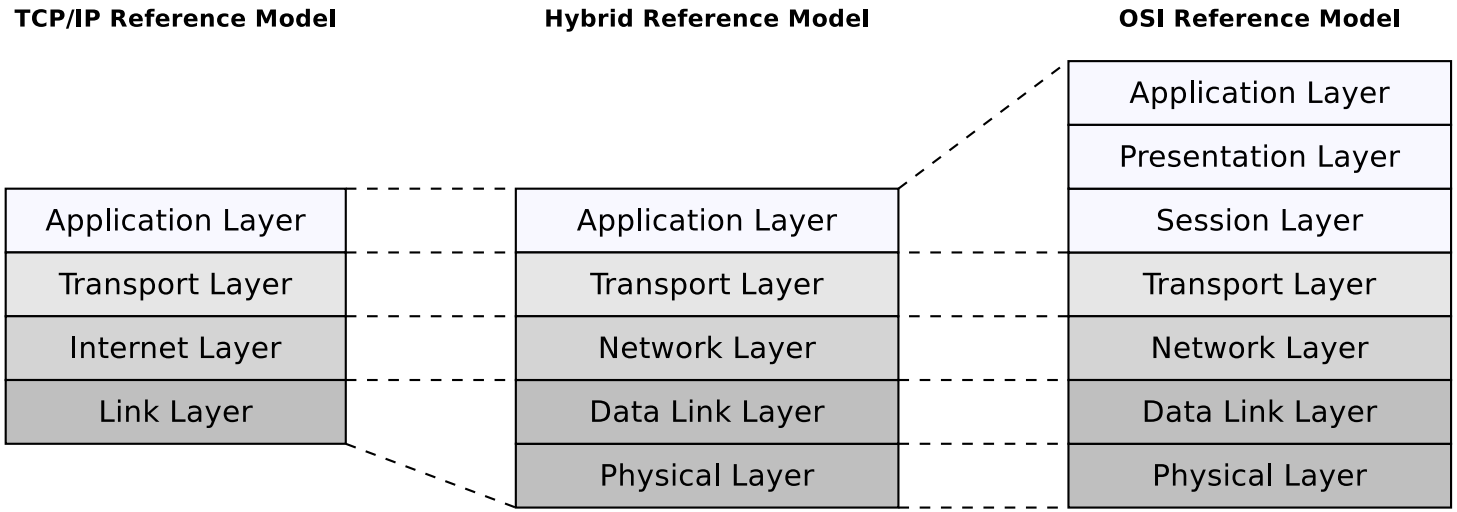
4th Slide Set
Computer Networks

Prof. Dr. Christian Baun

Frankfurt University of Applied Sciences
(1971–2014: Fachhochschule Frankfurt am Main)
Faculty of Computer Science and Engineering
christianbaun@fb2.fra-uas.de

Data Link Layer

- Functions of the Data Link Layer
  - Sender: Pack packets of the Network Layer into frames
  - Receiver: Identify the frames in the bit stream from the Physical Layer
  - Ensure correct transmission of the frames inside a physical network from one network device to another one via error detection with checksums
  - Provide physical addresses (MAC addresses)
  - Control access to the transmission medium



Exercise sheet 3 repeats the contents of this slide set which are relevant for these learning objectives

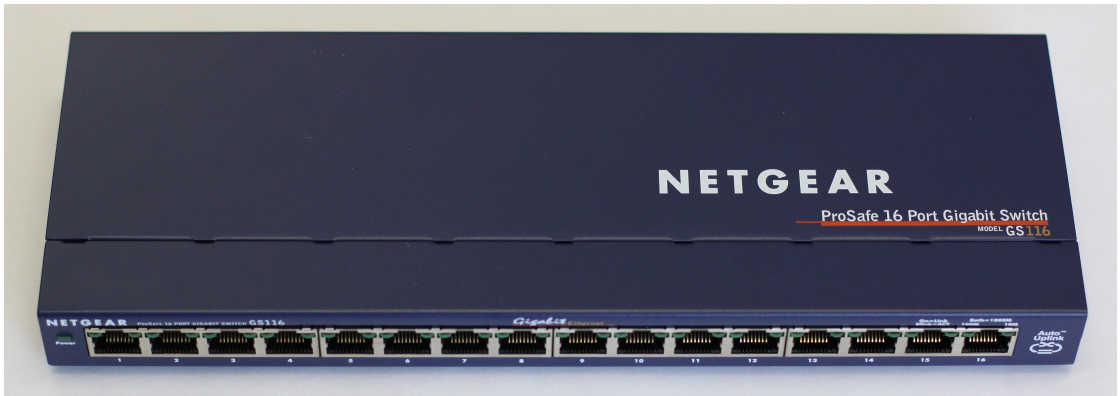
- Devices: Bridge, Layer-2-Switch (Multiport-Bridge), Modem
- Protocols: Ethernet, Token Ring, WLAN, Bluetooth, PPP

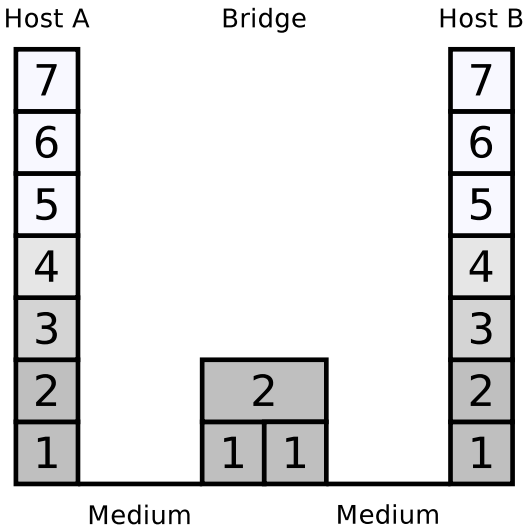
Learning Objectives of this Slide Set

- Data Link Layer (part 1)
  - Devices of the Data Link Layer
    - Learning Bridges
    - Loops on the Data Link Layer
    - Spanning Tree Protocol
    - Impact on the collision domain
  - Addressing in the Data Link Layer
    - Format of MAC addresses
    - Uniqueness of MAC addresses
    - Security aspects of MAC addresses

Devices of the Data Link Layer: Bridges

- Devices of the Physical Layer increase the length of physical networks
  - For connecting different physical networks, **Bridges** are required because they forward frames from one physical network to another one
- A Bridge has only 2 ports
  - Such bridges usually connect networks based on different technologies (transmission media) $\Longrightarrow$ see slides 6 and 7
- Simple Bridges forward all incoming frames





- Bridges with > 2 ports are called **Multiport Bridge** or **Layer-2-Switch**
  - They typically provide 4-48 Interfaces

Functioning of Bridges and Layer-2-Switches

- Bridges and Switches check the correctness of the frames via **checksums**
- Bridges do **not need addresses** for filtering and forwarding the frames, because they do not actively participate in the communication
  - They operate transparent, just like the devices of the Physical Layer
    - Reason: They do not communicate on a higher protocol layer as the Data Link Layer

Example of Bridges in everyday Life (1/2) – WLAN Bridge

- Integrates network devices with RJ45 jacks (e.g. network printers, desktops, gaming consoles,. . .) into a wireless local area network (WLAN)
- Connects a cable-based network with a wireless network

Example of Bridges in everyday Life (2/2) – Laser Bridge

- Connect 2 buildings via laser
  - Each building is equipped with a send and receive unit
  - Interesting alternative to cables if the field of view is not blocked

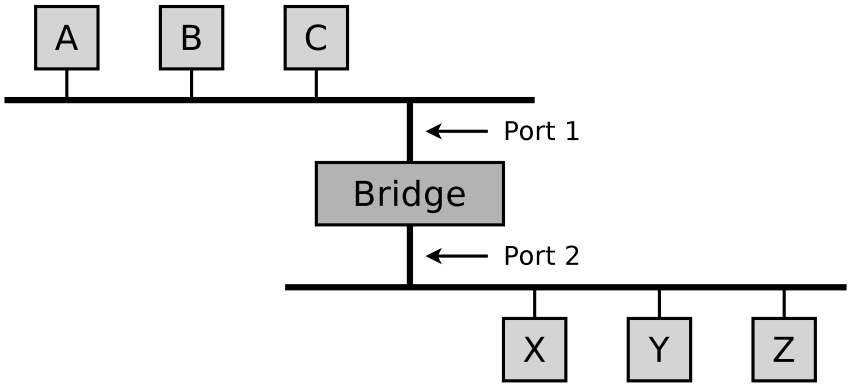


Image source: `http://www.made-in-zelenograd.com` and `http://www.laseritc.ru`

Interesting build instructions for your own Laser Bridge

- `https://hackaday.com/2017/04/19/go-wireless-with-this-diy-laser-ethernet-link/`
- `http://blog.svenbrauch.de/2017/02/19/homemade-10-mbits-laser-optical-ethernet-transceiver/`

Learning Bridges (1/2)



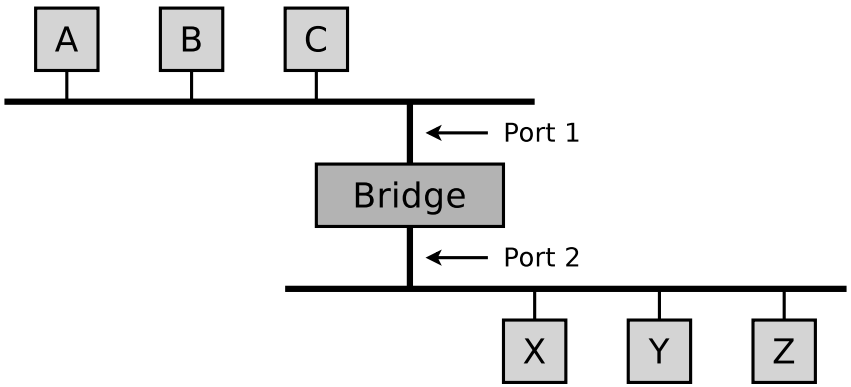
- Optimization: **Learning Bridges**
- The figure demonstrates that it is not useful when a Bridge forwards all frames
- Example: If a frame from participant B for participant A arrives at port 1, it is not required that the Bridge forwards this frame via port 2

Device	Port
A	1
B	1
C	1
X	2
Y	2
Z	2

- Bridges need to learn which network devices are accessible via which port
- Administrators could maintain the tables inside the Bridges
  - This would be very time consuming
- Manual intervention is not required, because the Bridges maintain their **forwarding tables** themselves

Learning Bridges (2/2)

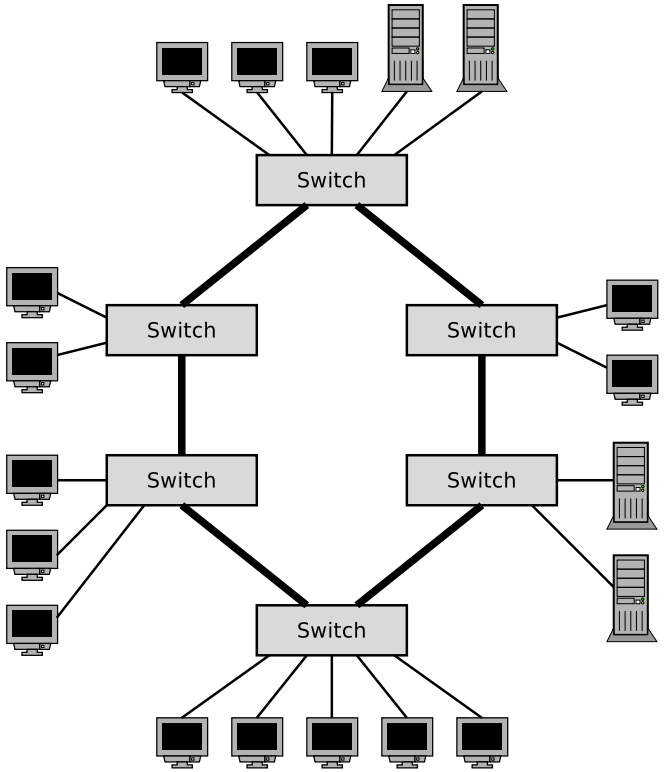
- Strategy:
  - **Bridges store the sender addresses of the frames they receive**
    - If device A sends a frame to another host, the Bridge stores the information that the frame from device A was received on port 1
  - This way, the forwarding table is populated over time with entries that specify what network devices are connected to which physical networks



- During bootup of a Bridge, its forwarding table is empty
  - The entry are recorded over time
  - Each entry has an **expiration date** (Time To Live – TTL)
    - They are only valid for a certain period of time
- The forwarding table is not complete all the time
  - This is not a problem, because the table is only used for **optimization**
    - If for a network device, no entry exists in the forwarding table, the Bridge forwards the frame in every case

Loops on the Data Link Layer

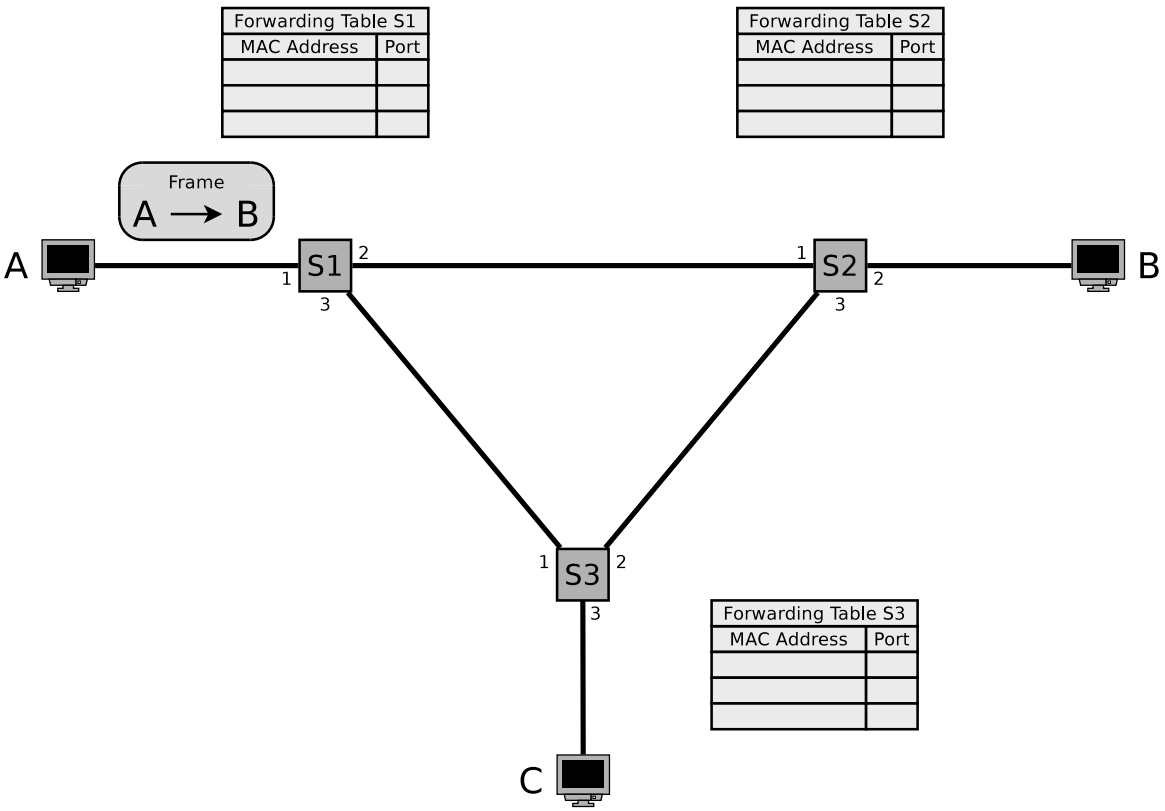
- **Loops** are a potential issue on the Data Link Layer
  - Computer networks should always provide only a single path to each possible destination on the Data Link Layer
    - That is to avoid that frames are duplicated and arrive multiple times at the destination
  - Loops can reduce the performance of the network or even lead to a network failure
    - On the other hand, redundant connections serve as a backup in case of a cable failure



Reasons why loops can occur on the Data Link Layer

- Careless administrators
- Intention to prevent that connections can fail via redundant cables

Example of Loops in a LAN (1/6)

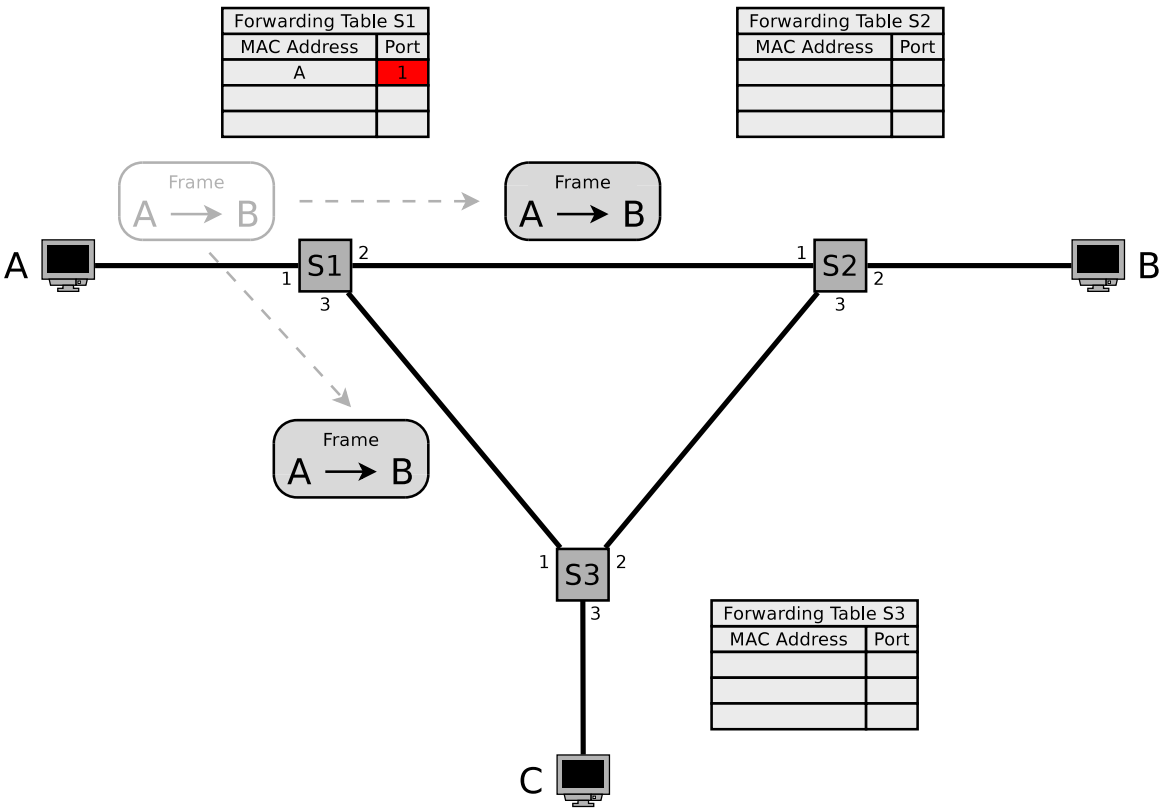


- A local area network has loops on the Data Link Layer
- The forwarding tables of the Switches are empty
- Node A wants to send a frame to node B

Similar examples can be found here:

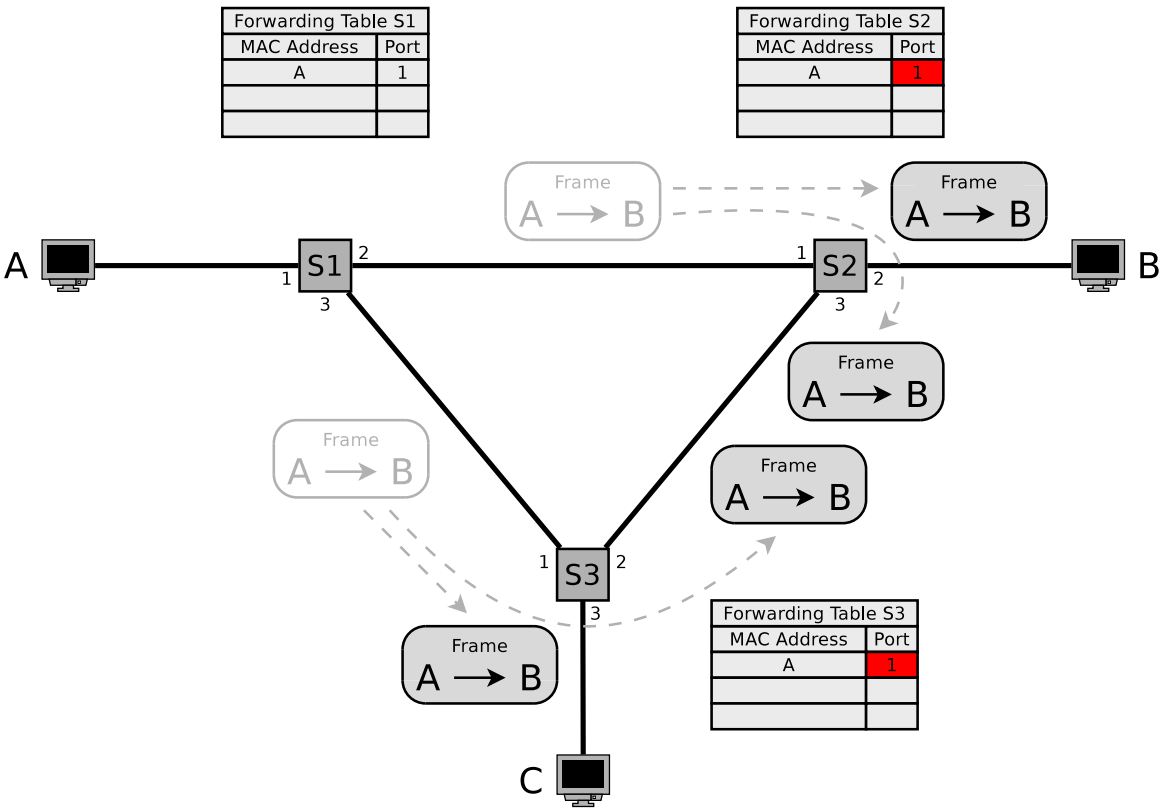
- *Olivier Bonaventure*. `http://cnp3book.info.ucl.ac.be/2nd/html/protocols/lan.html`
- *Rüdiger Schreiner*. Computernetzwerke. Hanser (2009)

Example of Loops in a LAN (2/6)



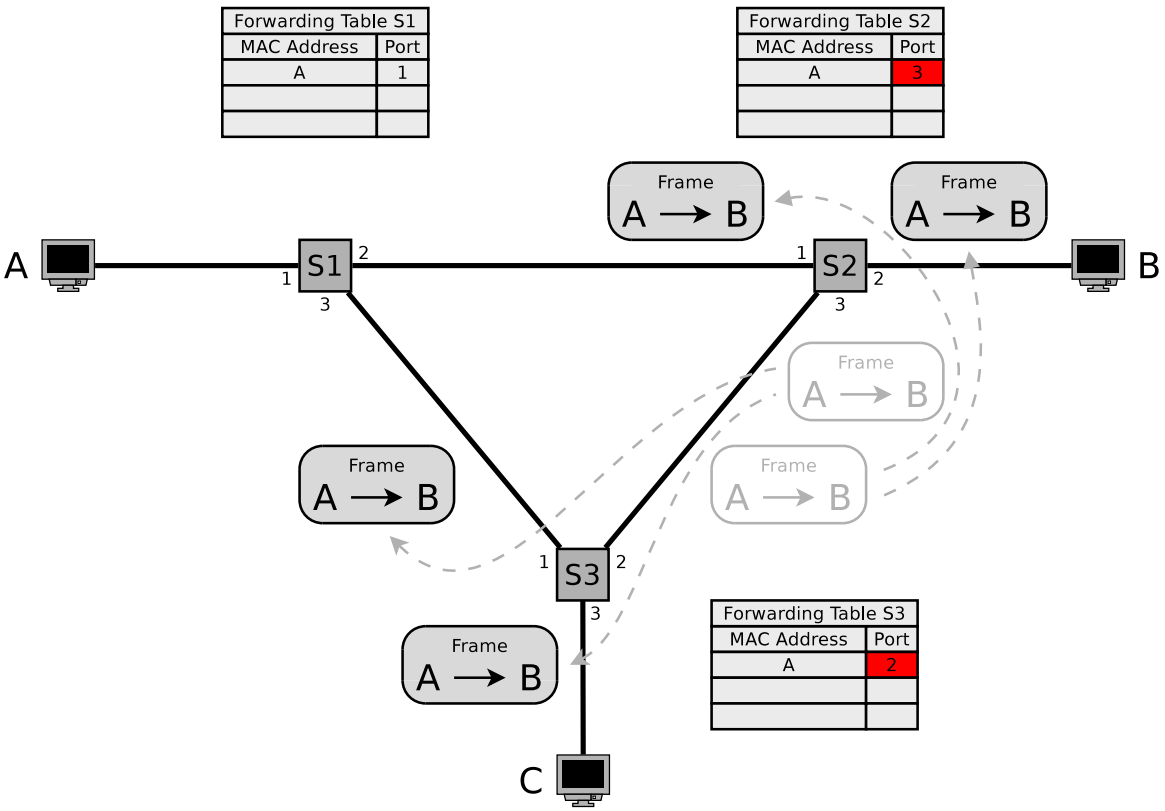
- The frame passes Switch 1
- Switch 1 stores the port to node A in its table
- Switch 1 don't know the path to node B
  - Therefore, it sends copies of the frame to all ports (except port 1)

Example of Loops in a LAN (3/6)



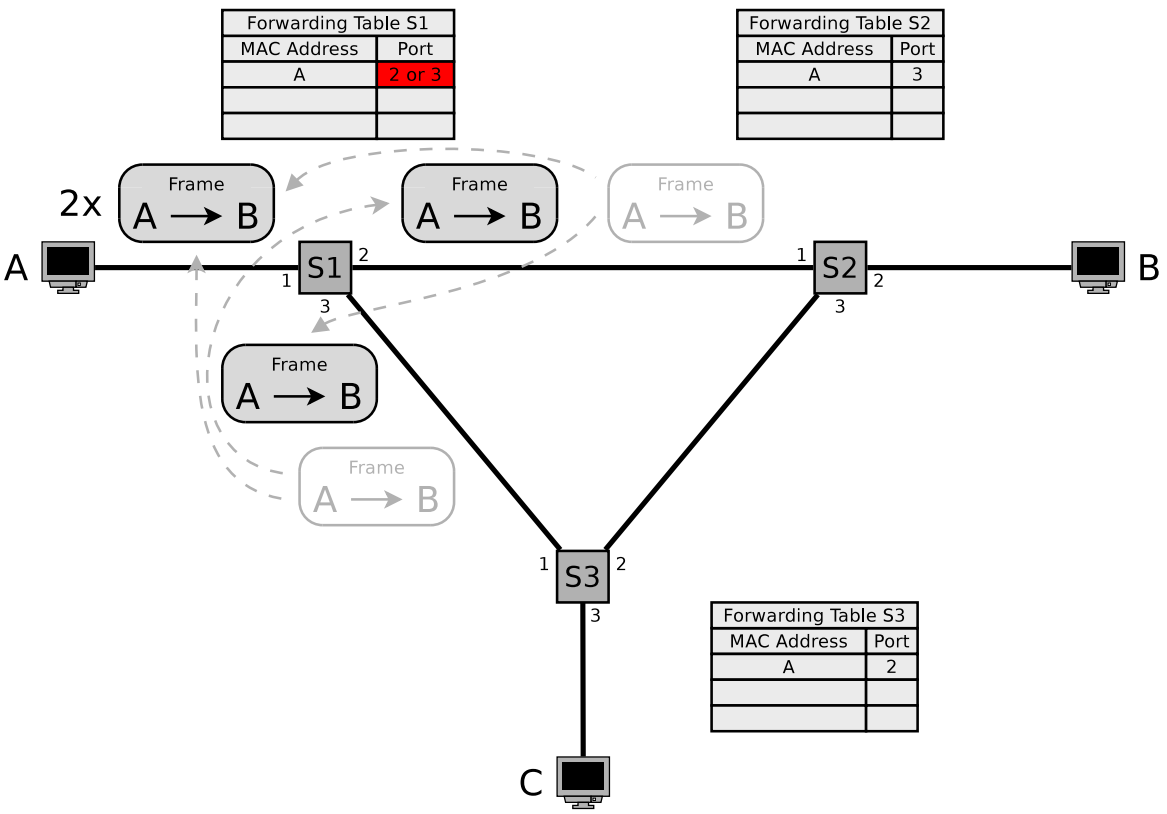
- The frame passes Switch 2 and 3
- Switch 2 and 3 store the port to node A in their tables
- Switch 2 and 3 both don't know the path to node B
  - Therefore, Switch 2 and 3 both send copies the frame to all ports except to ones, where the frame reached Switch 2 and 3

Example of Loops in a LAN (4/6)



- Copies of the frame again pass Switch 2 and 3
- Switch 2 and 3 update their tables

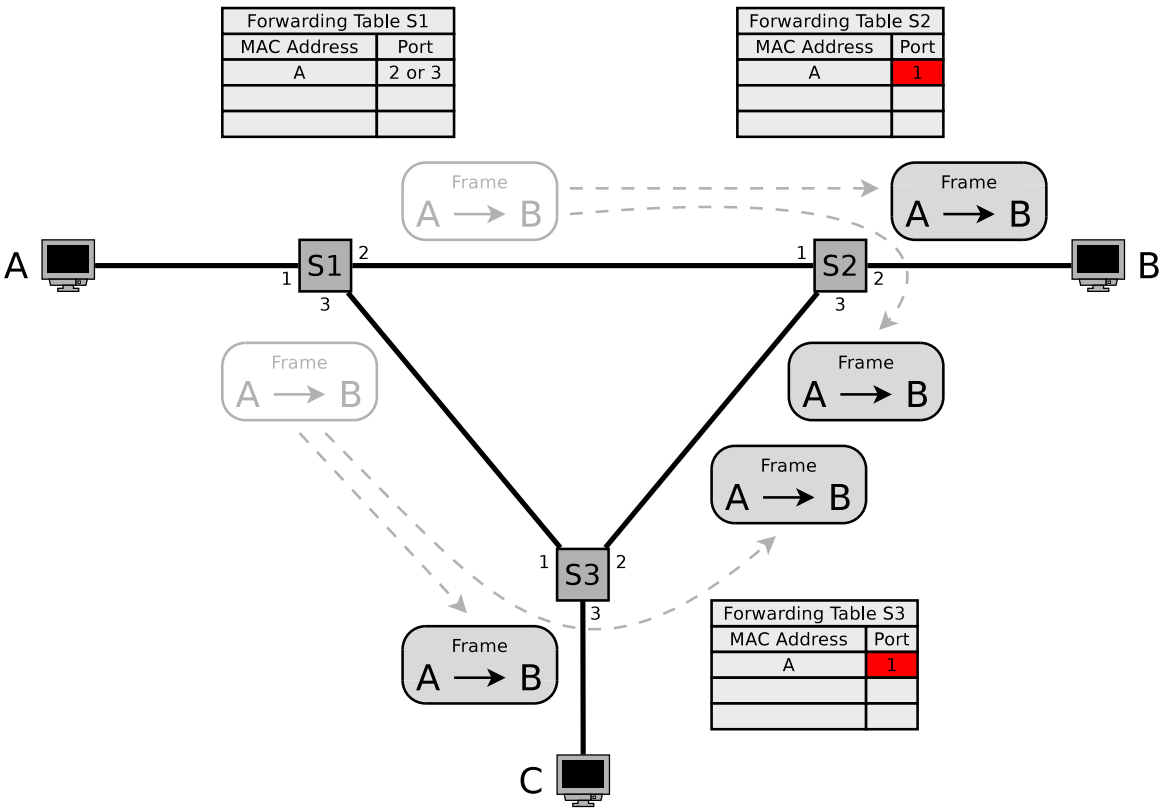
Example of Loops in a LAN (5/6)



- 2 copies of the frame arrive at Switch 1
  $\Longrightarrow$ **Loop!**
- Switch 1 sends copies of the frame, received on...
  - port 2 via port 1 and 3
  - port 3 via port 1 and 2
- Switch 1 updates its table 2 times

- It is impossible to predict the order in which the frames reach Switch 1

Example of Loops in a LAN (6/6)

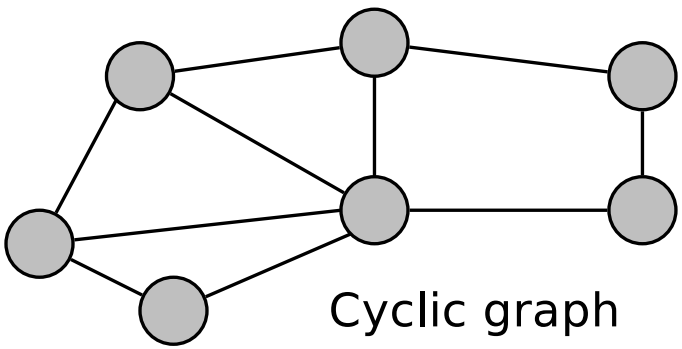


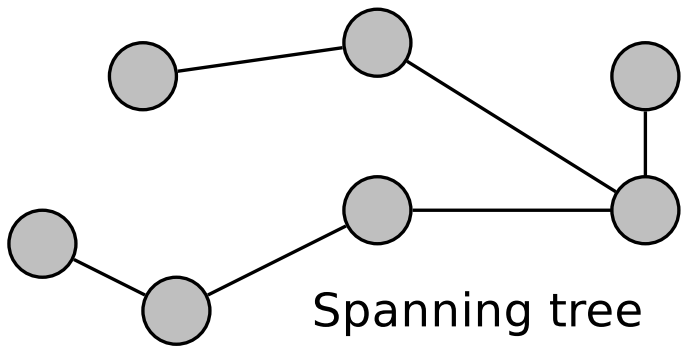
- Each frame of node A causes 2 copies, which infinitely circulate in the network
  - If node A sends further frames, the network gets flooded and will collapse at some point in time

- Copies of the frame again pass Switch 2 and 3
- Switch 2 and 3 update their tables
- **Ethernet does not contain any TTL or HopLimit**
  - Therefore, this loop will not stop until the tables in the switches contain an entry for node B

Handle Loops in the LAN

- Bridges need to be able to handle loops
- Solution: **Spanning Tree Algorithm**

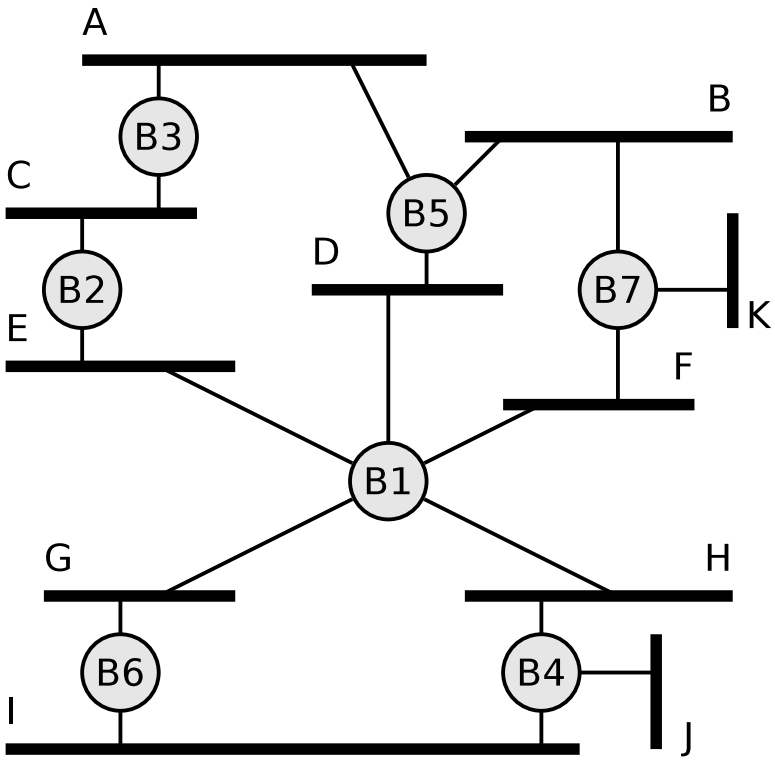




- A computer network, which consists of multiple physical networks, is a graph that may contain loops
  - The spanning tree is a subgraph of the graph that covers all nodes, but is cycle-free, because edges have been removed
  - The implementation of the algorithm is the **Spanning Tree Protocol** (STP)

Spanning Tree Protocol

Image source: Peterson, Davie. *Computernetze*



The STP was developed in the 1980s by Radia Perlman at Digital Equipment Corporation (DEC)

- The figure contains multiple loops
  - Via the STP, a group of Bridges can reach an **agreement for creating a spanning tree**
    - By **removing single ports of the Bridges**, the computer network is reduced to a cycle-free tree
- The algorithm works in a **dynamic** way
  - If a Bridge fails, a new spanning tree is created

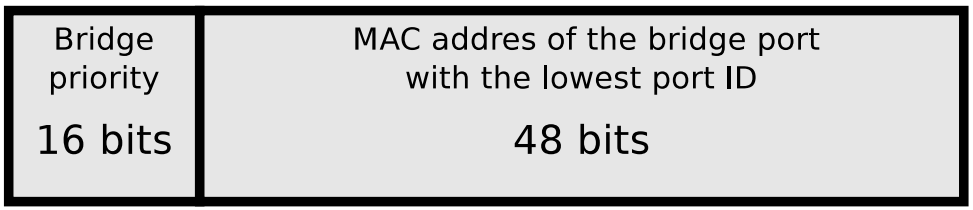
The protocol and format of the configuration messages are described in detail in the standard IEEE 802.1D

The Spanning Tree Protocol will only be discussed in this course in a simplified way

Spanning Tree Protocol – Precondition (1/2)

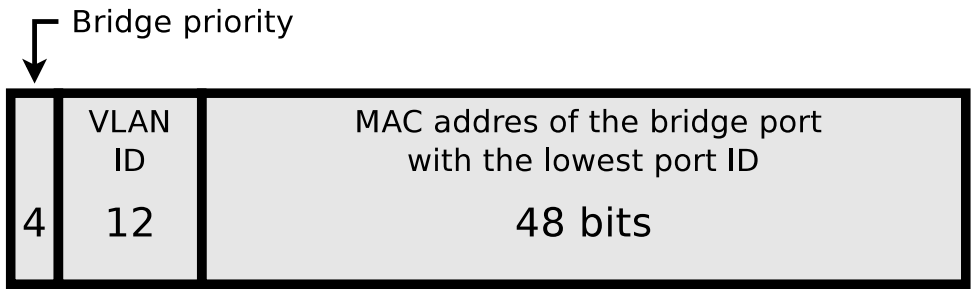
- For the functioning of STP, each Bridge needs an unique identifier
  - Length of the identifier (**Bridge ID**): 8 bytes
  - 2 different implementations of the Bridge ID exist

1. **Bridge ID according to IEEE**
   - The Bridge ID consists of the Bridge priority (2 bytes) and MAC address (6 bytes) of the Bridge port with the lowest port ID
     - The Bridge priority can be set by the administrator himself and can have any value between 0 and 65,535
     - Default value: 32,768



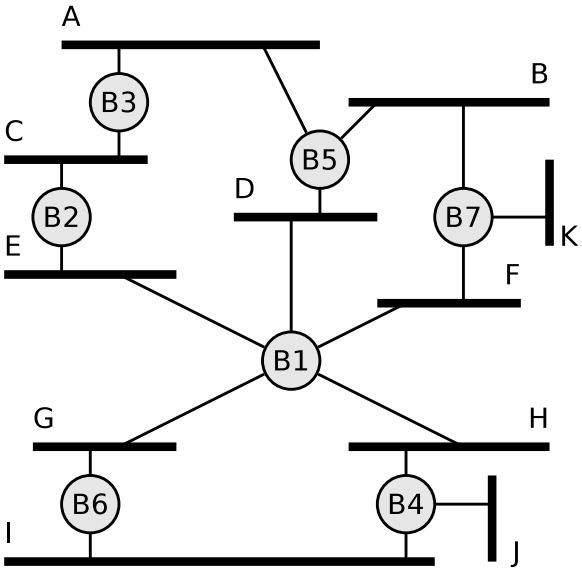
Spanning Tree Protocol – Precondition (2/2)

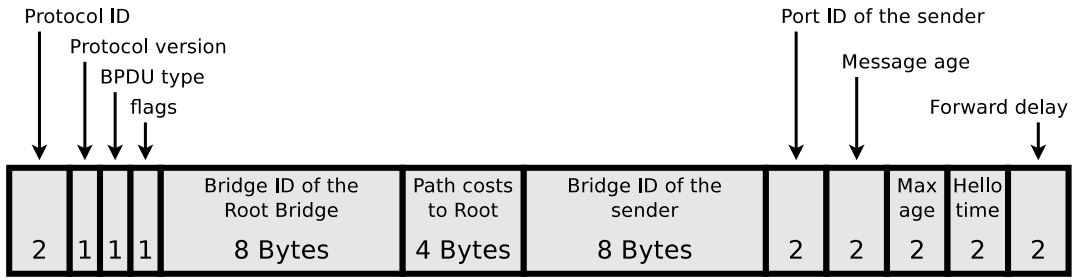
2. **Cisco extension of the Bridge ID, introducing the Extended System ID**
   - Cisco supports for its Bridges that each virtual LAN (VLAN) creates its own spanning tree
   - The original 2 bytes long part for the Bridge priority is subdivided
     - 4 bits now represent the Bridge priority
       $\Longrightarrow$ only 16 values can be represented
       $\Longrightarrow$ the value of the Bridge priority need to be zero or a multiple of 4,096
       $\Longrightarrow$ 0000 = 0, 0001 = 4,096 ... 1110 = 57,344, 1111 = 61,440
     - 12 bits are called **Extended System ID** and encode the VLAN ID
       $\Longrightarrow$ The content matches the VLAN tag of the Ethernet frames
       $\Longrightarrow$ With 12 bits, 4,096 different VLANs can be addressed



Spanning Tree Protocol – Functioning (1/2)

- The Bridges exchange information about Bridge IDs and path costs via special data frames, called **Bridge Protocol Data Unit** (BPDU)
  - They are sent in the payload field of Ethernet frames via broadcast to the neighboring Bridges





- First, the Bridges determine the Bridge with the lowest Bridge Priority value inside the Bridge ID
  - This Bridge is the **Root Bridge** of the spanning tree to be generated
  - If the Bridge priority is equal for multiple Bridges, the Bridge with the lowest MAC address becomes the Root Bridge

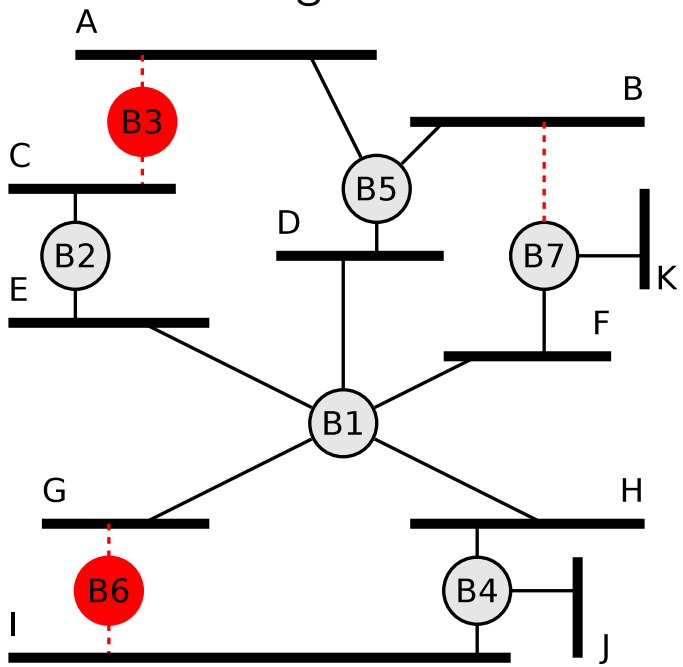
Spanning Tree Protocol – Functioning (2/2)

- **For each physical network**, a single one of the directly connected Bridges needs to be selected as responsible for **forwarding the frames into the direction of the Root Bridge**
  - The Bridge is called **Designated Bridge** for this network
  - Always the Bridge with the **lowest path costs to the Root Bridge** becomes the Designated Bridge
    - The path cost to the Root Bridge is the sum of the path costs of the different physical networks on the path to the Root Bridge

Data rate	Path costs
10.000 Mbps	2
1.000 Mbps	4
100 Mbps	19
16 Mbps	62
10 Mbps	100
4 Mbps	250

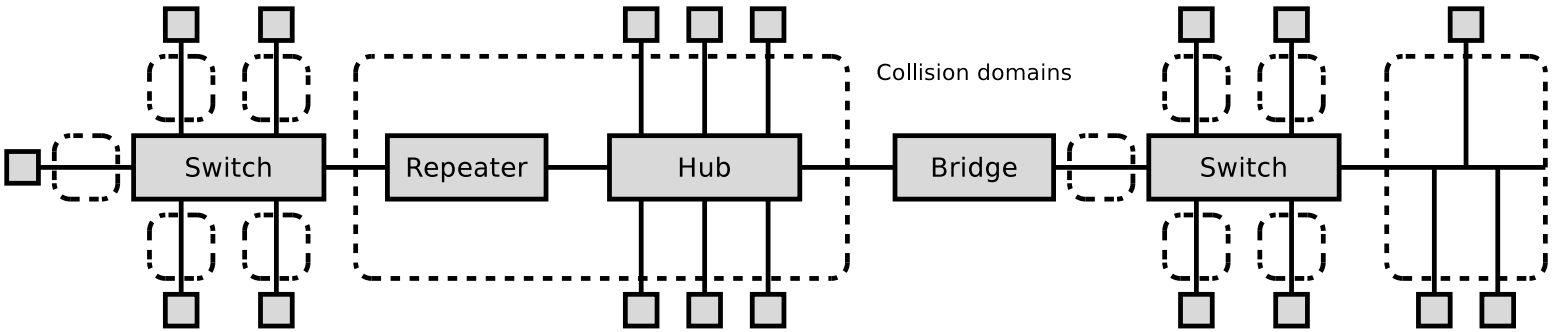
The path costs have been standardized by the IEEE, but can be adjusted manually

The exchange of the BPDU messages will not be discussed in detail in this course



Collision Domain – Bridges and Layer-2-Switches

- Bridges and Switches operate on Data Link Layer and forward frames from one physical network to other ones
- **Each physical network is a separate collision domain**
  - If a physical network is split by a Bridge or a Switch, also the collision domain is split
    - As effect, the number of collisions drops
- For Bridges and Switches, each port forms its own collision domain



- In a *fully switched network*, each port of the Switches is connected with just a single network device
  - Such a network is collision-free and state of the art

Addressing in the Data Link Layer

- The Data Link Layer protocols specify the format of the physical network addresses
- **Terminal devices (Hosts)**, **Routers** and **Layer-3-Switches** require physical network addresses
  - Such devices must be addressable on Data Link Layer because they provide services at upper protocol layers
- **Bridges** and **Layer-2-Switches** do not actively participate in the communication
  - Therefore, they don't require physical network addresses for their basic functionality, which is the filtering and forwarding of frames
  - Bridges and Switches require physical network addresses, when they implement the STP to avoid loops, or when they offer services from an upper protocol layer
    - Examples are monitoring services or graphical web interfaces for administration tasks
- **Repeaters** and **Hubs** that operate only at the Physical Layer, have no addresses

MAC Addresses (1/2)

- The **physical network address** are called **MAC addresses** (Media Access Control)
  - They are independent from the logical addresses of the Network Layer
- Ethernet uses the **Address Resolution Protocol** (ARP) to resolute the logical addresses of the Network Layer (IPv4 addresses) to MAC addresses
  - For IPv6, the **Neighbor Discovery Protocol** (NDP) provides the identical functionality and operates in a similar way
- MAC addresses have a length of 48 bits (6 bytes)
  - Thus, the address space contains 2^{48} possible addresses
- In order to make the representation compact and human-friendly to read, MAC addresses are usually written in hexadecimal notation
  - The bytes are separated from each other with dashes (-) or colons (:)
- Example of the notation: `00-16-41-52-DF-D7`

MAC Addresses (2/2)

- Each MAC address is intended to be permanently assigned to a network device and unique
  - But it is often possible to modify MAC addresses by software
    - However, this modification applies only until the next reboot of the computer
- **MAC broadcast address**
  - If a network device wants to send a frame to all other devices in the same physical network, it inserts MAC broadcast address in the destination address field of the frame
  - All 48 bits of this MAC address have the value 1
  - Hexadecimal notation: `FF-FF-FF-FF-FF-FF`
  - Bridges and Switches do not forwarded frames to other physical networks, that contain the MAC broadcast address in the destination address field

Uniqueness of MAC Addresses

- The first 24 bits of the MAC address space are managed by the Institute of Electrical and Electronics Engineers (IEEE)
  - These 24 bits long addresses are called **MA-L** (MAC Address Block Large) or **OUI** (Organizationally Unique Identifier)
  - The OUIs can be checked in this IEEE database:

    `http://standards.ieee.org/develop/regauth/oui/public.html`
- The remaining 24 bits are specified by the hardware vendors independently for their network devices
  - That address space allows $2^{24} = 16,777,216$ individual device addresses per OUI

MAC addresses	Manufacturer	MAC addresses	Manufacturer	MAC addresses	Manufacturer
00-20-AF-xx-xx-xx	3COM	00-03-93-xx-xx-xx	Apple	00-0C-6E-xx-xx-xx	Asus
00-00-0C-xx-xx-xx	Cisco	00-50-8B-xx-xx-xx	Compaq	08-00-2B-xx-xx-xx	DEC
00-01-E6-xx-xx-xx	Hewlett-Packard	00-02-55-xx-xx-xx	IBM	00-02-B3-xx-xx-xx	Intel
00-04-5A-xx-xx-xx	Linksys	00-09-5B-xx-xx-xx	Netgear	00-04-E2-xx-xx-xx	SMC

- Smaller address spaces are available too: **MA-S** (MAC Address Block Small) and **MA-M** (MAC Address Block Medium)

Security Aspects of MAC Addresses

- For WLAN, MAC filters are often used to protect the Access Point
  - In principle, this makes sense, because the MAC address is the unique identifier of a network device
- However, the security level of MAC filters is low because MAC addresses can be modified via software
  - The method is called **MAC spoofing**

Working with MAC addresses under Linux

- Read out the own MAC address(es): `ip link` or `ifconfig`
- Read out the MAC address(es) of the neighbors (mostly the Routers): `ip neigh`
- Set MAC address: `ip link set dev <Interface> address <MAC Address>`
- Alternative: `ifconfig <Interface> promisc`
  and next: `ifconfig <Interface> hw ether <MAC Address>`