(1 - e^{-1}) = 0.632$ as $N \to \infty$—by dropping interfering packets at the end of each time slot, rather than storing them in the input queues. This improvement is possible, however, only when the utilization of the input trunks exceeds a second critical threshold—approximately $\ln (1 + \sqrt{2}) = 0.881$ for large N.**

$\sqrt{N}$ connections may be contending for use of the same center link. The use of a blocking network as a packet switch is feasible only under light loads or, alternatively, if it is possible to run the switch substantially faster than the input and output trunks.

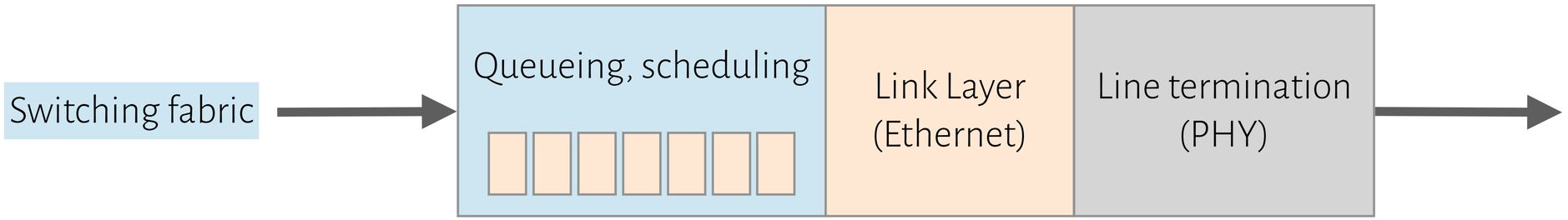
In this paper, we consider only nonblocking networks. A simple example of a nonblocking switch fabric is the crossbar interconnect with N^2 switch points (Fig. 1). Here it is always possible to establish a connection between any idle input–output pair. Examples of other nonblocking switch fabrics are given in [3]. Even with a nonblocking interconnect, some queueing in a packet switch is unavoidable, simply because the switch acts as a statistical multiplexor; that is, packet arrivals to the switch are unscheduled. If more than one packet arrives for the same output at a given time, queueing is required. Depending on the speed of the switch fabric and its particular architecture, there may be a choice as to where the queueing is done: for example, on the input trunk, on the output trunk, or

How to mitigate HOL blocking?

Head of line (HOL) blocking



Output port functions



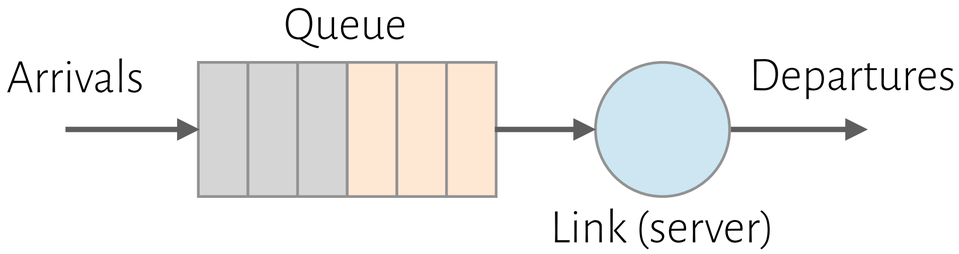
Queueing: required to handle the case where the speed packets depart from the switching fabric is faster than the transmission rate

- 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 $B = RTT \times C$

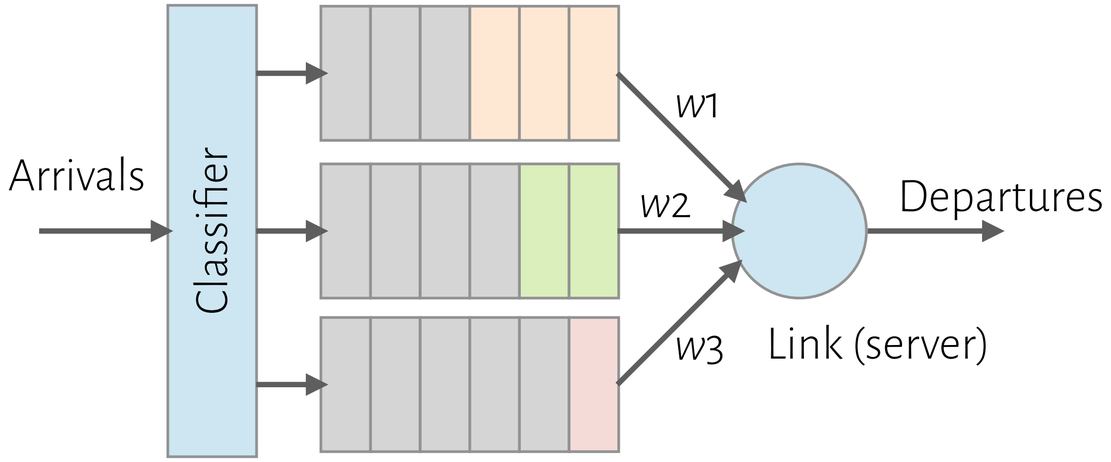
RFC 3439

Scheduling: decide which packet to go first on the wire

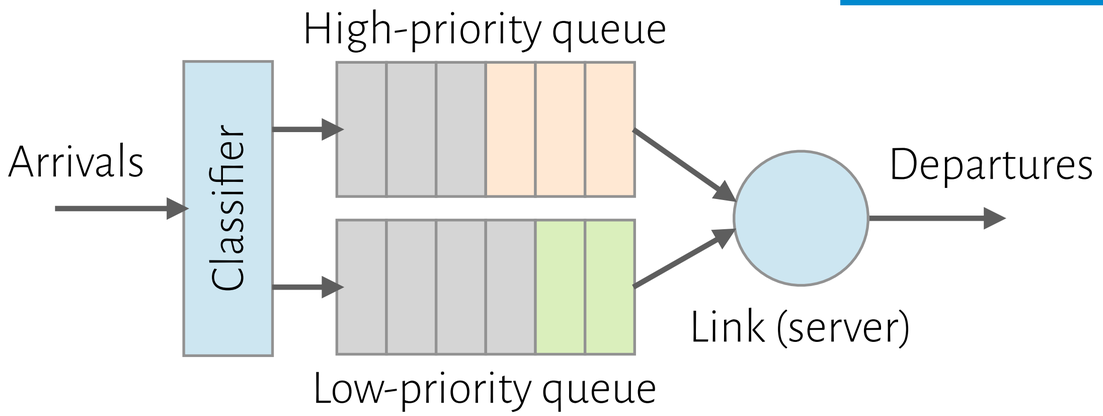
Packet scheduling policies



FIFO queueing model



Weighted fair queueing model

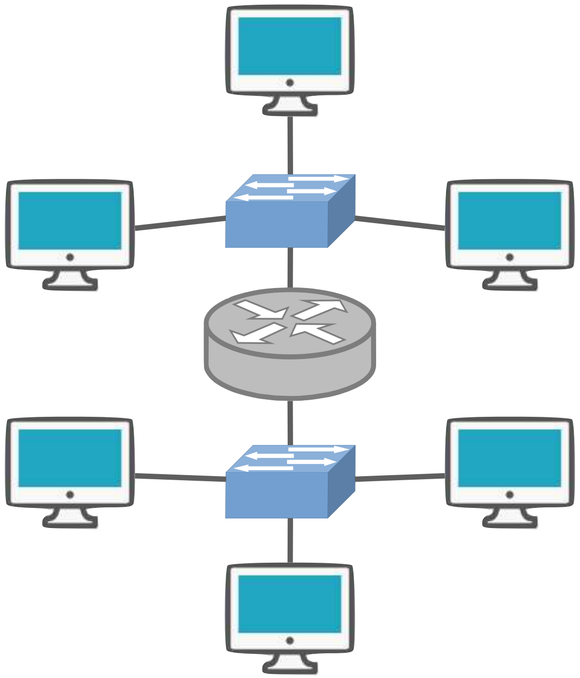


Priority (non-preemptive) queueing model

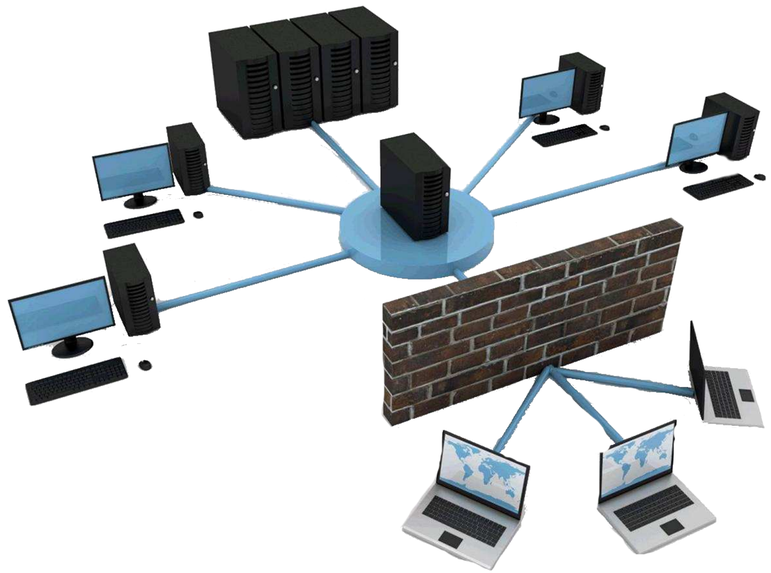
Summary

Lecture 7: Network forwarding and routing

- What does the link layer do?
- Switching in Ethernet
- STP, VLAN
- What does the network layer do?
- IP Forwarding and routing
- Traffic engineering, MPLS
- Router architecture



Next week: software defined networking



How do we manage the complex networks?

- Remember all the protocols
- Remember the configurations with every protocol
- Diagnose problems with networking tools like ping, traceroute, tcpdump?

Home > News

Misconfigured CenturyLink database caused global internet outage

By Sead Fadilpašić 20 days ago

Global internet traffic dropped by 3.5 percent.