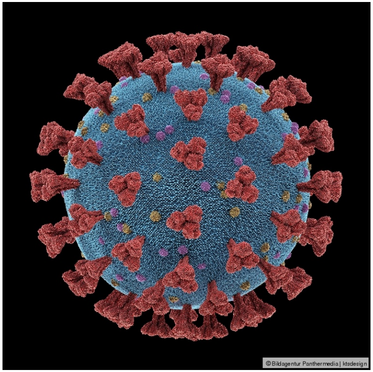
COVID-19 Measures

- Always wear a mask (medical or FFP2)
- Open the windows periodically whenever posible
- Behave reasonable and use common sense



Organizational

- Load balancing
  - Exercise on **Tuesday 11:45 – 13:15**, room BCN-421 is crowded
  - Exercise on **Thursday 14:15 – 15:45**, room 1-235 had recently only three students ...
  - $\rightarrow$ consider changing the group

Organizational

- Load balancing
  - Exercise on **Tuesday 11:45 – 13:15**, room BCN-421 is crowded
  - Exercise on **Thursday 14:15 – 15:45**, room 1-235 had recently only three students . . .
  - → consider changing the group
- Humming → RFC 7282 [1]

Rough Consensus

"We reject: kings, presidents and voting.
We believe in: rough consensus and running code."

`Dave Clark, IETF, 1992`

Organizational

- Load balancing
  - Exercise on **Tuesday 11:45 – 13:15**, room BCN-421 is crowded
  - Exercise on **Thursday 14:15 – 15:45**, room 1-235 had recently only three students . . .
  - $\rightarrow$ consider changing the group
- Humming $\rightarrow$ RFC 7282 [1]
  - You have understood how humming is supposed to work

Rough Consensus

"We reject: kings, presidents and voting.
We believe in: rough consensus and running code."

`Dave Clark, IETF, 1992`

Organizational

- Load balancing
  - Exercise on **Tuesday 11:45 – 13:15**, room BCN-421 is crowded
  - Exercise on **Thursday 14:15 – 15:45**, room 1-235 had recently only three students ...
  - $\rightarrow$ consider changing the group
- Humming $\rightarrow$ RFC 7282 [1]
  - You have understood how humming is supposed to work
  - You think the lecture is presented fairly interesting

Rough Consensus

"We reject: kings, presidents and voting.
We believe in: rough consensus and running code."

`Dave Clark, IETF, 1992`

Organizational

- Load balancing
  - Exercise on **Tuesday 11:45 – 13:15**, room BCN-421 is crowded
  - Exercise on **Thursday 14:15 – 15:45**, room 1-235 had recently only three students ...
  - → consider changing the group
- Humming → RFC 7282 [1]
  - You have understood how humming is supposed to work
  - You think the lecture is presented fairly interesting
  - The speed of the lecture is too high/too slow/good

Rough Consensus

"We reject: kings, presidents and voting.
We believe in: rough consensus and running code."

`Dave Clark, IETF, 1992`

Organizational

- Load balancing
  - Exercise on **Tuesday 11:45 – 13:15**, room BCN-421 is crowded
  - Exercise on **Thursday 14:15 – 15:45**, room 1-235 had recently only three students . . .
  - → consider changing the group
- Humming → RFC 7282 [1]
  - You have understood how humming is supposed to work
  - You think the lecture is presented fairly interesting
  - The speed of the lecture is too high/too slow/good
  - The exercises are helpful

Rough Consensus

"We reject: kings, presidents and voting.
We believe in: rough consensus and running code."

`Dave Clark, IETF, 1992`

Computer Networks

Data Link Layer - Framing and Switching

Prof. Dr. Oliver Hahm

Frankfurt University of Applied Sciences
Faculty 2: Computer Science and Engineering
`oliver.hahm@fb2.fra-uas.de`
`https://teaching.dahahm.de`

November 23, 2021

Agenda

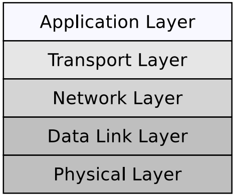
- Framing
  - Frame Detection
  - Ethernet (IEEE 802.3) Frames
  - WLAN (IEEE 802.11) Frames
- Addresses
- Switching
  - Devices
  - Forwarding
  - Loops

Data Link Layer

- Functions of the Data Link Layer

Framing	Encapsulate network layer datagrams into frames
Addressing	Provide physical addresses (MAC addresses)
Media Access	Coordinate the access of the transmission medium
Error Control	Detect and potentially correct errors
Flow Control	Ensure that the data rate does not exceed the receiver's capacity

Hybrid Reference Model



The Data Link Layer can be split into

- Media Access Control (MAC) sublayer and
- Logical Link Control (LLC) sublayer

Devices: Bridge, Switch, Modem

Protocols: Ethernet, Token Ring, WLAN, Bluetooth, PPP

Agenda

- Framing
  - Frame Detection
  - Ethernet (IEEE 802.3) Frames
  - WLAN (IEEE 802.11) Frames
- Addresses
- Switching
  - Devices
  - Forwarding
  - Loops

Agenda

Example: Problems in telegraph systems

A	· —	M	— —	Y	— · — —
B	— · · ·	N	— ·	Z	— — · ·
C	— · — ·	O	— — —	1	· — — — —
D	— · ·	P	· — — ·	2	· · — — —
E	·	Q	— — · —	3	· · · — —
F	· · — ·	R	· — ·	4	· · · · —
G	— — ·	S	· · ·	5	· · · · ·
H	· · · ·	T	—	6	— · · · ·
I	· ·	U	· · —	7	— — · · ·
J	· — — —	V	· · · —	8	— — — · ·
K	— · —	W	· — —	9	— — — — ·
L	· — · ·	X	— · · —	0	— — — — —

Morse Code

Used for telegraph systems

Problem

- The sender meant:
  — · · — — — $\rightarrow$ *DO*
- The receiver understood:
  · · — — $\rightarrow$ *EAT*

Framing

- The receiver needs to split the bit stream from the Physical Layer into frames
- The sender encapsulates the packets from the Network Layer into frames
- The start of each frame needs to be marked
- Different ways exist to mark the frames' borders
  - Character count in the header
  - Byte/Character stuffing
  - Bit stuffing
  - Line code violations of Physical Layer with illegal signals
- All these different procedures have advantages and drawbacks

Character Count in the Frame Header

- Include the character count in the header of the frame
- **Example:** the byte-oriented **Digital Data Communications Message Protocol** (DDCMP) of DECnet
- In each frame, the field `Count` contains the number of bytes payload inside the frame

8 Bits	14 Bits	2 Bits	8 Bits	8 Bits	8 Bits	16 Bits		16 Bits
SOH	Count	Flags	RESP	NUM	ADDR	CRC 1	Body	CRC 2
Start of Header	Number of bytes in payload					Checksum of header	Payload	Chacksum of payload

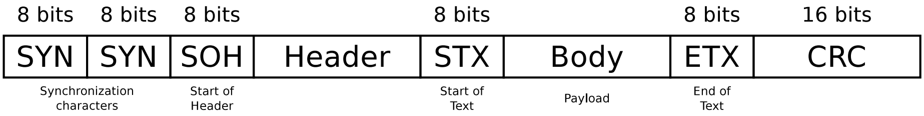
- Potential issue: If the field `Count` is modified during transmission, the receiver is unable to correctly detect the end of the frame

Byte/Character stuffing

- Control characters ("Sentinel characters") mark the start and end of the frames
- The method is called **Byte Stuffing** or **Character Stuffing**, because the. . .
  - sender **inserts** (stuffs) extra characters into the payload
  - receiver **removes** the stuffed characters from the received payload, before passing it to the Network Layer
- Drawback:
  - Strong relationship with the character encoding (e.g., ASCII)
    - More recent protocols of this layer no longer operate byte-oriented, but bit-oriented because this allows using any character encoding

Example: Byte/Character stuffing

- A protocol, which highlights the frames border with special characters, is the byte-oriented (character-oriented) protocol **Binary Synchronous Communication** (BISYNC), which was invented by IBM in the 1960s

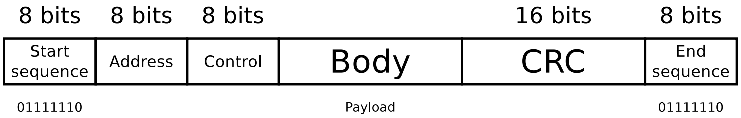


- The start of a frame highlights the character `SYN`
- The start of the header highlights the character `SOH` (*Start of Header*)
- The payload is between `STX` (Start of text) and `ETX` (*End of Text*)

- If the payload (body) contains an `ETX` character, it it must be escaped by a stuffed `DLE` (*Data Link Escape*)
- The `DLE` character is represented in the payload by sequence `DLE DLE`

Bit Stuffing

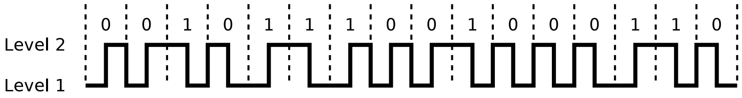
- When bit-oriented protocols are used, each frame begins and ends with a special bit pattern
- **Examples:** The protocol **High-Level Data Link Control** (HDLC) and the **Point-to-Point Protocol** (PPP), which implments HDLC
  - Each frame begins and ends with the sequence 01111110



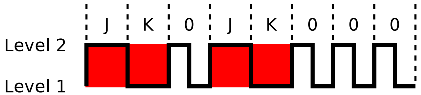
- If the HDLC protocol in the Data Link Layer. . .
  - of the sender discovers 5 consecutive 1-bits in the bit stream from the Network Layer, it stuffs a 0-bit in the outgoing bit stream
  - of the receiver discovers 5 consecutive 1-bits, followed by a 0-bit in the bit stream from the Physical Layer, it removes (destuffs) the 0-bit
- Advantages:
  - Ensures that the start/end sequence does not occur in the payload
  - Every character encoding can be used with this framing method

Line Code Violations

- Depending on the line code used in the Physical Layer, illegal signals can be used to highlight the frame boundaries
  - **Example:** Token Ring uses the Differential Manchester Encoding
    - With this line code, a signal level change occurs inside each bit cell



Starting delimiter of Token Ring

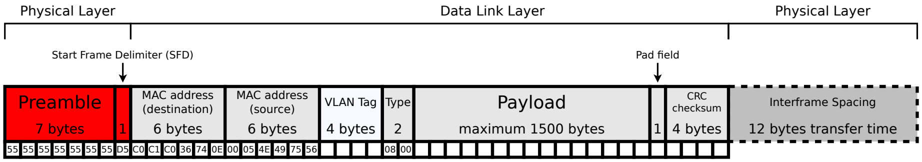


- If Token Ring is used, frames start with a byte (**starting delimiter**) which **contains 4 line code violations**

The second last byte (**ending delimiter**) of a Token Ring frame contains the same 4 line code violations as the starting delimiter

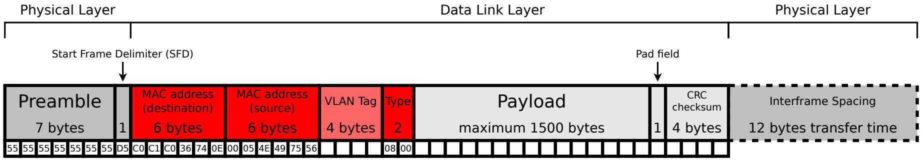
Agenda

Preamble and SFD



- Preamble is a 7 bytes long bit sequence 101010 . . . 1010
  - Allows the receiver to synchronize with the clock and to identify the beginning of the frame
  - Is followed by the SFD (1 byte) with the bit sequence 10101011

Addresses and VLAN Tag

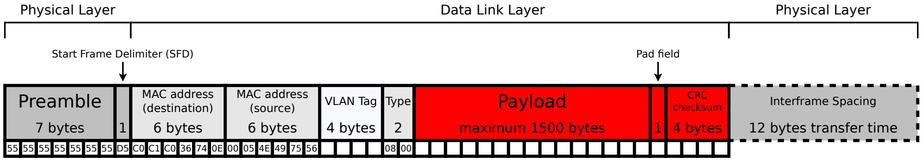


- The fields for the physical addresses (MAC addresses[2]) of sender and destination are 6 bytes long each
- The 4 bytes long optional tag[3]contains, among others. . .
  - a 12 bits long VLAN ID
  - and a 3 bits long field for the priority information
- The field Type contains the information what protocol is used in the network layer
  - If IPv4 is used, the field Type has value 0x0800
  - If IPv6 is used, the field Type has value 0x86DD
  - If the payload contains an ARP message, the field Type has value 0x0806

[2] The address format has been adopted by other IEEE 802 standards like WLAN or FDDI.

[3] Introduced by IEEE 802.1Q or IEEE 802.1ad.

Frame Size and Checksum



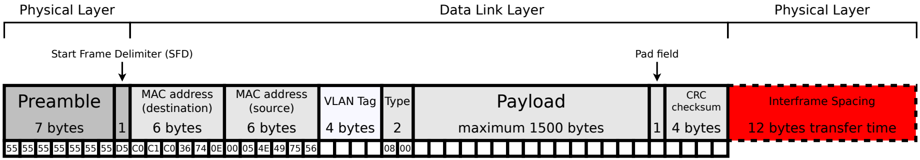
- Minimum size of an Ethernet frame: 72 bytes
- Maximum size (incl. preamble and SFD): 1526 bytes (1530 bytes incl. VLAN tag)

- The maximum frame size [4] of Ethernet limits the payload to 1500 bytes
  - With the Pad field, the frame length can be increased to the minimum frame size (72 bytes) when needed
    - This is required for the collision detection → medium access control
- The last field contains a checksum (32 bits) [5]

[5]Generically called the Maximum Transfer Unit (MTU).

[5]The checksum covers all fields except for preamble and SFD.

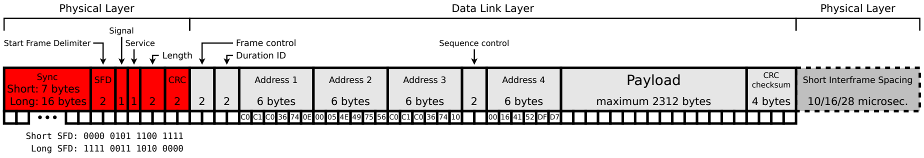
Interframe Spacing



- The Interframe Spacing or Interframe Gap is the minimum idle period between the transmission of Ethernet frames
- The minimum idle period is 96 bit times (12 bytes)
  - It is 9.6 microseconds when using 10 Mbps Ethernet
  - It is 0.96 microseconds when using 100 Mbps Ethernet
  - It is 96 nanoseconds when using 1 Gbps Ethernet
- Some network devices allow to reduce the Interframe Spacing period
  - Benefit: Better data rate is possible
  - Drawback: For the receiver it may become impossible to detect the frames' borders ($\Longrightarrow$ the number of errors may rise)

Agenda

Preamble and Layer 1 Header



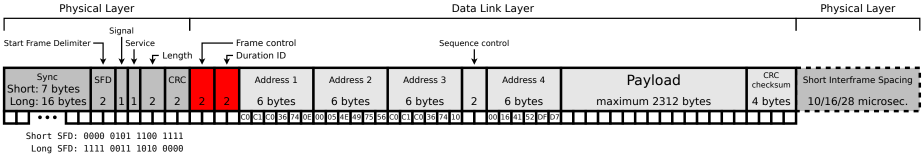
6

- For the Physical layer, the standard comprises. . .
  - a preamble to synchronize the receiver including a SFD [7]
  - a Signal field, specifying the payload data rate (1 to 11 Mbit/s)
  - a Service field, may contain additional information
  - a Length field, specifying the transmission time for the payload in microseconds
  - a CRC field, which contains a checksum over the fields Signal, Service and Length

[7] Frame format for IEEE 802.11b

[7] The Short Preamble Format is an optional standard which is not supported by all devices

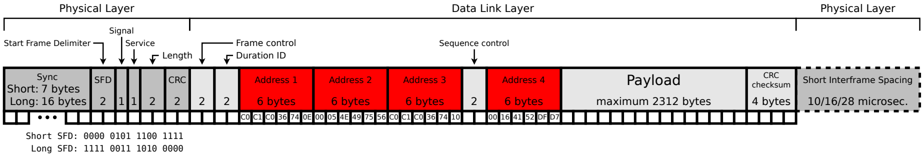
Frame Size, Frame Control, and NAV



- Maximum frame size of a WLAN frame (link layer part): 2346 bytes

- The field Frame Control (2 bytes) contains several smaller fields
  - Among other things, the protocol version, the type of frame (e.g., data frame or beacon), encryption with the WEP method
- The field Duration ID (2 bytes) contains a duration value for the update of the counter variable Network Allocation Vector (NAV) $\rightarrow$ Medium Access Control

Address Fields and SSIDs

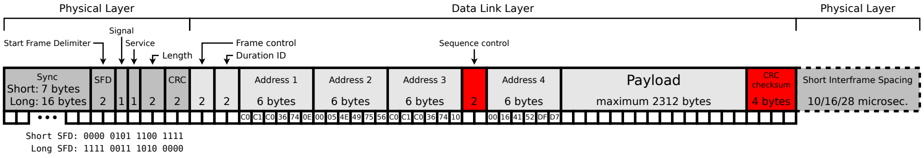


- The content of the 4 address fields varies and depends, among others, on whether infrastructure mode or ad-hoc mode are used
- An address field may contain a **Service Set Identifier** (SSID)
  - SSID: is the network name for the basic service set (BSS)
  - BSSID: is the MAC address of the Access Point's radio device for that BSS
  - ESSID: is used across multiple access points as part of the same WLAN

Possible use of the address fields:

- Address 1: MAC of receiver (Destination Address)
- Address 2: MAC of sender (Source Address)
- Address 3: Used for filtering
- Address 4: is used for communication between APs in ESSID configuration or in a mesh network

Sequence Control and CRC



- The field Sequence Control (2 bytes) consists of a fragment number (4 bits) and a sequence number (12 bits)
  - If a frame has been split into several fragments, the sequence number is equal for all fragments
- The final field contains a CRC checksum (32 bits) that covers all fields, except the payload

Agenda

Addressing in the Data Link Layer

- The Data Link Layer protocols specify the format of the physical network addresses
- Terminal devices (Hosts) or Routers
  - Such devices must be addressable on Data Link Layer because they provide services at upper protocol layers
- Bridges and Switches do not actively participate in the communication
  - Typically they do not require an address, because their main purpose is filtering and forwarding of frames
  - Their address becomes relevant to establish a hierarchy ($\rightarrow$ see slide 46) or providing a configuration interface
- Note: **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
- Network Layer protocols may employ their own protocols to resolve logical address on layer 3 to physical addresses of layer 2
  - IPv4 uses the Address Resolution Protocol (ARP)
  - IPv6 uses the Neighbor Discovery Protocol (NDP)
- IEEE 802 MAC addresses have a length of 48 bits (6 bytes)
- The human-friendly representation is typically given in hexadecimal notation with dashes (-) or colons (:) as separators
  - Example of the notation: `00-16-41-52-DF-D7`

EUI-48 and EUI-64

MAC addresses can be formed according to the numbering spaces based on Extended Unique Identifiers (EUI) managed by the IEEE: EUI-48 (e.g., Ethernet, WLAN, Bluetooth) and EUI-64 (Firewire, 6LoWPAN, Zigbee)

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
- MAC broadcast address
    - In IEEE 802 networks 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 Organizationally Unique Identifier (OUI)
  - 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
00-20-AF-xx-xx-xx	3COM	00-0C-6E-xx-xx-xx	Asus
00-00-0C-xx-xx-xx	Cisco	08-00-2B-xx-xx-xx	DEC
00-01-E6-xx-xx-xx	Hewlett-Packard	00-02-B3-xx-xx-xx	Intel
00-04-5A-xx-xx-xx	Linksys	00-04-E2-xx-xx-xx	SMC
00-03-93-xx-xx-xx	Apple	00-50-8B-xx-xx-xx	Compaq
00-02-55-xx-xx-xx	IBM	00-09-5B-xx-xx-xx	Netgear

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>`

Agenda

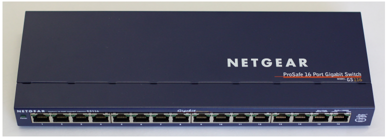
- Framing
  - Frame Detection
  - Ethernet (IEEE 802.3) Frames
  - WLAN (IEEE 802.11) Frames
- Addresses
- Switching
  - Devices
  - Forwarding
  - Loops

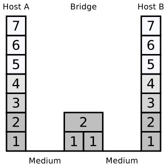
Agenda

- Framing
  - Frame Detection
  - Ethernet (IEEE 802.3) Frames
  - WLAN (IEEE 802.11) Frames
- Addresses
- Switching
  - Devices
  - Forwarding
  - Loops

Devices of the Data Link Layer: Bridges

- Remember: Devices of the Physical Layer increase the length of physical networks
- For connecting different physical networks, bridges are required
- A bridge has only 2 ports
    - They typically connect networks based on different technologies $\Longrightarrow$ see slides 39 and 40
- Bridges with $>$ 2 ports are called Switch





Virtual Bridges

Bridges can be virtualized in software (e.g., to connect virtual machines)

- Simple bridges forward all incoming frames
- Bridges and switches check the correctness of the frames via checksums
- They operate transparently

Example: 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: 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`

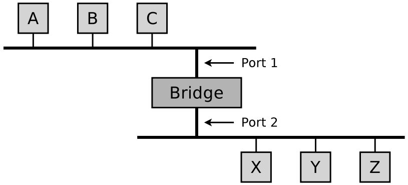
Interesting build instructions for your own laser bridge

- `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/`

Agenda

- Framing
  - Frame Detection
  - Ethernet (IEEE 802.3) Frames
  - WLAN (IEEE 802.11) Frames
- Addresses
- Switching
  - Devices
  - Forwarding
  - Loops

Learning Bridges (1/2)



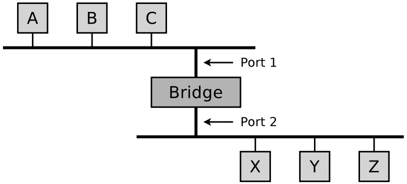
- Example: If a frame from participant B for participant A arrives at port 1, it is not required that the bridge forwards this frame via port 2

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
- As a consequence they maintain their forwarding tables themselves and automatically updates them based on received frames
- Administrators could maintain the tables inside the bridges, but this is mostly not required
  - Managed switches allows for advances configuration and fine-tuning

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 via which port



- 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))
- The forwarding table is not complete all the time
  - This is not a problem, because the table is only used for optimization
    - If no entry for a given address exists, the frame is typically sent on all ports

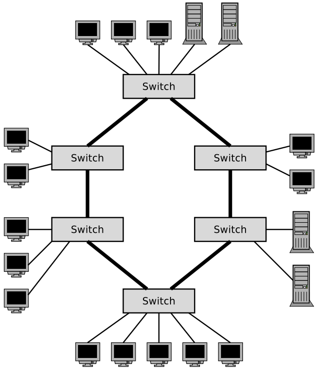
Forwarding Strategies

- A switch can implement different forwarding strategies:
  - Store-and-Forward
    The whole frame is received and buffered. After checking its integrity it is forwarded
  - Cut-Through
    As soon as the destination address field has been received, the frame is forwarded towards the receiver
  - Adaptive Cut-Through
    Cut-Through strategy is used unless a certain error threshold is reached ($\rightarrow$ store-and-forward)
  - Fragment-Free-Cut-Through
    If the first 64 bytes are received without an error, the frame is forwarded

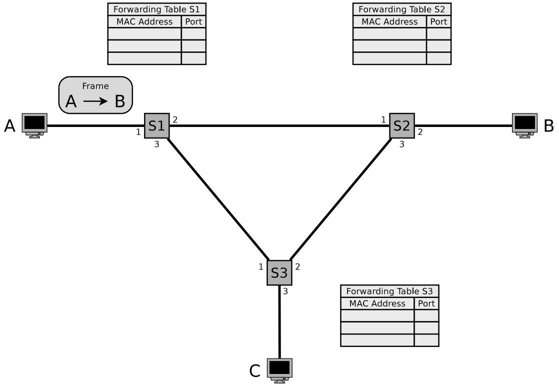
Agenda

Loops on the Data Link Layer

- Loops are a potential issue on the Data Link Layer
  - On the Data Link Layer only one path per destination should exist at one point in time
    - Otherwise frames get 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



Example of Loops in a LAN (1/6)

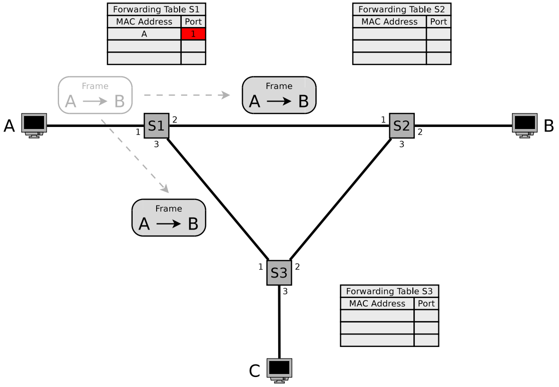


- A local area network has loops on the Data Link Layer
- The forwarding tables of the switches are empty
- Node A wants to send a frame to node B

Similar examples can be found here:

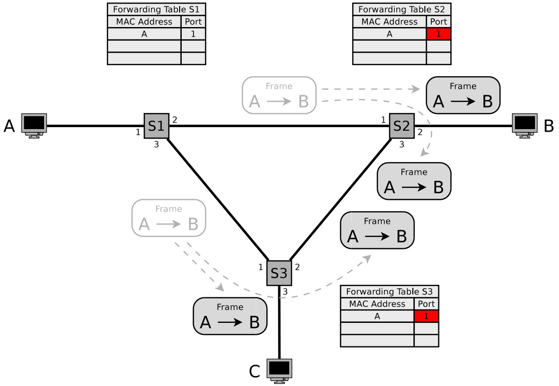
- *Olivier Bonaventure*. `http://cnp3book.info.ucl.ac.be/2nd/html/protocols/lan.html`
- *Rüdiger Schreiner*. Computernetzwerke. Hanser (2009)

Example of Loops in a LAN (2/6)



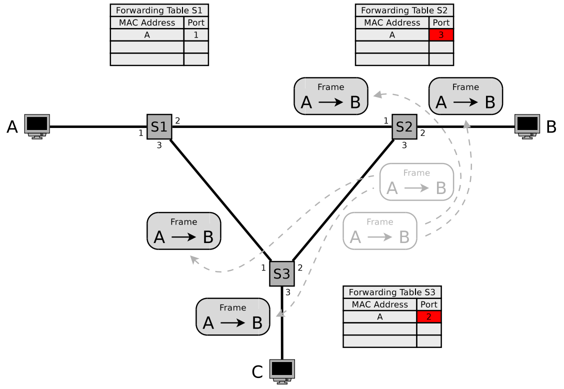
- The frame passes Switch 1
- Switch 1 stores the port to node A in its table
- Switch 1 don't know the path to node B
  - Therefore, it sends copies of the frame to all ports (except port 1)

Example of Loops in a LAN (3/6)



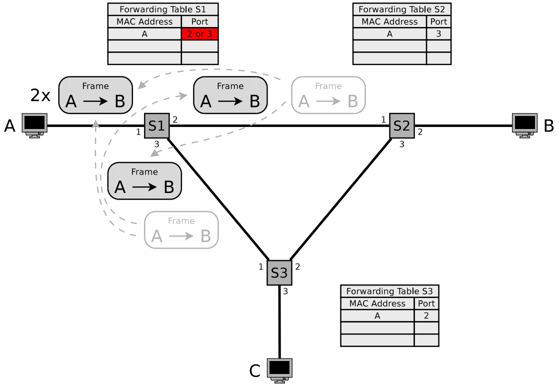
- The frame passes switch 2 and 3
- Switch 2 and 3 store the port to node A in their tables
- Switch 2 and 3 both do not know the path to node B
  - Therefore, Switch 2 and 3 both send copies the frame to all ports except to ones, where the frame reached Switch 2 and 3

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

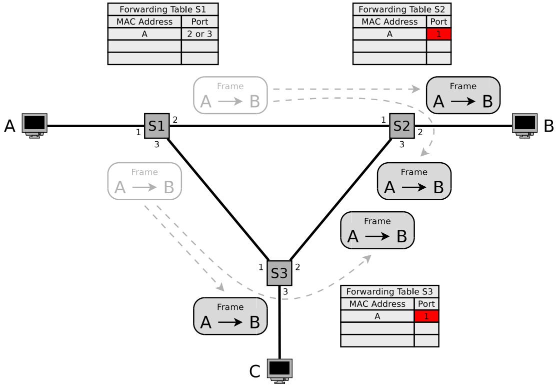
Example of Loops in a LAN (5/6)



- 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)



Forwarding Table S1

MAC Address	Port
A	2 or 3

Forwarding Table S2

MAC Address	Port
A	1

Forwarding Table S3

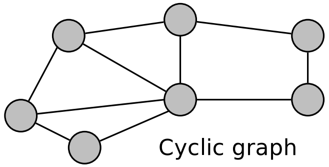
MAC Address	Port
A	1

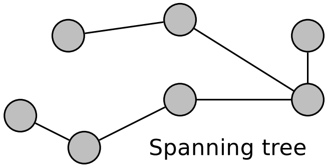
- Copies of the frame again pass Switch 2 and 3
- Switch 2 and 3 update their tables
- Ethernet does not contain any TTL or HopLimit
  - Therefore, this loop will not stop until the tables in the switches contain an entry for node B

- 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

Handle Loops in the LAN

- Bridges need to be able to handle loops
- Solution: create logical hierarchy

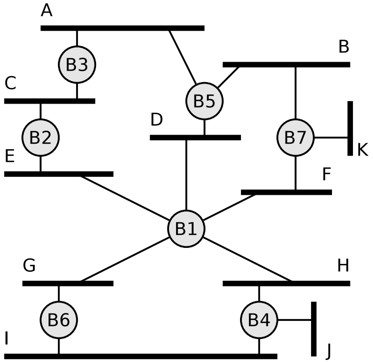




- A computer network, which consists of multiple physical networks, is a graph that may contain loops
  - The spanning tree is a subgraph of the graph that covers all nodes, but is cycle-free, because edges have been removed
  - The implementation of the algorithm is the Spanning Tree Protocol (STP)

Spanning Tree Protocol

Image source: Peterson, Davie. *Computernetze*



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

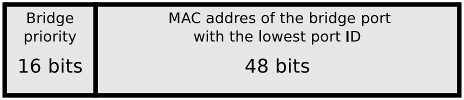
- The figure contains multiple loops
  - Via the STP, a group of bridges can reach an agreement for creating a spanning tree
    - By removing single ports of the bridges, the computer network is reduced to a cycle-free tree
- The algorithm works in a dynamic way
  - If a bridge fails, a new spanning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

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

Spanning Tree Protocol – Precondition (1/2)

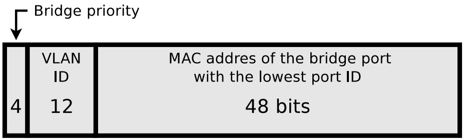
- For the functioning of STP, each bridge needs an unique identifier
  - Length of the identifier (bridge ID): 8 bytes
  - 2 different implementations of the bridge ID exist

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



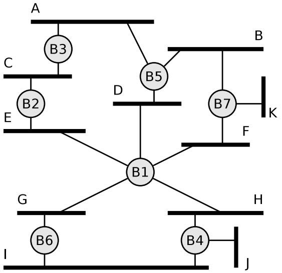
Spanning Tree Protocol – Precondition (2/2)

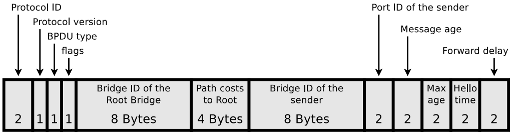
2. Cisco extension of the bridge ID, introducing the Extended System ID
   - Cisco supports bridges where each virtual LAN (VLAN) creates its own spanning tree
   - The original 2 bytes long part for the bridge priority is subdivided
     - 4 bits now represent the bridge priority
       $\Longrightarrow$ 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

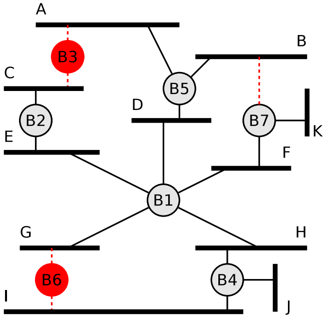
Spanning Tree Protocol – Functioning (2/2)

- 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



You should now be able to answer the following questions:

- What are the tasks of the Data Link Layer and what are the sublayers?
- Which mechanisms exist to detect the mark a frame?
- Which information does a frame contain?
- What are the properties of a MAC address and how do it look for IEEE 802 networks?
- How does switching/forwarding work?
- What is the problem of loops on the Data Link Layer and how can it be tackled?