






Computernetze 1 | CN 1

Zusammenfassung

1. NETWORK FUNDAMENTALS

				
Wireless Router	Lan Switch	Router	Multilayer Switch	Firewall Appliance

LAN: Local Area Network

Interconnect end devices in a limited area

Administered by a single organization or individual

Provides high-speed bandwidth to internal devices

WAN: Wide Area Network

Interconnect LANs over wide geographical areas

Typically administered by one or more service providers

Typically provides slower speed links between LANs

The **internet** is a worldwide collection of interconnected LANs and WANs. IETF was developed to help maintain structure on the internet.

- **Communications Fundamentals:** Devices must agree on “how” to communicate. There are three elements to any communication: Source (sender), Destination (receiver) and Channel (media)
- **Communication Protocols:** All communications are governed by protocols. They are the rules that communications will follow. These rules vary depending on the protocol.

2. THE LAYERED MODEL

2.1. BENEFITS

- **Assist in protocol design** because protocols that operate at a specific layer have defined information that they act upon and a defined interface to the layers above and below
- **Foster competition** because products from different vendors can work together
- **Prevent technology or capability changes** in one layer **from affecting** other layers above and below
- Provide a **common language** to describe networking functions and capabilities

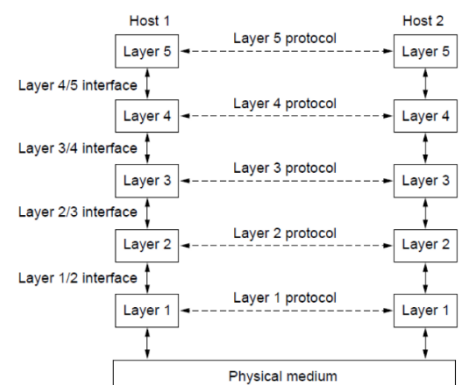
2.2. DISADVANTAGES

Performance suffers because of the many layers.

2.3. THE LAYERS

Data communication is a **complex interaction** of many components (Hardware, Software, Protocols).

- **Similar functions** should be in the same layer
- Independent functions in different layers
- well-defined **handover** between the layers
- each layer **receives a service** from the layer below it

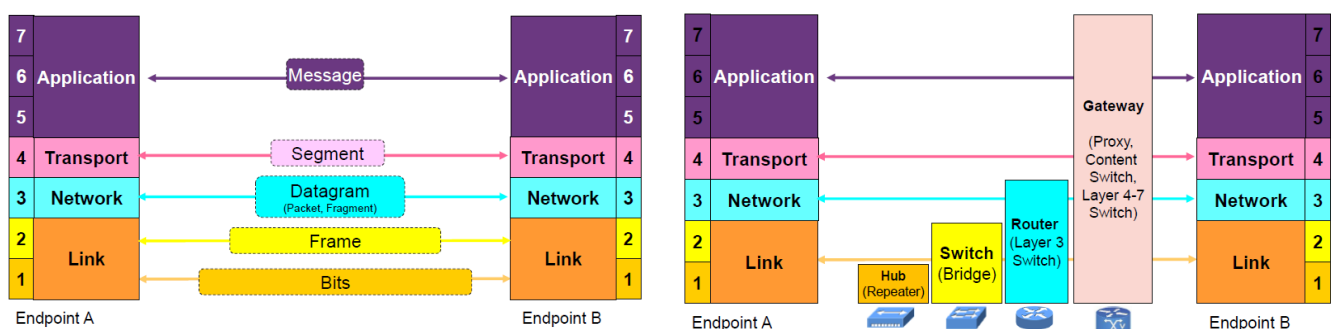


3. THE OSI REFERENCE MODEL | OSI MODEL

7 Application Layer (Anwendungsschicht)	Provides the <i>interface between applications</i> used to communicate and the underlying network.	Data. Mail (smtp / imap), FTP, TFTP, telnet, SSH, HTTP, DHCP, DNS, IP Telephony, Firewall (L7)
6 Presentation Layer (Darstellungsschicht)	Formats or presents data at the source device into a compatible format for recipient. Compresses data in a way that can be decompressed by the destination device. Encrypts data for transmission and decrypts upon receipt	Data. ASCII, JPEG, GIF, MP4, encryption
5 Session Layer (Sitzungsschicht)	Handles the <i>exchange of information</i> to initiate dialogs, keep them active, and to restart sessions that are disrupted or idle for a long period of time.	Data. SMB, iSCSI, interprocess communication, TLS, SSL
4 Transport Layer (Transportschicht)	Logical <i>end-to-end connection</i> between applications on different computers	Segments (TCP) / Datagrams (UDP). TCP, UDP, Segment, Firewall (L4)
3 Network Layer (Vermittlungsschicht)	Addressing end devices, encapsulation, routing , de-encapsulation. Responsible for delivering the IP packet from original source to destination.	Packets. IPv4, IPv6, ICMP, Routingprotokolle (RIP, OSPF, BGP etc), ping, Router, IP-Sec
2 Data Link Layer (Sicherungsschicht)	Elementary error detection mechanisms. Responsible for delivering the data link frame from one NIC to another on the same network. Protocols are defined by IEEE, ITU, ISO, ANSI	Frames. Ethernet, WLAN, PPP, LTE, Frame, MAC address, switch, network card, broadcast, ARP
1 Physical Layer (Bitübertragungsschicht)	Transmission of bit information via the physical medium	Bits. Encoding, modulation, cable, connector, bits, hub, collision, network card

4. THE TCP/IP REFERENCE MODEL

Layer	Description	OSI Model
Application Layer	Represents data to the user, plus encoding and dialog control	Application Layer Presentation Layer Session Layer
Transport Layer	Supports communication between devices across networks	Transport Layer
Internet Layer	Determines the best path through the network	Network Layer
Network Access / Link	Controls the hardware devices and media that make up the network	Data Link Layer Physical Layer



5. PHYSICAL LAYER / TRANSPORT LAYER / LAYER 1

Transports bits across the network media. Before any network communications can occur, a **physical connection** to a local network must be established. **A Network Interface Card (NIC) connects a device to the network.**

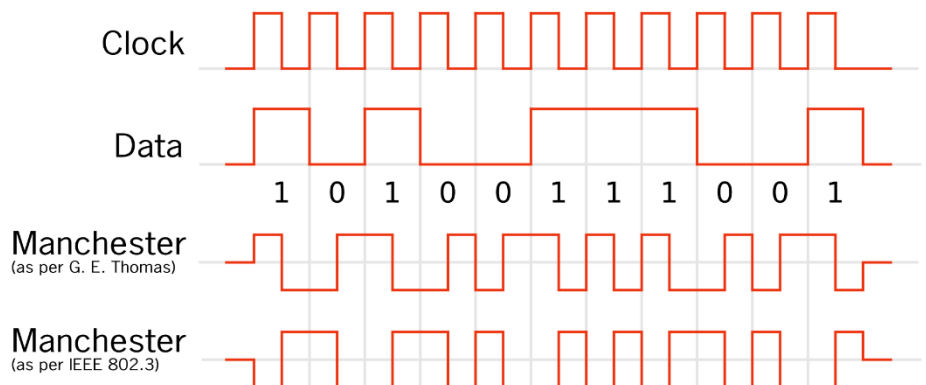
5.1. ENCODING

Messages sent across the network are **converted to bits**. The **bits are encoded into a pattern of light, sound, or electrical impulses**. Encoding converts the stream of bits into a format recognizable by the next device in the network path. This “coding” provides predictable patterns that can be recognized by the next device.

5.1.1. Manchester

In der **Codedefinition nach G.E. Thomas**, auch bezeichnet als Biphase-L oder Manchester-II, bedeutet eine fallende Flanke eine logische Eins, eine steigende Flanke eine logische Null.

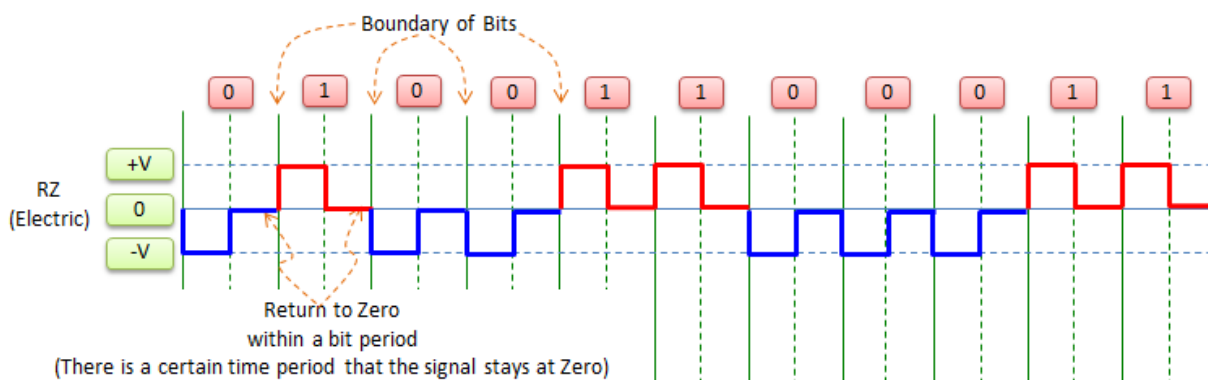
In der **Codedefinition nach IEEE 802.3**, wie sie bei 10-Mbit/s-Ethernet verwendet wird, bedeutet eine fallende Flanke eine logische Null und eine steigende Flanke eine logische Eins.



Advantages: free of DC components (no zero), possible to transmit via pulse transformers with galvanic isolation, self-clocking. **Disadvantages:** Requires twice as much bandwidth as binary coding, because 1B2B coding. *Two codebits are required to encode one user data bit, bit rate only half the baud rate (symbol rate in transmission).*

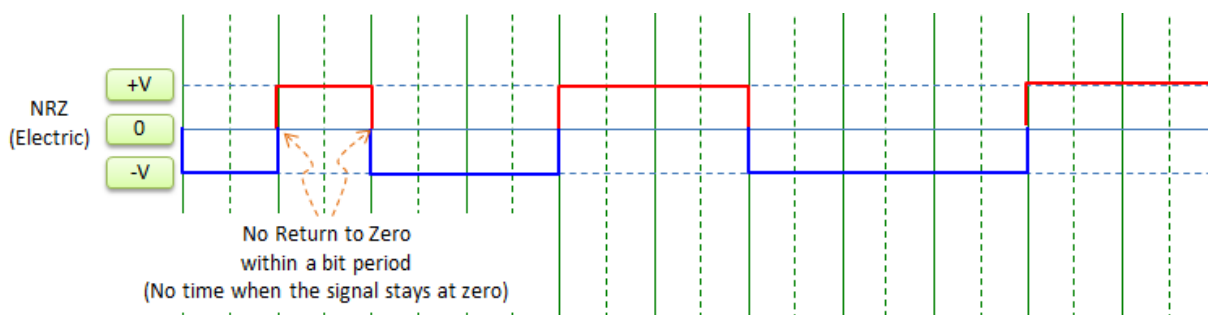
5.1.2. RZ (Return to Zero)

Refinement of NRZ, self-clocking



5.1.3. NRZ (No Return to Zero)

Simplest coding, not self-clocking



5.1.4. 8b/10b (Clock recovery)

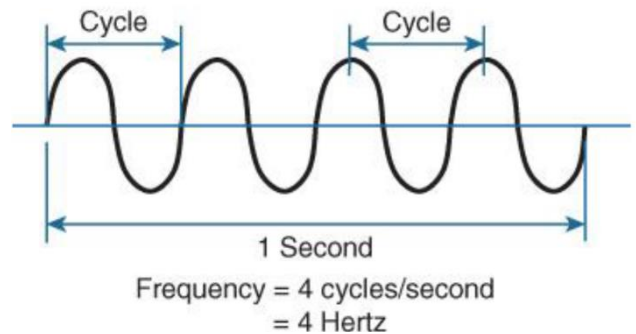
Maps 8-bit words to 10-bit symbols – prevents too many zeros or ones in a row.

5.2. WIRELESS

Sender (transmitter) sends an alternating current into a section of wire (an antenna), which sets up moving *electric and magnetic fields that propagate out and away from the antenna as traveling waves*.

5.2.1. Frequency

Hertz (Hz) corresponds to the number of cycles per second



5.2.2. Channels

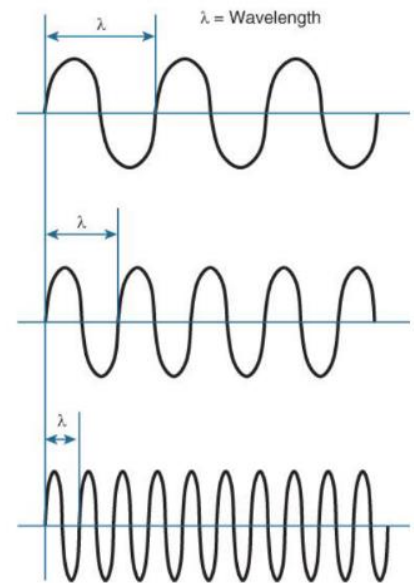
2.4 GHz – between 2.400 and 2.4835 GHz

5 GHz – between 5.150 to 5.350 and 5.470 to 5.825 GHz
(Radar between 5.350 and 5.470)

Bandwidth – Width of frequency space required within the band

5.2.3. Wavelength

- Wavelength is a *measure of the physical distance that a wave travels* over one complete cycle
- RF (Radio Frequency) waves travel at a constant speed, *slightly less than speed of light* (in vacuum speed of light)
- *Wavelength decreases as the frequency increases*. As the wave cycles get smaller, they cover less distance. *5GHz less range than 2.4GHz*



5.2.4. Power and dB

- **Logarithmic function** to transform exponential ranges into linear ones
- dBm (ratio to 1mW / ratio to 1 milliWatt)
- **Law of 3s**: 3dB more: double the power, 3dB less: half the power
- **Law of 10s**: 10dB more: 10x the power, 10dB less: 1/10x the Power
- **SNR**: Difference between noise and our signal
- **Receiver Sensitivity**: Up to which level signals can be received
- **100mW = 20dB** (10mW = 10dB)

$$\text{decibels} = 10 * \log_{10} (\text{milliwatts}), \quad \text{milliwatts} = 10 * \frac{\text{decibels}}{10}$$

5.2.5. Modulation

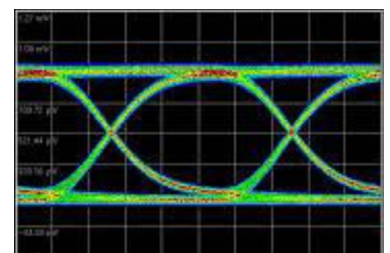
Altering the carrier Signal. Attributes: Frequency, Phase and Amplitude

5.3. OPTICAL NETWORK

Single-Mode Fiber	Multimode Fiber
Very <i>small</i> core	<i>Larger</i> core
Uses <i>expensive</i> lasers	Uses <i>less expensive</i> LEDs
<i>Long-distance</i> applications	LEDs transmit at different angles
	Up to 10 Gbps over 550 meters

5.3.1. Eye Diagram

- An eye diagram shows a *relative performance* of the signal
- The opening of the eye provides valuable information about the ability of the receiver to detect the signal correctly
- For a good transmission system, the eye opening should be as wide and open as possible
- Horizontal shift is called jitter, which can be caused by imprecise clocks



5.3.2. Attenuation in Fiber (Possible Interferences)

Absorption, Scattering of the light, Microbends (small distortions in manufacturing), Macrobends (wrapping fiber around a corner), Back reflections, Fiber splices, Mechanical connections.

5.3.3. Types of Dispersion (Spreading of the light pulse)

- **Chromatic Dispersion:** Different wavelengths travel at different speeds
- **Polarization Mode Dispersion (PMD):** Single-mode fiber supports two polarization states, fast and slow axes have different group velocities

5.3.4. Regeneration (fix the dispersion)

- **Re-amplifying** makes the analog signal stronger / the light brighter
- **Reshaping** restores the original pulse shape
- **Retiming** restores the original timing between the pulses

5.4. CALCULATIONS

5.4.1. Speed of Signals

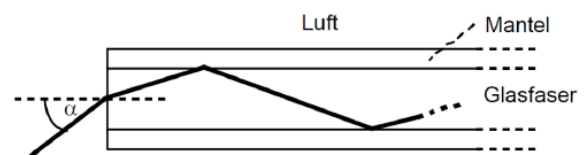
In fiber glass, signals travel about **2/3 of the speed of light** (200'000km/s).

The Time a Signal needs is calculated as follows:

$$T(s) = \frac{\text{Length of cable the Signal needs to travel (km)}}{\text{Speed of the Signal (km/s)}}$$

5.4.2. Maximum possible entry angle

$$\alpha \leq \arcsin\left(\frac{n_{\text{Glas}}}{n_{\text{Luft}}} \sqrt{1 - \frac{n_{\text{Mantel}}^2}{n_{\text{Glas}}^2}}\right)$$



How does Alpha behave in relation to the refraction

indexes? If Alpha becomes larger, the term within arcsin should also be chosen as large as possible.

5.4.3. Optical budget

Can be calculated like this: *Transmissionpower – Receiversensitivity*

6. ETHERNET (WIRED) / LAYER 2

6.1. ETHERNET OVERVIEW

Ethernet **specifies and implements encoding and decoding schemes** that enable frame bits to be carried as signals across both **copper and fiber cables**. Ethernet separates the functions of the data link layer into two sublayers:

Logical Link Control and **Media Access Control**.

Logical Link Control (LLC)	Media Access Control (MAC)
Handles communication between the network layer and the MAC sublayer. Provides a way to identify the protocol that is passed from the data link layer to the network layer.	Data encapsulation: Includes frame assembly before transmission, frame parsing upon reception of a frame, data link layer MAC addressing and error detection. Media Access Control: Ethernet is a shared media and all devices can transmit at any time.

6.2. LEGACY ETHERNET TECHNOLOGIES (OUTDATED)

6.2.1. The Bus

10BASE-2, 10BASE-5 Transmits 10 Mbps over a single coaxial cable bus. A coaxial cable connects each device on the ethernet network and creates an electrical circuit called a bus.

A bus is a broadcast and collision domain.

6.2.2. The Hub

10BASE-T uses cheap and easy to install Unshielded Twisted Pair (UTP) copper cable. UTP cables connect each device on the hub and creates an electrical circuit called a bus.

A hub creates a broadcast and collision domain.

6.2.3. Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

Defines how the Ethernet logical bus is accessed. It is in effect within a collision domain and if a device's network interface card (NIC) is operating in *half-duplex* mode. It helps prevent collisions and defines how to act when a collision does occur.

- **Carrier Sense:** Listen to the medium
- **Multiple Access:** Sending if medium is free, else waiting for a random time and try again
- **Collision:** The amplitude of the signal increases because a collision occurs.
- **Collision Detection / Backoff algorithm:** The nodes stop transmitting for a random period of time, which is different for each device.

After **16 tries**, the **host gives up** the transmission attempt and discards the frame. The network is overloaded or broken.

6.2.3.1. What happens when a collision occurs?

- A **jam signal** informs all devices that a collision occurred.
- The collision invokes a **random backoff algorithm**.
- Each device on the Ethernet segment **stops transmitting** until their **backoff timers expire**.
- All hosts have **equal priority** to transmit after the timers have expired.

6.2.3.2. Effects of heavy collisions:

Delay, Low throughput, Congestion

6.2.4. Full Duplex Ethernet

Requires **point-to-point connection** where only **two nodes** are present. The data is sent on a different set of wires than the received data, so **no collisions will occur**. When a NIC detects that it can operate in full-duplex mode, CSMA/CD is disabled.

6.3. CURRENT ETHERNET TECHNOLOGIES (BROADCAST & COLLISION DOMAINS)

6.3.1. Bridges/Switches/Routers

Routers, **Switches** and **Bridges** separate LANs into separate **collision domains**.

Routers separate LANs into multiple **broadcast domains**.

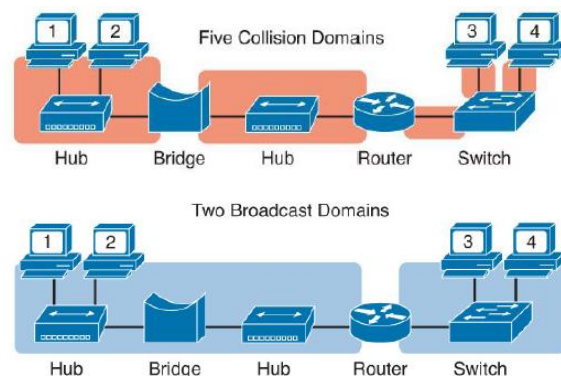
6.3.2. Standards

Each new physical layer standard from the IEEE requires differences at the physical layer.

Name	Speed	Alternative Name	Name of IEEE Standard	Cable Type, max length
Ethernet	10Mbps	10BASE-T	802.3	Copper, 100m
Fast Ethernet	100Mbps	100BASE-TX	802.3u	Copper, 100m
Gigabit Ethernet	1000Mbps	1000BASE-LX	802.3z	Fiber, 550m
Gigabit Ethernet	1000Mbps	1000BASE-T	802.3ab	Copper, 100m
10GigE	10Gbps	10GBASE-T	802.3an	Copper, 100m

6.3.3. Power over Ethernet

Power over Ethernet is a standard *that provides direct current* (DC) **electrical power** over Ethernet twisted-pair cabling. **Does not work with Fiber**.



6.4. UTP CABLING

UTP cablings include either *two or four pairs of wires*. The cable ends typically in an RJ-45 connector. The RJ-45 connector has eight specific physical locations into which the eight wires in the cable can be inserted, called pin positions. *Transmits on Pins 1,2*: PC NICs (Clients), Routers, Wireless AP *Transmits on Pins 3,5*: Hubs, Switches

There are two different Cables:

Straight through: all pins connect to the same pins on the other side

Crossover: pins get switched on the way. 1, 2 switch to 3, 6 and 3, 6 switch to 1, 2.

Newer cisco switches have a feature called *auto-mdix* that notices when the wrong cable is used and automatically changes its logic to make the link work.

6.4.1. Auto-negotiation

The speed, duplex and pins information are negotiated between 2 Ethernet equipment's.

Operation of Auto-Negotiation: Each device advertises its capabilities to the link partner. The protocol selects the highest common denominator between the devices.

6.5. ETHERNET ADDRESSING (MEDIA ACCESS CONTROL – MAC)

MAC Addresses are 6-byte-long (48bit-long) binary numbers. Is represented as a 12-digit hexadecimal number.

00:0D:88:3C:30:F9 or 000D.883C.30F9 00000000 00001101 10001000 00111100 00110000 11111001

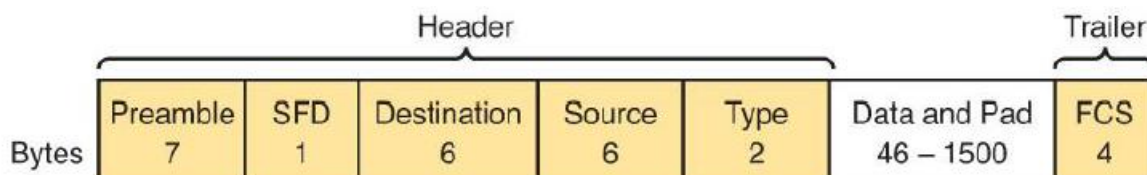
Organizationally Unique Identifier (OUI), first part: 00 0D 88

Vendor Assigned (NIC Cards, Interfaces), second part: 3C 30 F9

Unicast	Broadcast	Multicast
Frame is sent <i>directly</i> to MAC address of Host 2 (in same LAN)	Frame is sent to the broadcast address (FFFF.FFFF.FFFF) – <i>every station in the same Broadcast domain receives the frame</i> . This is called <i>flooding</i> . In large flat networks, <i>broadcast storms</i> can happen. Because of that it is recommended to use <i>very small broadcast domains</i> .	Frame is <i>received by a group</i> of devices which belong to the same multicast group. (DA MAC Address 01-00-5E-00-00-00 if IPv4 and DA MAC Address 33-33-XX-XX-XX-XX if IPv6)

6.6. ETHERNET FRAMING

6.6.1. Ethernet II Frame



Field	Field Length	Description
<i>Preamble</i>	7 bytes	Synchronization
<i>Start Frame Delimiter (SFD)</i>	1 byte	Signifies that the next byte begins the Destination MAC Address field
<i>Destination MAC Address</i>	6 bytes	Identifies the intended recipient of this frame
<i>Source MAC Address</i>	6 bytes	Identifies the sender of this frame
<i>Type</i>	2 bytes	Defines the type of protocol listed inside the frame; today, most likely identifies IP version 4 or 6
<i>Data and Pad</i>	46-1500 bytes	Holds data from a higher layer, typically an L3PDU (IPv4 or IPv6 packet). The sender adds padding to meet the minimum length requirement for this field.
<i>Frame Check Sequence (FCS)</i>	4 bytes	Provides a method for the receiving NIC to determine whether the frame experienced transmission errors

Type: Some common EtherTypes:

- **IPv4:** 0x0800
- **IPv6:** 0x86DD
- **802.1Q:** 0x8100
- **MPLS:** 0x8847
- **ARP:** 0x0806
- **LLDP:** 0x88CC
- **MACSec:** 0x88E5

FCS: Error Detection with Frame Check Sequence

- The sender **applies a math formula** to the frame before sending it, storing the result in the FCS field.
- The receiver **applies the same math formula** to the received frame and compares with the sender's result.
- If the results are the same, the frame did not change. If the **results are different**, an **error occurred**, and the receiver **discards the frame**. The Ethernet device does **not attempt to recover** the lost frame.

6.7. BENEFITS OF USING BRIDGES/SWITCHES

A **hub is not intelligent**. For more intelligent stuff, we need bridges and switches.

6.7.1. Bridge

The **purpose of a bridge is to manage traffic on a LAN**. It makes its decisions based on MAC addresses. The bridge is a **layer 2** device. It has **only two ports**. Bridges remember the source MAC address of each frame received on a port and enter this information into the **MAC table** called a **forward/filter table**. Bridges split a LAN into **several smaller collision domains** (but not broadcast domains) and therefore reduce the LAN traffic. MAC table has a **timeout of 5min / 300s**. If it does not hear from a source in this time, its entry in the table gets deleted.

6.7.2. Switch

The **switch is like a bridge** but with several ports. All ports are **full-duplex**, so no problems with collisions.

Flooding

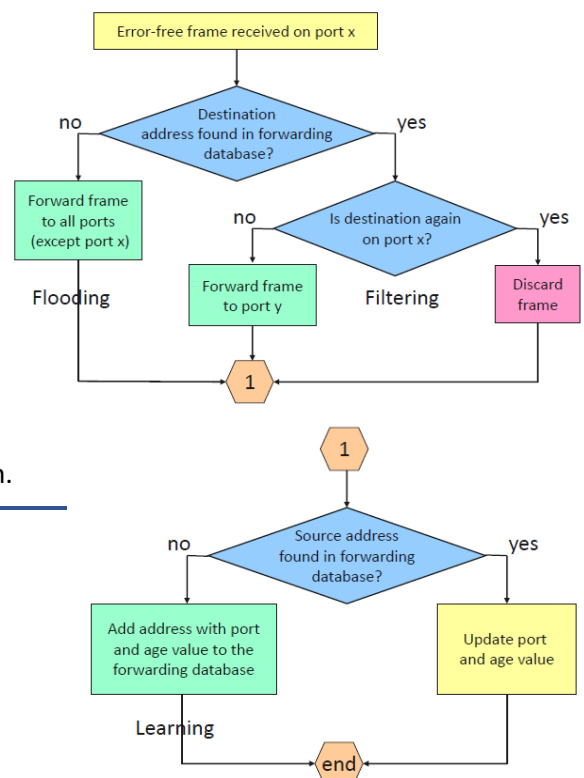
When a bridge gets a data packet, and it did **not know** the DA, it **floods the information** to all ports but the one where it received the data. *Unicast flooding*

Filtering

When a bridge gets a data packet, and **already knows** that the DA is on the same port as the SA, it **filters** the information and does **not flood** it, because the other bridges do not need to know. This reduces traffic.

Forwarding

If the destination MAC address comes from another port within the switch, then the frame is **sent to the identified port** for transmission.



6.8. ADDRESS RESOLUTION PROTOCOL (ARP)

Why does a client know the MAC addresses of the other clients? It knows the IP, but not the MAC address.

For this, the ARP is needed:

- PC A **sends a broadcast**: "Who has the IP 10.10.10.30?"
- The ARP Request is **flooded**
- The PC with the **sought IP sends his ARP Reply** "I have the IP, here is my MAC Address". This is sent as a **unicast** because the Switch already knows PC A.
- Now the **PC A knows the MAC address** of 10.10.10.30 and can send its Packet.

6.8.1. ARP and Default gateway

A host compares the destination IPv4 address and its own IPv4 address to **determine if the two IPv4 addresses are located on the same Layer 3 network**. If the destination host **is not on the same network**, the source **checks its ARP table** for an entry with the IPv4 address of the default gateway. If there is **no entry**, it uses the **ARP process** to

determine a MAC address of the default gateway. *Ethernet devices also maintain an ARP table* (also called ARP cache). Entries in the ARP table are time stamped and can time out.

6.8.2. ARP Spoofing

Attackers can *respond to ARP requests* (e.g. for the *default gateway*) and *pretend to be providers* of services.

6.9. BANDWIDTH

The *capacity* at which the *medium can carry data*. Digital bandwidth measures *how many bits can be transmitted in a second*. Physical media properties, current technologies and the laws of physics play a role in determining available bandwidth.

Unit of Bandwidth	Abbreviation	Equivalence
Bits per second	Bps	1bps = fundamental unit
Kilobits per second	Kbps	1 Kbps = 1'000 bps = 10^3 bps
Megabits per second	Mbps	1 Mbps = 1'000'000 bps = 10^6 bps
Gigabits per second	Gbps	1 Gbps = 1'000'000'000 bps = 10^9 bps
Terabits per second	Tbps	1 Tbps = 1'000'000'000'000 bps = 10^{12} bps

7. ETHERNET (WIRELESS) / LAYER 2

MAC in Layer 2 is different in wireless than in wired.

7.1. CHANNEL BONDING

Two or more adjacent channels within a given frequency band are *combined to increase throughput* between two or more wireless devices.

7.2. CSMA/CA (CARRIER-SENSE MULTIPLE ACCESS WITH COLLISION AVOIDANCE)

In wireless, it is also possible to have *collisions*, because it is a *shared medium*.

- Client *sends an RTS* (request to send) "Can I send for xy time?"
- Access point *answers with a CTS* (clear to send), which *all connected devices get*. "Access Point XY is now sending for xy amount of time (minus the time for the RTS)"
- Transmission

7.2.1. Hidden Node

It is *not possible to use CSMA/CD* because we do not know if everyone receives everything. If there is a *wall between to clients* for example, the *clients do not know if the other is sending at the same time*.

7.2.2. Distributed Coordination Function (DCF)

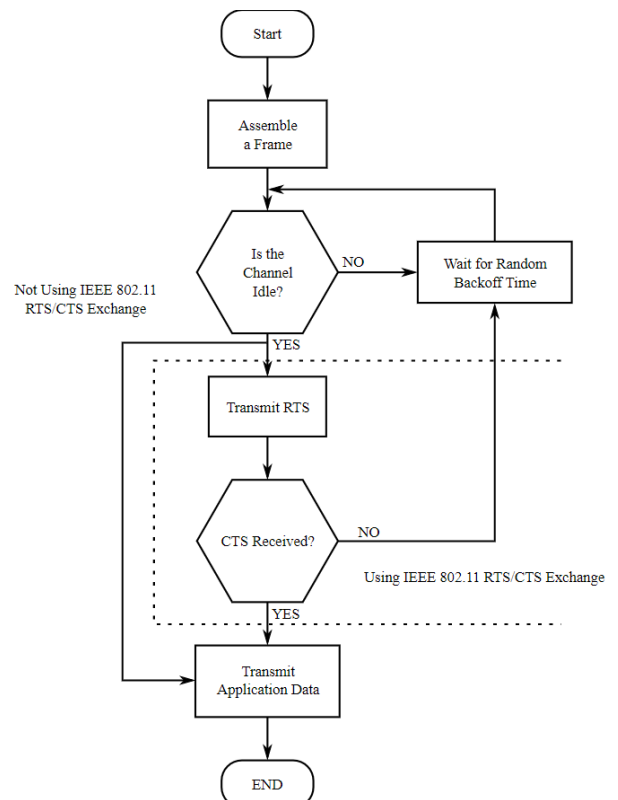
Function which creates the backoff time for CSMA/CA. *CTS, ACK and Block ACK (SIFS)* have the *highest* priority and the shortest backoff time. *PIFS* have a *middle* priority and *DIFS* the *lowest*.

7.2.3. Network Allocation Vector (NAV)

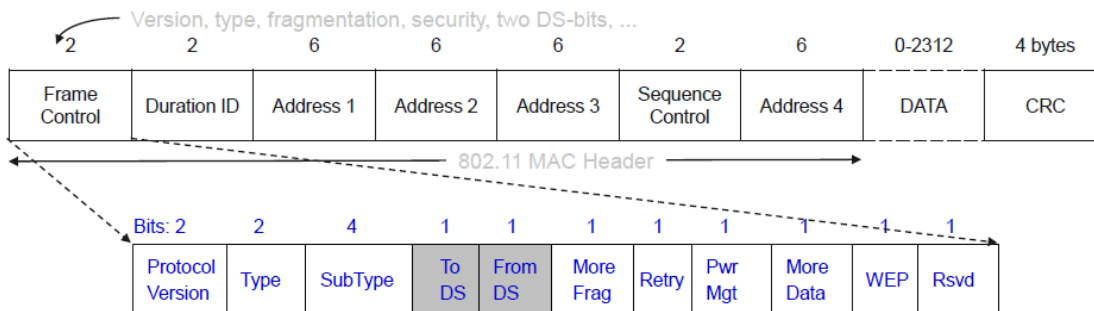
Listening Stations can mark the medium as busy with the Network Allocation Vector (NAV), while another station is sending.

7.2.4. Acknowledge (ACK)

Receiver sends Acknowledge when he received the packet with a correct checksum (CRC). With no Ack, the packet will be transmitted again.



7.3. 802.11 FRAME STRUCTURE



7.3.1. Address Fields

Scenario	To DS	From DS	Address 1	Address 2	Address 3	Address 4
Ad-hoc network	0	0	DA	SA	BSSID	-
From AP	0	1	DA	BSSID (AP)	SA	-
To AP	1	0	BSSID (AP)	SA	DA	-
Within DS	1	1	RA (AP 1)	TA (AP 2)	DA	SA

AP: Access Point | **BSSID:** Basic Service Set Identifier | **DS:** Distribution System | **DA:** Destination Address | **RA:** Receiver Address | **SA:** Source Address | **TA:** Transmitter Address

- **Ad-hoc network:** Network between wireless devices without an access point. Needs three Addresses: DA, SA and BSSID (The MAC address of the next hop (Access Point))
- **From AP:** If a Packet is sent from an Access Point to a device, the Destination Address stays, but the source address is now the address of the access point. The old source address is new in the address 3 field.
- **To AP:** If a Packet is sent to an Access Point (for example replay to the "from AP" message), the DA is the BSSID, the Source address stays and the real DA is in the address 3 field.
- **Within DS:** If a packet gets forwarded between two Access Points, the first two addresses are from the access points, and the original DA and SA are in the fields 3 & 4.

7.4. MANAGEMENT FEATURES

7.4.1. Beacon

Is needed to know which BSSIDs are available. All Access Points (AP) send beacons to advertise their BSSID.

BSSID: Every AP has a BSSID. **ESSID / SSID:** Every WLAN has an ESSID. In an SSID there can be multiple BSSIDs. Probe Request & Probe Response same, but queried (request a beacon).

7.4.2. Association / Reassociation (Active Scanning ex.)

How does a client connect to an AP?

1. Client **sends Probe**
2. AP Sends **Probe Response**
3. Client **selects best AP**
4. Client sends **auth request** to selected AP
5. AP **confirms authentication** and registers client
6. Client **sends association request** to selected AP
7. AP **confirms association** and registers client

7.5. ROAMING

Switching to another AP with better signal strength.

A client is connected to an AP. If there is an AP that is at least say 10dB better and the signal strength of the current AP is below a limit of say 75dB (handoff threshold), a handover occurs.

1. Station **sends probe**
2. AP sends **Probe response**
3. Client **selects best AP**
4. Client sends a **reassociation request** to the new AP
5. New AP sends a **reassociation response**
6. Client **sends a disassociation request** to the old AP
7. The old AP sends the **unacknowledged data to the new AP**, using the inter Access Point Protocol (802.11f)

Roaming *usually takes (too much) time* because of the many steps listed above.

There are ways to *improve* roaming, for example with direct handover from AP to AP without re-authentication (802.11r).

7.6. WIRESHARK

Most WLAN-adapters *only record frames that belong to my Access Point* and are directed to my MAC-Address. So that all WLAN-frames and WLAN-header can be displayed, the WLAN-Adapter would have to support the so-called *Radio Frequency Monitor Modus* (RF-MON). ICMP packets get displayed twice in a Wireshark trace, because the sniffer records first the frame from the station to the Access Point and then from the Access Point to the station.

8. SPANNING-TREE, VLANS AND TRUNKING / LAYER 2

8.1. THE NEED FOR SPANNING TREE

Bridging loops occur any time there is a redundant path and when the topology is connected back to itself. Ethernet does not have *any built-in protection* and it cannot prevent frames from continuously circulating when a loop is present (Broadcast Storm). *Spanning-Tree creates a Loop-free network*. It is active by default.

8.2. IEEE 802.1D OVERVIEW (SPANNING TREE)

STP uses messages called bridge protocol data units (BPDU):

- SA: Source MAC address of the port
- DA: Multicast Address 01-80-c2-00-00-00 (0180.c200.0000)

There are two types of BPDUs:

- Hello or configuration BPDU *sent by the root bridge*
- Topology Change Notification (TCN) BPDU *Sent by a different switch to the root*

8.2.1. The Algorithm

- | | |
|---------------------------------|-----------------------------------|
| 1. Lowest <i>bridge ID</i> | 3. Lowest sender <i>bridge ID</i> |
| 2. Lowest <i>root path cost</i> | 4. Lowest sender <i>port ID</i> |

8.2.1.1. Election of a root bridge ("root")

There is one root bridge per network.

The Bridge with the *lowest Bridge ID* is the root bridge.

The Bridge ID is created from the Bridge Priority (2 bytes, default: 32'768) and the MAC address (6 bytes).

8.2.1.2. Election of the root ports ("RP")

There is one root port per nonroot bridge.

The port with the *lowest cost to the root bridge* is the root port. Traffic on its way to the root switch *flows out of root ports*.

10BaseT / Ethernet: Cost 100 , 100BaseT(x) / Fast Ethernet: Cost 19 , 1000BaseT / Gigabit Ethernet: Cost 4

If two ports have the same cost, the Port ID decides.

The Port ID is created from the port priority (4 Bits) and the port number (12 Bits).

8.2.1.3. Set the designated ports ("DP")

There is one designated port per segment. DPs are in forwarding state.

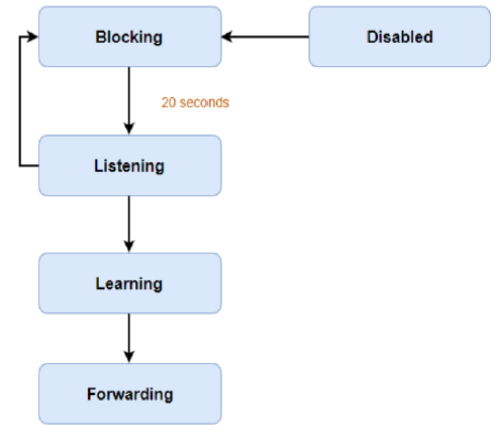
- The ports on the *opposite end of the root ports* are automatically designated ports.
- Traffic on its way to the root switch *flows into* the designated ports.
- *All ports on the root bridge* are designated ports.
- For the other ports, the *one with the lower root path cost* gets assigned as designated port.
- If the ports have the same cost, the one on *the Bridge with the lower ID* gets selected.
- If the ports are on the same bridge and in the same direction, the one with the *lowest port ID* on the opposite gets selected.

8.2.1.4. Block all other ports ("|")

All other ports are blocked. (The ports, on which the traffic would flow out of on its way to the root port)

8.2.2. Port States

- **Blocking (stable):** Does not transmit or forward data frames
- **Listening:** The port receives BPDUs but does not forward data frames. Port starts sending BPDUs
- **Learning:** Like listening, except that the port is MAC learning
- **Forwarding (stable):** Forwards data frames. The port receives and processes BPDUs
- **Shutdown/Disabled:** The port has been administratively shutdown



8.2.3. Timers

Hello: Controls the rate at which configuration BPDUs are issued from the root switch. By default, 2 seconds.

Max age: If the age of the last BPDU exceeds the max age timer value (20 seconds), the comparison algorithm will have to be rerun. The receiver has to find a new path to the root. The timer is reset every time new information is received via a BPDU.

Forward delay: This timer monitors the time spent in each of the transitional port states. By default, 15 seconds.

When plugging in a computer and receiving a link light, the ports come up blocking, then listening for 15 seconds, then learning for another 15 seconds, and finally forwarding (30 seconds in total).

8.2.4. Topology changes

When a topology change occurs (expiration of the max age timer, addition or removal of a switch, links going up / down, receipt of new information via a BPDU), **the switch sends a TCN BPDU out its root port to the root bridge**. The BPDU carries **no data** about the change. Once the root bridge received the BPDU, the root bridge sets the **Topology Change flag** in its Configuration BPDU. This causes all other bridges to **shorten their bridge table aging times** from the default (300 seconds) to the forward delay value (default 15 seconds), so the table gets deleted.

- **The duration of Direct Topology Changes** is 2x forward delay period: **30 seconds**
- **The duration of Indirect Topology Changes** is the max age timer (20s) plus the time until the next Configuration BPDU was received (2s) plus the time the port spends in the listening and learning states (forward delay period, 30s): **52 seconds**.
- **Insignificant Topology Changes:** No BPDUs are sent – no delay, the port is brought right into the Forwarding state. (Only when the STP PortFast feature is enabled)

8.3. PORT AGGREGATION WITH ETHERCHANNEL (LACP)

If you have **multiple parallel links** between two switches, STP **blocks** all the links except one.

EtherChannel avoids the possibility of bridging loops with the **Link Aggregation Protocol (LACP)** by **bundling parallel links into a single logical link**. STP is aware of the single **port channel** interface, which is kept in the Forwarding state. A set of up to **16 potential links** can be defined for each EtherChannel. A switch **selects two to eight** of these having the **lowest** port priorities as **active** EtherChannel links at any given time. The other links are placed in a **standby state** and will be enabled in the EtherChannel if one of the active links goes down. Failover in less than a few milliseconds. The load is **not balanced equally** across all the links. Frames are forwarded on a specific link as a result of a **hashing algorithm**. The hash algorithm computes a binary pattern that selects a links number in the bundle to carry each frame. The links must have the **same bandwidth** and the if possible have the same length to be bundled.

LACP active mode: the switch actively asks a far-end switch to negotiate an EtherChannel

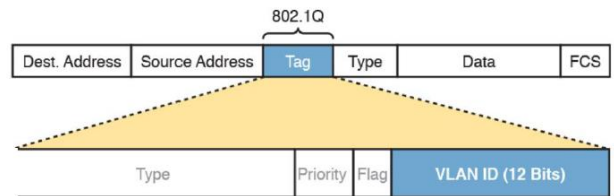
LACP passive mode: the switch negotiates an EtherChannel only if the far end initiates it.

8.4. VLANS

In a traditional LAN segmentation, one needs two different Ethernet LAN switches to create two different LANs. With virtual LANs (VLAN) it is possible to create two broadcast domains (and spanning-tree domain) with a single switch.

8.4.1. Trunk (IEEE 802.1Q)

If several switches in several VLANs are to communicate with each other, they would have to have a *separate connection for each VLAN* without VLAN trunking. This quickly becomes impractical if there are many VLANs. *With VLAN trunking, all data can be sent over one link.*



In an Ethernet frame, 802.1Q adds a *4-byte tag* just after the Source Address field. This tag is only used between the switches. If *no tag* is used, the frame belongs to the *native* VLAN.

- **Type:** 2 bytes, value of 0x8100 indicates a trunking tag.
- **Priority:** 3 bits, is used to implement class of service (CoS) functions
- **VLAN identifier (VID):** 12 bits. Can have values from 0 to 4095, but VLANs 0, 1 and 4095 are reserved.

8.4.2. Per-VLAN Spanning Tree (PVST)

PVST operates a *separate instance of STP* for each individual VLAN. This allows the STP on each VLAN to be configured independently. Each BPDU contains the VLAN ID. *Useful for load balancing.*

8.4.3. InterVLAN Routing

Devices in different VLANs are in different IPv4 subnets. *Layer 2 switches will not forward data between two VLANs.*

When configured with some ports in VLAN 10 and others in VLAN 20, the *switch acts like two separate switches* in which it will forward traffic. The job of *forwarding data* into and out of a VLAN falls to *routers (Layer 3)*. With an interface connected on the router in each VLAN, the *router can route IP packets between the subnets*. One physical interface is used per VLAN. The preferred option is to use a *VLAN trunk* between the switch and router, requiring only one physical link between a router and a switch, or even between server hardware and a switch. *This design is called a router-on-a-stick.*

Layer 3 switches can be configured to act only as a Layer 2 switch, or they can be configured to do both Layer 2 switching as well as Layer 3 *routing*. Today Layer 3 switches are widely used to route packets between subnets (VLANs).

9. IPV4 / LAYER 3

9.1. DATA TRANSFER IN THE NETWORK

9.1.1. Network addresses

Network source IP address: The IP address of the sending device, the original source of the packet.

Network destination IP address: The IP address of the receiving device, the final destination of the packet.

Data Link Addresses: As the IP packet travels it is encapsulated in a new data link frame by each router.

9.1.2. Devices on the same network

When the devices are on the same IP network, *no routing* is necessary. The destination MAC address of the frame is the MAC address of the final destination. The IP Address is irrelevant if on the same network.

9.1.3. Devices on a remote network

The source and destination IP addresses are on different networks. The frame is sent to the router (*default gateway*). The destination MAC address of the frame is the MAC address of the interface on the router.

9.2. NETWORK LAYER PROTOCOL

The network layer provides services that provide *end-to-end transport* across a network.

Addressing of end devices, Encapsulation, Routing, De-encapsulation.

IP is a **connectionless** protocol. The packet is sent to the destination **without prior establishment** of a connection. Senders do not know whether the destination is present, reachable, or functional before sending packets. – **Best effort**. Focus on pathfinding, not reliability. IP knows where the packet comes from and where it is headed. IP is **media independent**, does not care if sent over copper, fiber or wireless.

MTU: Maximum Transmission Unit. How big packets can be transmitted over this media?

9.2.1. IPv4 header

Total size: 20 bytes

<- 4 Bytes ->

Version	Length	DS Field	Packet Length	
Identification			Flags	Fragment Offset
Time to Live		Protocol	Header Checksum	
Source IP Address				
Destination IP Address				

Field	Meaning
Version	Version of the IP protocol. Most networks use version 4 today.
IHL	IP Header Length. Defines the length of the IP header, including optional fields.
DS Field	Differentiated Services Field. It is used for marking packets for the purpose of applying different quality-of-service (QoS) levels to different packets.
Packet Length	Identifies the entire length of the IP packet, including the data.
Identification	Used by the IP packet fragmentation process: all fragments of the original packet contain the same identifier.
Flags	3 bits used by the IP packet fragmentation process. First bit: unused, always 0. Second bit: DF (Don't Fragment) flag. 0: fragmentation is allowed, 1: not allowed. Third bit: MF (More Fragments) flag. Tells the receiver if more fragments are on the way. 0 means, no more fragments need to be sent or there are no other fragments.
Fragment offset	A number used to help hosts reassemble fragmented packets into the original larger packet. 13-bit field. <i>The offset value expresses the offset as a number of 8-byte units.</i>
TTL	Time to live. A value used to prevent routing loops . This field starts with 255 (if not manually defined) and decrements with each router hop. If the field is reduced to 0, the packet is dropped. Outside of a loop, this number of hops is not reached.
Protocol	A field that identifies the contents of the data portion of the IP packet. For example, protocol 6 implies that a TCP header is the first thing in the IP packet data field.
Header Checksum	A value used to store an FCS value, whose purpose is to determine if any bit errors occurred in the IP header.
Source IP address	The 32-bit IP address of the sender of the packet.
Destination IP address	The 32-bit IP address of the intended recipient of the packet.

Fragmentation

When a message is sent to the IP layer and it is too large to fit in one datagram (MTU too small), IP fragments the message into multiple datagrams, giving all datagrams the same identification number (Field "identification"). This number is used on the receiving end to reassemble the original message. Next-hop-routers do not reassemble even if the next link has a bigger MTU.

9.3. IPV4 NETWORK ADDRESSES

IPv4 addresses are expressed in 32 binary bits divided into 4 octets (or bytes)

11000000.10101000.00001010.00000000 or in DDN: **192.168.10.0**

An IPv4 address is composed of a **Network portion** and a **Host portion**. All devices on the same network must have the identical network portion. **The subnet mask** helps devices identify the network portion and host portion.

A subnet mask of /24 or 255.255.255.0 means that the first 24 bits are the network portion and the last 8 bits the host portion. (See more in next chapter)

Three addresses must be configured on a host: Unique IPv4 address, Subnet mask and Default gateway

The **first** and the **last** Address of a network are **reserved** for the **Network Address** and the **Broadcast address**. The **First Host Address** is therefore the Network Address + 1 and the **Last Host Address** the final Address – 1. **n-2** Addresses are usable for hosts.

9.3.1. Network classes

Name	Mask	Addresses	Description
Class A	/8	0.0.0.0/8 to 127.0.0.0/8	more than 16 million host addresses (<i>127 is a reserved address</i>)
Class B	/16	128.0.0.0/16 to 191.255.0.0/16	Up to 65'000 hosts
Class C	/24	192.0.0.0/24 to 223.255.255.0	Maximum of 254 hosts

9.3.2. Private Addresses

- Not routable
- Introduced in mid 1990 due to depletion of IPv4 addresses
- Must be translated to a public IPv4 to be routable
- Defined by RFC 1918

Private Address Blocks:

- 10.0.0.0/8 (10.0.0.0 to 10.255.255.255)
- 172.16.0.0/12 (172.16.0.0 to 172.31.255.255)
- 192.168.0.0/16 (192.168.0.0 to 192.168.255.255)

Loopback addresses (127.0.0.0/8 or 127.0.0.1)

Used for loopback addresses to the local host

Link-Local addresses (169.254.0.0/16 or 169.254.0.1)

Used for local communications within a private network, used by windows client to self-configure if no DHCP server is available.

9.4. SUBNETTING

Creating smaller networks inside one network. Useful for:

- Improving **security**
- Improving **performance**
- Enhanced **scalability**
- Enhanced **organization**

Networks are most easily subnetted at the octet boundary of /8, /16 and /24. Subnets are created by borrowing host bits and use them as network bits.

Subnet Mask (DDN)	32-bit Address	Prefix (CIDR)	# of subnets in a /24 net	# of addresses (-2 for usable)
255.0.0.0	11111111.00000000.00000000.00000000	/8		16'777'214
255.255.0.0	11111111.11111111.00000000.00000000	/16		65'534
255.255.128.0	11111111.11111111.10000000.00000000	/17		32'768
255.255.192.0	11111111.11111111.11000000.00000000	/18		16'384
255.255.224.0	11111111.11111111.11100000.00000000	/19		8192
255.255.240.0	11111111.11111111.11110000.00000000	/20		4096
255.255.248.0	11111111.11111111.11111000.00000000	/21		2048
255.255.252.0	11111111.11111111.11111100.00000000	/22		1024
255.255.254.0	11111111.11111111.11111110.00000000	/23		512
255.255.255.0	11111111.11111111.11111111.00000000	/24		256
255.255.255.128	11111111.11111111.11111111.10000000	/25	2	128
255.255.255.192	11111111.11111111.11111111.11000000	/26	4	64
255.255.255.224	11111111.11111111.11111111.11100000	/27	8	32
255.255.255.240	11111111.11111111.11111111.11110000	/28	16	16
255.255.255.248	11111111.11111111.11111111.11111000	/29	32	8
255.255.255.252	11111111.11111111.11111111.11111100	/30	64	4

Variable Length Subnet Masks (VLSM)

Subnets do not have to be equal sizes if their address ranges do not overlap.

Organizations which manage and maintain IPv4 addresses: ARIN, RIPE, AfriNIC, LACNIC

9.4.1. IP classful and IP classless addressing

Classless addressing rules define a two-part IP address structure: the **prefix and the host part**. This logic **ignores Class A, B and C rules**, and can be applied to the 32-bit IPv4 addresses from any address class. By ignoring Class A, B and C rules, classless addressing ignores any distinction as to the network part of an IPv4 address. -> More **efficient allocation** of IP addresses.

9.5. ROUTING

A host can send a packet to:

- **Local host:** This is a host on the same local network. The hosts share the same network address.
- **Remote host:** This is a host on a remote network. The hosts do not share the same network address.

Routing Information Base (RIB): Used to store routing information in a router.

Forwarding Information Base (FIB): Used to store next hop information in a router.

A Router's **routing table** contains:

- **Directly connected routes:** These routes come from the active router interfaces configured with IP addresses.
- **Remote routes:** these routes come from remote networks connected to other routers. They are either configured manually or learned through a dynamic routing protocol.
- **Default route:** This is where the packet is sent when a route does not exist in the routing table.
(`ip route 0.0.0.0 0.0.0.0 <next-hop>`)

9.5.1. The next-hop Address

When a packet arrives at a router destined for a remote network, it will send the packet to the **next hop address** corresponding to the destination network address in its routing table. If the router receives a packet for a network that **isn't in the routing table**, it will be **dropped**, if no **default gateway** is defined.

9.5.2. Learning about Networks

A router learns about remote networks in two ways:

- **Manually** entered into the routing table using static routes (those are not automatically updated and must be reconfigured when the topology changes)
- **Dynamically** learned using a routing protocol

9.5.3. Summarization

It is possible to summarize IP routes. For example:

- `ip route 172.20.0.0 255.255.0.0 10.1.1.1 (172.20.0.0/16)`
- `ip route 172.21.0.0 255.255.0.0 10.1.1.1 (172.21.0.0/16)`
- `ip route 172.22.0.0 255.255.0.0 10.1.1.1 (172.22.0.0/16)`
- `ip route 172.23.0.0 255.255.0.0 10.1.1.1 (172.23.0.0/16)`

can be summarized to:

`ip route 172.20.0.0 255.252.0.0 10.1.1.1 (172.20.0.0/14)`

9.6. NAT

Because there are not enough IPv4 Addresses, we need NAT, the translation of a private address to a public one. NAT can also be **nested**, we can map private addresses to other private addresses and so forth.

9.6.1. NAT Characteristics

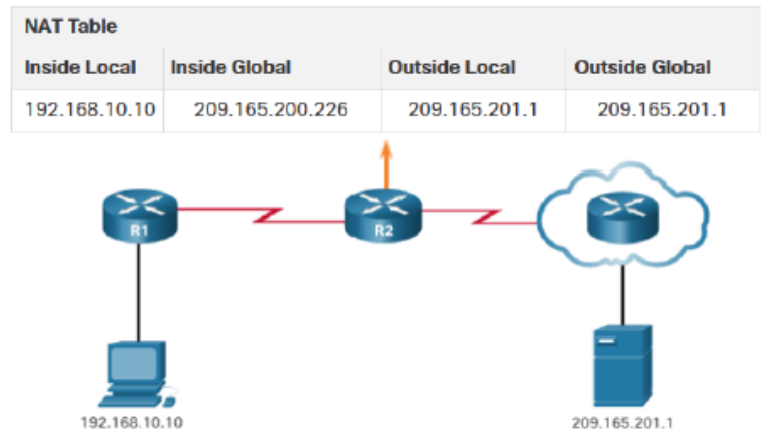
Private IPv4 addresses cannot be routed over the internet and are used within an organization or site to allow devices to communicate locally. To allow a device with a private IPv4 address to access devices and resources outside of the local network, the private address must first be **translated to a public address**.

The primary use of NAT is to **conserve public IPv4 addresses**. A NAT router typically operates at the border of a stub network. When a device inside the network wants to communicate with a device outside of its network, the packet is forwarded to the border router which performs the NAT process, translating the internal private address of the device to a public, outside, routable address.

NAT Terminology

Nat includes four types of addresses:

- **Inside local:** The address of the client, as it appears on the inside of the network
- **Inside global:** The address of the client, as it appears on the outside of the network
- **Outside local:** The address of the destination device, as it appears on the inside of the network
- **Outside global:** The address of the destination device, as it appears on the outside of the network



Local: before NAT, Global: after NAT

How NAT works

- The **client sends a packet** addressed to the web server.
- The router receives the packet and reads the source IPv4 address to determine if it needs **translation**.
- The router **adds mapping of the local to global address to the NAT table**.
- The router sends the packet with the translated source address towards the destination.
- The **web server** responds with a packet **addressed** to the **inside global address** of the client
- The router receives the packet and **translates** the inside global address with the help of its NAT table to the inside local address and **forwards the packet towards the client**.

9.6.2. Advantages and Disadvantages of NAT

Advantages: Conserves addresses, increases flexibility, provides internal consistency, hides IPv4 addresses of users

Disadvantages: Increases forwarding delays, end-to-end addressing & traceability is lost, complicates the use of tunneling protocols like IPsec, Services like UDP or TCP can be disrupted

9.6.3. Static NAT

Uses a **one-to-one mapping** of local and global addresses **configured by the network administrator** that remain **constant**. Useful for web servers or devices that must have a **consistent address** that is accessible from the internet, such as a company web server. Also useful for devices which must be accessible by authorized personnel when offsite, but not by the public on the internet.

Requires that **enough public addresses are available** to satisfy the total number of simultaneous sessions.

Configure Static NAT

- **Step 1:** Create a mapping between the inside local address and the inside global
- **Step 2:** The interfaces participating in the translation are configured as inside or outside relative to NAT.

9.6.4. Dynamic NAT

Uses a **pool of public addresses** and assigns them on a first-come, first-served basis. Dynamic NAT requires that **enough public addresses are available** to satisfy the total number of simultaneous user sessions.

If all addresses in the pool are in use, a device **must wait for an available address** before it can access the outside network. Translation entries time out after **24 hours** if not specified differently.

Configure Dynamic NAT

- **Step 1:** Define the pool of addresses
- **Step 2:** Configure a standard ACL to identify only those addresses that are to be translated
- **Step 3:** Bind the ACL to the pool
- **Step 4:** Identify which interfaces are inside
- **Step 5:** Identify which interfaces are outside

9.6.5. PAT (Port Address Translation)

Standard today. **Maps multiple private IPv4 addresses to a single public IPv4 address.** With PAT, when the NAT router receives a packet from the client, it uses the **source port number** (Layer 4) to uniquely identify the specific NAT translation. PAT ensures that devices use a different TCP port number for each session with a server on the internet.

Next Available Port: PAT **attempts to preserve** the original source port. If the original source port is already used, PAT assigns the **first available port number** starting from the beginning of the appropriate port group. When there are no more ports available and there is more than one external address in the address pool, PAT **moves to the next address** to try to allocate the original source port.

Packets without a Layer 4 segment: Some packets **do not contain a Layer 4 port number**, such as ICMPv4 messages. Each of these types of protocols is **handled differently** by PAT. *For example, ICMPv4 query messages, echo requests, and echo replies include a Query ID: ICMPv4 uses the Query ID to identify an echo request with its corresponding echo reply.*

9.6.6. NAT and PAT Comparison

NAT	PAT
Only modifies the IPv4 Addresses	Modifies both the IPv4 address and the port number
One-to-one mapping between Inside Local and Inside Global Addresses	One Inside Global address can be mapped to many inside local addresses
Uses only IPv4 addresses in translation process	Uses IPv4 addresses and TCP or UDP source port numbers in translation process
A unique Inside Global Address is required for each inside host accessing the outside network	A single unique inside Global Address can be shared by many inside hosts accessing the outside network

9.6.7. NAT64 (NAT for IPv6)

IPv6 was developed with the intention of making NAT for IPv4 unnecessary. However, IPv6 does include its own IPv6 private address space, unique local addresses (ULAs). **Provides translation between IPv4 and IPv6.** Used to provide access between IPv6-only and IPv4-only networks. It is not used as a form of private IPv6 to global IPv6 translation. Should not be used as a long-term strategy.

10. IPV6 / LAYER 3

There are **not enough Public IPv4 addresses**. There are only 2^{32} IPv4 addresses but 2^{128} IPv6 addresses. The IPv6 standard exists since 1998. The use of the IPv6 Internet is constantly growing but at a moderate pace. More and more companies have their public servers reachable natively on the IPv6 internet. Most companies still use IPv4 in their internal network. **IPv4 will not be replaced by IPv6. They will co-exist.**

10.1. THE IPV6 PROTOCOL

10.1.1. IPv4 and IPv6 Header Comparison

IPv4 Header

Version	IHL	Type of Service	Total Length	
Identification			Flags	Fragment Offset
Time to Live		Protocol	Header Checksum	
Source Address				
Destination Address				
Options				Padding

IPv6 Header

Version	Traffic Class	Flow Label	
Payload Length		Nex Header	Hop Limit
Source Address			
Destination Address			

Fields not kept in IPv6 / Fields name kept from IPv4 to IPv6 / Name and Position Changed in IPv6 / New Field in IPv6

Fields removed in IPv6 header

- **Header length:** Ipv6 has a fixed header length (40 bytes)
- **Fragmentation:** IPv6 does not do fragmentation. If a sending host wants to do fragmentation, it will do it through extension headers.
- **Checksum:** not needed because both media access and upper layer protocol (UDP and TCP) have the checksum. IP is best-effort, plus removing checksum helps expedite packet processing

Name and Position Changed in IPv6

- **Traffic class (instead of ToS):** 8-bit field, tags the packet with a traffic class that can be used in differentiated services. Same functionalities as in IPv4
- **Payload length (instead of Total Length):** Same functionalities, except it does *not include* the header
- **Hop Limit (instead of TTL):** Same functionality.
- **Next header (instead of Protocol):** The value in this field tells you what type of information follows, if there is an extension header (TCP, UDP, extension header.. see next Chapter).

New Field in IPv6

- **Flow Label (RFC3697):** 20-bit field to identify specific flows needing special QoS. Flow classifiers had been based on 5-tuple: SA/DA, protocol type and port numbers of transport. Flow label value of 0 is used when no special QoS is requested (common case today).

10.1.2. Extension Headers (RFC2460)

It is possible to add any number of extension headers. Some examples:

Order	Header Type	Next Header Code
1	Basic IPv6 Header	-
2	Hop-by-Hop Options	0
	ICMP	1
3	Destination Options (with Routing Options)	60
4	Routing Header	43
5	Fragment Header	44
6	Authentication Header	51
7	Encapsulation Security Payload Header	50
8	Destination Options	60
9	Mobility Header	135
	No next header	59
Upper Layer	TCP	6
Upper Layer	UDP	17
Upper Layer	ICMPv6	58

10.1.3. IPv4 vs. IPv6 Fragmentation

IPv4	IPv6
Hop-by-hop fragmentation if necessary and if DF bit is 0	Fragmentation performed by the end hosts using an extension header that gives the details of the fragment
IPv4 routers are able to fragment	IPv6 routers don't fragment
Minimum IPv4 MTU is 68 bytes	Minimum IPv6 MTU is 1280 bytes
The end-hosts are responsible for re-assembly	The end-hosts are responsible for re-assembly
The host can perform path MTU discovery	The host must first discover the MTU and then performing fragmentation as necessary (1280 bytes can be safely used)

Path MTU Discovery

Path MTU Discovery depends on an ICMPv6 Packet Too Big message (ICMPv6 type 2 code 0) from the system that discards a datagram because it is too large for the next link.

10.2. IPV6 ADDRESSING

8 x 16-bit (128bit) in hexadecimal format.

2031:0000:130F:0000:0000:09C0:876A:130b

4 hexadecimal digits = 16 binary digits: FFFF -> 1111 1111 1111 1111

Simplify IPv6 Addresses

- Consecutive zeros can be replaced with a double colon "::" (only once per address): **2031:0000:130F::09C0:876A:130b**
- Leading zeros can be omitted (as many times as needed): **2031:0:130F::9C0:876A:130b**

10.2.1. Ipv6 Address Family

- **Multicast:** Assigned (Well Known, Temp) and Solicited Node
- **Unicast:** Unique Local, Link Local, Global, Special, Embedded
- **Anycast** (for example DNS)

No Broadcast! Multicast is used instead.

10.2.2. Addressing Model

Interfaces have multiple addresses. Link Local, Unique Local and Global. Addresses have lifetime.

10.2.3. Subnets

In IPv6 it is best practice to not make subnets smaller than /64. With /64, subnets look like this:

- 2001:DB8:ACAD:1::/64
- 2001:DB8:ACAD:2::/64
- 2001:DB8:ACAD:3::/64

10.2.4. Global Unicast Address

Used to **uniquely identify a device on the internet**. Is used to route traffic to and from the device. Globally routable, default global range: **2000::/3**

10.2.5. Link Local Address

Used to **uniquely identify a device on a local network**. Only valid inside the local network. Default link-local subnet: **FE80::/10**. Is Automatically assigned by the Router as soon as IPv6 is enabled. Also used for Next-Hop calculation in Routing Protocols. Remaining 54 bits could be Zero or any manually configured value.

MAC **AA:0A:48:07:F9:91** becomes **fe80::a80a:48ff:fe07:f991/64**

- **fe80** Link-Local Identifier
- **a80a:48** First half of the MAC with the seventh bit of the first byte flipped
- **ff:fe** Statistical delimiter
- **07:f991** Second half of the mac

10.2.6. Unique Local Address

Same as Link Local, but globally unique. Not really used anymore.

10.2.7. Special Unicast IPv6 addresses

- **Localhost:** ::1
- **Unspecified address:** ::/128
- **Documentation Prefix:** 2001:db8::/32
- **Discard Prefix:** 0100::/64
- **Default Route:** ::/0
- **Link-Local:** FE80::/10

10.2.8. Multicast IPv6 address

IPv6 multicast address has a prefix FF00::/8 (1111 1111), the second octet defines the lifetime and scope of the multicast address.

8-bit	4-bit	4-bit	112-bit
1111 1111	Lifetime	Scope	Group-ID
	0: If Permanent	1: Node	
	1: If Temporary	2: Link (routers)	
		5: Site	
		8 Organization	
		E: Global	

IPv6 Address	IPv4 Address	Purpose
FF02::1	Subnet-Broadcast-Address	<i>All hosts</i> of the link / nodes on a segment
FF02::2	Not available	<i>All routers</i> of the link / on a segment. <i>If a node joins this multicast group, it means that the node is a router</i>
FF02::5, FF02::6	224.0.0.5, 224.0.0.6	OSPF neighbors
FF02::9	224.0.0.9	RIPv2 neighbors
FF02::1:2	No address	DHCP-Relay-Agents / DHCP server

Solicited-node multicast

Prefix FF02::1:FF + lower 24 bits from unicast IPv6 address.

For each unicast and anycast address configured, there is a *corresponding solicited-node multicast*.

Used for the *replacement of ARP* and for Duplicate Address Detection (DAD). This is to make sure that the client gets assigned an address which is not already in use.

When a client assigns itself a IPv6 address, it sends a request to the corresponding solicited-node multicast address. Everyone in the scope of the solicited-node multicast address gets this (if they have the same last 24 bits). If someone has the same address, it replies, and the client needs to select a different IPv6 address.

10.3. ICMPV6 AND NEIGHBOR DISCOVERY PROTOCOL (RFC 4861)

ICMPv4 and ICMPv6 roles:

- Host Confirmation
- Destination or Service Unreachable
- Time Exceeded
- Router Redirection
- Messaging between an IPv6 router and an IPv6 device:
- Router Solicitation Message (RS)
- Router Advertisement Message (RA)
- Messaging between IPv6 devices:
- Neighbor Solicitation Message (NS)
- Neighbor Advertisement Message (NA)
- Duplicate Address Detection (DAD)

10.3.1. Neighbor Discovery (RFC 4861)

Replaces IPv4 ARP (Layer-2 address resolution for IPv6)

Adds new enhanced functionality (Finding Routers, Redirect, Stateless addressing, etc.)

10.3.2. Router Advertisement (RA) and Router Solicitation (RS)

Routers send periodic Router Advertisements to the all-nodes multicast address. At boot time, nodes send Router Solicitations to receive promptly Router Advertisements.

10.3.3. Duplicate Address Detection (DAD)

DAD uses neighbor solicitation to verify the existence of an address to be configured.

10.3.4. Redirect

Redirect is used by a router to signal the reroute of a packet to a better route.

10.4. IPV6 CONFIGURATION

There are 5 configuration options for Global Unicast addresses

- *Static configuration*: The IPv6 address and the prefix length are both configured on the interface
- *EUI-64*: Configuration of the prefix and prefix length. The interface ID is created automatically based on the MAC address. The 48-bit MAC address is expanded to 64 bit by inserting FFFE into the middle 16 bits. The 7. bit from the start is inverted. Problematic because of privacy issues.
- *Stateless address autoconfiguration (SLAAC)*: The prefix and the prefix length is derived from the ND Router Advertisement. The interface ID is created automatically based on the MAC address (EUI-64)
- *SLAAC+ stateless DHCP (RFC 3736)*: DHCP is used to get additional information such as the DNS servers.
- *Stateful autoconfiguration (RFC 3315) or DHCPv6*: similar to DHCP for IPv4

There are 2 configuration options for Link-Local addresses:

- *Dynamically*, using EUI-46 or a random generated Interface ID
- *Statically*, entering the link-local address manually.

IPv6 interfaces are expected to have **multiple** IP addresses: one or many global unicast addresses and one link-local address.

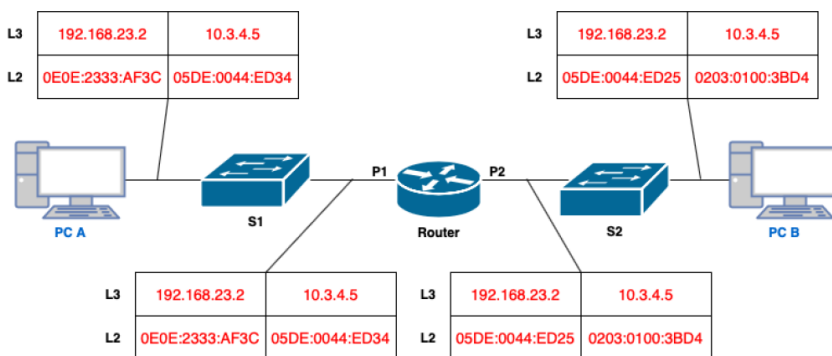
10.5. ROUTING

PC A is sending a packet to PC B. What are the Layer 2 and Layer 3 SA and DA of the packet at each step?

Device	MAC Address	IP Address
PC A	0E0E:2333:AF3C	192.168.23.2
PC B	0203:0100:3BD4	10.3.4.5
Switch S1	0a02:2222:5FFC	192.168.23.253
Switch S2	0a02:2222:C2B4	10.3.4.251
Router P1	05DE:0044:ED34	192.168.23.1
Router P2	05DE:0044:ED25	10.3.4.1

Solution to Ethernet 5.5.5

Picture with addresses



11. LINK-STATE ROUTING (DYNAMIC ROUTING) / LAYER 3

Dynamic Routing relies on a routing protocol to share knowledge among routers. A routing protocol define how to send updates, what knowledge is contained in these updates, when to send this knowledge and how to locate recipients of the updates. **A router passes routing information to its neighbors.**

Routers have a **global view** of network topology. Each router has **exact knowledge about all other routers**, all links and their costs of a network stored in the topology database.

The routing table in Link-State routing is only sent if it is triggered, if there is a problem. Is not sent periodic.

How do all nodes know about the network topology and find the best path? See next:

11.1. HELLO PROTOCOL

First, neighboring routers **have to establish a relationship** called adjacency. They **need to agree on a set of protocol specific parameters** such as timers, capabilities etc.

Once the routers agree on adjacent neighborship, they **exchange and verify database information** until their DBs are identical. They continue to exchange "keepalives" (e.g. every 10 sec.). Routers are considered dead if keepalives are not responded for ca 30-40 sec.

The OSPF packets are sent as IP multicast packets on LAN, point-to-point and point-to-multipoint links.

11.2. CONTROLLED FLOODING OF LINK STATE ADVERTISEMENTS (LSA)

In the **case of a topology change**, an LSA is generated and **sent to every adjacent neighbor**. Each received LSA is copied to every other link except on the link that the LSA was received on. **When all routers have received all LSAs, the flooding stops.** All LSAs use sequence numbers and the latest LSA is kept. There is also an Age field in all LSAs that is incremented as the LSA is re-transmitted across the network.

An LSA includes:

- **Router Link Information:** advertises a router adjacent neighbors with a triple of Router ID, Neighbor ID and Cost where the cost is the cost of the link to the neighbor

- **Stub Network Information:** advertises a routers directly connected stub networks (networks with no neighbors) with a triple of Router ID, Network ID and Cost.

Each router creates its own topology database by storing the LSA received from all other routers as a series of records.

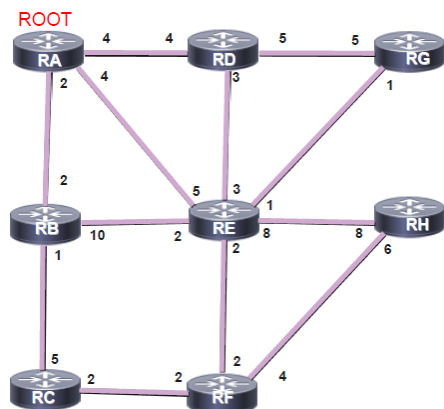
11.3. SHORTEST PATH FIRST (SPF) ALGORITHM (DIJKSTRA)

Constructs a **tree** of minimum total lengths between the n nodes (The tree graph is a graph with only one path between every two nodes). Every time there is a **change in the network topology**, Routers must **rerun SPF**. An inter-area topology change does not trigger the SPF recalculation.

Create a table of three columns:

Candidate LSA	Cost to root	Tree
---------------	--------------	------

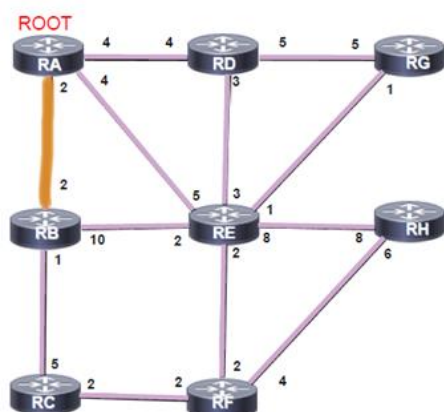
1. A router initializes the **Tree Database** by adding itself as the root.
2. All triples in the link state database describing links to the roots routers neighbors are added to the **Candidate Database**
3. The cost from the root to each link in the **Candidate Database** is calculated. The link in the **Candidate Database** with the lowest cost is moved to the **Tree Database**. If two or more links are an equally low cost from the root, choose one.
4. The Neighbor ID of the link just added to the **Tree Database** is examined. Except for any triples whose Neighbor ID is already in the **Tree Database**, triples in the link state database describing that routers neighbors are added to the **Candidate Database**.
5. If entries remain in the **Candidate Database**, return to step 3. If the **Candidate Database** is empty, then terminate the algorithm. At termination, a single Neighbor ID entry in the **Tree Database** should represent every router, and the shortest path tree is complete.



Candidate LSA	Cost to root	Tree
RA, RD, 4	4	RA, RB (2)
RA, RB, 2	2	
RA, RE, 4	4	

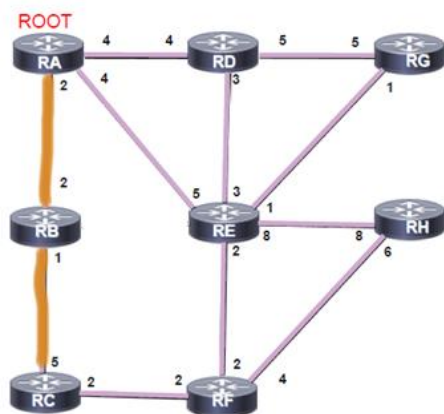
Vom root gehen 3 Pfade weg, welcher davon ist der Kürzeste? -> RA RB. Wird in Tree Spalte geschrieben.

Die anderen verbleiben in der Candidate Spalte. Nun werden die Pfade von RB weg angesehen.



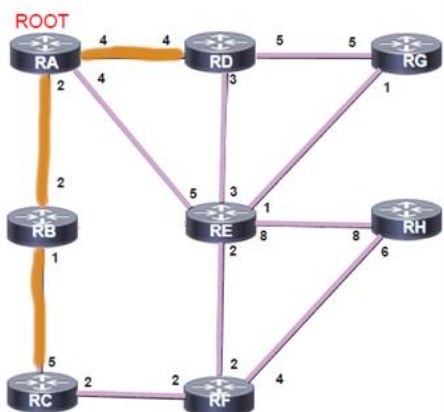
Candidate LSA	Cost to root	Tree
RA, RD, 4	4	RA, RB (2)
RA, RE, 4	4	RB, RC (3)
RB, RE, 10	12	
RB, RC, 1	3	

Welches ist der kürzeste Pfad weg von RB? -> RB RC. Wird in Tree Spalte geschrieben. RB RE kann aus der Tabelle entfernt werden, weil bereits ein kürzerer Weg zu RE in der Tabelle steht (RA RE). Als nächstes sehen wir uns die Pfade von RC weg an.



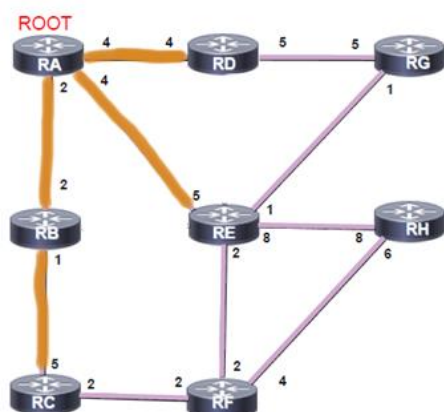
Candidate LSA	Cost to root	Tree
RA, RD, 4	4	RA, RB (2)
RA, RE, 4	4	RB, RC (3)
RC, RF, 2	5	RA, RD (4)

Da wir zu RF hin noch nicht alle Pfade haben, können wir nichts in die Tree Spalte schreiben. Nun müssen wir einen anderen Router ansehen. Da RA RD und RA RE die gleiche Cost haben, nehmen wir einfach einen davon (bspw. mit tiefster ID). Wir schauen uns also als nächstes RD an.



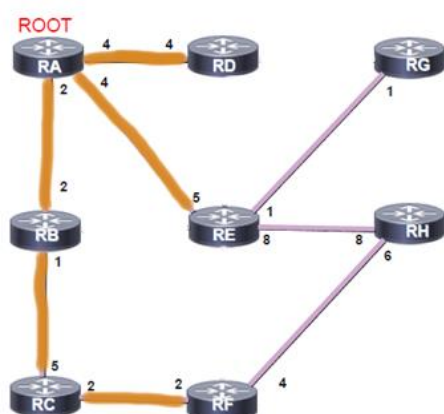
Candidate LSA	Cost to root	Tree
RA, RE, 4	4	RA, RB (2)
RC, RF, 2	5	RB, RC (3)
RD, RE, 3	7	RA, RD (4)
RD, RG, 5	9	RA, RE (4)

Nun haben wir alle Pfade zu RE in der Liste. Welches ist der kürzeste Pfad? -> RA RE. Wird in Tree Spalte geschrieben. RD RE kann aus der Tabelle entfernt werden, weil bereits ein kürzerer Weg zu RE in der Tabelle steht (RA RE). Als nächstes sehen wir uns die Pfade von RE weg an.



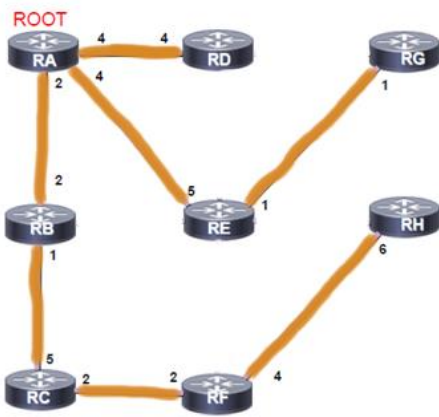
Candidate LSA	Cost to root	Tree
RC, RF, 2	5	RA, RB (2)
RD, RG, 5	9	RB, RC (3)
RE, RG, 1	5	RA, RD (4)
RE, RH, 8	12	RA, RE (4)
RE, RF, 2	6	RC, RF (5)

Nun haben wir alle Pfade von RE weg in der Liste. Es kann erkannt werden, dass RC RF der kürzeste Pfad zu RF ist. Dieser kann in die Tree Spalte geschrieben werden. Alle anderen Pfade zu RF können entfernt werden. Auch der Pfad RD RG kann entfernt werden, weil bereits ein kürzerer Pfad in der Liste zu sehen ist. Als nächstes sehen wir uns die Pfade von PF her an.



Candidate LSA	Cost to root	Tree
RE, RG, 1	5	RA, RB (2)
RE, RH, 8	12	RB, RC (3)
RF, RH, 4	9	RA, RD (4)
		RA, RE (4)
		RC, RF (5)
		RE, RG (5)
		RF, RH (9)

Es ist nun erkennbar, dass zu RG nur noch ein Pfad führt, nämlich RE RG. Dieser kann in die Tree Spalte übernommen werden. Und auch zu RH ist der kürzeste Weg sichtbar, RF RH. Auch dieser wird übernommen.



Candidate LSA	Cost to root	Tree
		RA, RB (2)
		RB, RC (3)
		RA, RD (4)
		RA, RE (4)
		RC, RF (5)
		RE, RG (5)
		RF, RH (9)

Nun steht nichts mehr in der Candidate Spalte. Der Algorithmus kann beendet werden.

11.4. STEPS OF A TRACEROUTE

- An ICMP packet with TTL=1 is sent to the destination IP address.
- The TTL will expire on R1 and R1 is sending an ICMP TTL expired back to PC1 (ICMP type 11 code 0)
- An ICMP packet with TTL=2 is sent to the destination IP address.
- The destination IP address is reached. R2 is sending an ICMP type 0 back to PC1.



11.5. OPEN SHORTEST PATH FIRST V2 (OSPFV2) FOR IPV4

The OSPF standard was developed by the Internet Engineering Task Force (IETF) *to overcome the limitations of distance vector routing protocols*. One of the main reasons why OSPF is largely deployed in today's enterprise networks is the fact that it is an open standard. *It offers a large level of scalability and fast convergence*. Despite its relatively simple configuration in small and medium-sized networks, OSPF implementation and troubleshooting in large-scale networks can be challenging.

11.5.1. OSPF operation overview

11.5.1.1. Establish neighbor adjacencies

OSPF-enabled routers *must form adjacencies* with their neighbor before they can share information with that neighbor. An OSPF enabled router *sends Hello packets* out all OSPF-enabled interfaces to determine whether neighbors are present on those links. If a neighbor is present, the OSPF enabled router *attempts to establish a neighbor adjacency* with that neighbor. For OSPF to form a neighborhood the following parameters must match:

- Area
- Network Time
- Subnet (unless P2P)
- Timers
- MTU
- Stub Flag
- Authentication

11.5.1.2. Exchange link-state advertisements (LSAs)

After adjacencies are established, *routers exchange link-state advertisements*.

– See Chapter "Controlled Flooding of Link State Advertisement"

11.5.1.3. Build the Link State Database

After the LSAs are received, OSPF-enabled routers *build the topology table* (LSDB) based on the received LSAs. The database eventually holds all the information about the topology of the network. It is important that *all routers in the area have the same information* in their LSDBs.

11.5.1.4. Execute the SPF algorithm

Routers then *execute the SPF algorithm*. The SPF algorithm creates the SPF tree.

11.5.1.5. Build the routing table

From the SPF tree, the *best paths are inserted into the routing table*. Routing decisions are made based on the entries in the routing table.

11.5.2. OSPF Features

Independent transport

OSPF works on top of IP and uses protocol number 89. It does *not rely on the transport layer protocols* TCP or UDP.

Efficient use of updates

When an OSPF router first discovers a *new neighbor*, it sends a *full update* with all known link-state information. All routers within an OSPF area must have identical and synchronized link-state information in their OSPF link-state databases. When an OSPF network is in a converged state and a new link comes up or a link becomes unavailable, an OSPF router sends only a *partial update* to *all its neighbors*. This update will then be *flooded* to all OSPF routers within an area.

Metric

OSPF uses a metric that is based on the cumulative costs of all outgoing interfaces from source to destination. The interface cost is *inversely proportional* to the interface bandwidth and can be also set up explicitly (Cost is a 16-bit integer). Default on Cisco Router: $10^8/BW$ where BW is the interface bandwidth in bit/s.

BW (b/s)	Cost
128K (128'000)	781
10M (10^7)	10
100M (10^8)	1

Designated routers and Backup Designated routers

OSPF uses *multicast and unicast*, rather than broadcast, for sending messages. The use of reserved multicast addresses *reduces the impact* on non-OSPF-speaking devices. The IPv4 multicast addresses used for OSPF are *224.0.0.5* to send information to all OSPF routers and *224.0.0.6* to send information to DR/BDR routers.

VLSM Support

OSPF is a *classless routing protocol*. It supports variable-length subnet masking (VLSM) and discontinuous networks. It *carries subnet mask information* in the routing updates.

Authentication

OSPF supports clear-text, MD5 and SHA authentication.

Route Summarization (creating areas)

Helps *solve two major problems: large routing tables* and *frequent LSA flooding*. Every time that a route disappears in one area, routers in other areas also get involved in shortest-path calculation. To reduce the size of the area database, you can configure summarization on an area boundary or AS boundary.

What hinders Routers to create an neighborhood?

MTU mismatch, Network mask mismatch, Authentication mismatch,...

OSPF neighbor states

- | | | |
|-----------------|------------------|-----------------|
| - Down state | - 2-Way state | - Loading state |
| - Attempt state | - Exstart state | - Full state |
| - Init state | - Exchange state | |

11.5.3. Hierarchical Structure of OSPF

Backbone Area or Area 0

Two principal requirements for the backbone area are *that it must connect to all other nonbackbone areas directly*. Normally, end users are not found within a backbone area.

Nonbackbone area

The primary function of this area is to *connect end users and resources*. Nonbackbone areas are usually set up according to functional or geographic groupings. Traffic between different nonbackbone areas must always pass through the backbone area.

Router roles

- **Area Border Router (ABR):** Connects different areas of an OSPF network
- **Autonomous System Border Router (ASBR):** Connects an OSPF network to another routing domain
- **Internal router:** Is part of an OSPF network, but neither an ABR nor an ASBR
- **Backbone router:** Is part of the backbone area of an OSPF network

Every area without an ASBR need to be connected to the area with the ASBR.

11.6. OSPFV3

OSPFv3 is the IPv6-capable version of the OSPF routing protocol. It is a rewrite of the OSPF protocol to support IPv6, although the foundation remains the same as in IPv4 and OSPFv2. It is not compatible with OSPFv2.

12. DNS/DHCP / LAYER 7

12.1. DOMAIN NAME SYSTEM (DNS)

DNS is a critical application that *permits hosts to query a database to get an IP address* by querying for a name. *DNS servers maintain a mapping between an IP address and the corresponding name.* DNS messages can be queries, replies or zone transfers.

DNS is run by Internet Service Providers (ISPs), organizations, and Internet authorities throughout the world. Every host runs a DNS client. It is a *Layer 7 (Application Layer) Protocol*.

The DNS server infrastructure consists of three levels: Root DNS Servers, Top-Level domain (TLD) servers and Authoritative servers.

The DNS services on the internet are *reachable using anycast IP addresses*. An anycast IP address is the same IP address that is used several times in the network. When a packet is sent to an anycast address, it is routed to the nearest server that has this address. The nearest server is found according to the measure of cost of the routing protocol. *Anycast provides high availability and load balancing* for stateless services such as DNS.

12.1.1. Process to Query the DNS Hierarchy

- **Step 1:** A host wants to connect to a website. The host sends a DNS query to its local DNS server.
- **Step 2:** If this local DNS server cannot obtain the name from its cache, it sends the query to the root name DNS server.
- **Step 3:** The root name DNS server sends to the local DNS server the IP addresses of the Top-Level Domain DNS server that handles that domain.
- **Step 4:** The local DNS server contacts the TLD DNS server
- **Step 5:** The TLD server sends to the local DNS server the IP address for the Authoritative DNS server that handles the domain for the host of the website.
- **Step 6:** The local DNS server sends the query to the authoritative DNS server
- **Step 7:** The local DNS gets the DNS response from the authoritative DNS server and delivers a DNS response to the host that initiated the query.

12.2. ROOT NAME SERVERS

13 Organizations manage the Root Name Servers, located in multiple sites worldwide. A Root Name Server *contains a Root zone database*. This DB represents the delegation details of top-level domains.

12.3. TOP LEVEL DOMAIN (TLD) SERVERS

For example .com DNS server or .ch DNS server. The Whois service is provided by the Top-level Domain DNS server.

12.4. AUTHORITATIVE DNS SERVER

For Example, amazon.com or swisscom.ch. Every organization, with hosts connected to the Internet, has at least one authoritative DNS server that provides authoritative hostname-to-IP address mappings for their organization, such as mail servers and web servers. These authoritative DNS servers, also known as *master servers*, contain the *original* set of data. A *secondary or slave name server* contains *data copies* that are normally obtained from direct synchronization with the master server. It is recommended that *three servers* should be provided for organizations

operating in the iterative mode.

In most cases, an authoritative DNS server is a name server, but not all name servers are authoritative DNS servers.

12.5. DNS QUERIES

12.5.1. Recursive Queries

A recursive query is a request for the DNS server *to resolve a domain name* to an IP address and return the result to the client. The authoritative DNS server, who receives the query *will do all the job* of fetching the answer. The DNS server queries other DNS servers in the internet on your behalf. The DNS server can be configured as a recursive name server. This can also be disabled.

12.5.2. Iterative Queries

An iterative query is a request for the DNS server *to provide the best information* it has available for a particular domain name. The authoritative DNS server *will not go and fetch the complete answer* for your query but will give back a referral to other DNS servers which might have the answer. It will give the answer if it has it in its records. All DNS servers must support iterative query. *All root servers and TLD servers are always iterative servers.*

12.5.3. Caching

DNS clients and DNS servers both use *caching to speed up the domain name lookup process* and to reduce traffic to the root servers. In an iterative or recursive query, a total of 8 messages will be sent. When a DNS query is resolved and the IP address of that domain is obtained, the DNS server caches the information from the reply.

For recursive queries: A local DNS server stores or caches an IP address of the TLD server to not ask the root DNS server for the IP address of the TLD server too often. Each time when the local DNS server receives a DNS replay from any DNS server, it caches the information.

For iterative queries: In the host, a cache preserves the mapping for a certain length of time.

Caching process *ensures that the search is conducted at the lowest level in the hierarchy*. Root and TLD name servers are *not often visited*. The TTL for the .com gTLD is two days. If a DNS server is going to provide caching services, then it must provide for recursive queries. Authoritative servers that provide Recursion Access Control (RAC) maintain control over hosts, which are permitted to use DNS recursive lookups, to reduce the computation and communication load.

12.5.4. DNS transport layer

DNS uses the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) port number 53 for lookups and transfers. DNS uses UDP for lookups. For zone transfer from primary authoritative to secondary authoritative server or if the response data size in a lookup exceeds 512 bytes, TCP is used.

12.6. DNS RESOURCE RECORDS (RR)

The Resource Records contain the information requested by DNS queries. This data is stored in a universal format. The format employed for these resource records:

(name, [pref.], value, type, [TTL], RDATA)

TTL is the lifetime of the cached RR and a 32-bit unsigned integer. The value zero indicates that the data should not be cached. TTL and pref are optional.

Type	Name	Value	Description
A	Hostname	IP address	hostname-to-IP address mapping
NS	Domain	Hostname of the authoritative name server for this domain	Used as a routing function for queries
CNAME	Alias name	Canonical name	Provides the canonical name when requested
MX	Domain Name	Name of the Mail server associated with this domain	A preference value is designated for each mail server if there are multiple MX RRs in a domain. When there are multiple MX RRs available, the mail server with the smallest Preference value is used.

12.6.1. Zone file

A zone file is a text file that describes a DNS zone and that is stored in the authoritative DNS server.

- **SOA (Start of Authority) record:** There is only one SOA record allowed in a zone file. The hostmaster.example.com represents the email address of hostmaster@example.com
- **Serial Number:** Is an unsigned 32-bit value. This value must increment when any resource record in the zone file is updated.
- **\$ORIGIN:** The base name to be used for name substitution.

12.6.2. Reverse DNS lookup

The reverse DNS lookup (rDNS) is a process used to **determine the hostname associated with some specific IP address**. Original use was with traceroute and ping (troubleshooting). It is also used in an email anti-spam technique. A Forward Confirmed Reverse DNS (FCrDNS) verification can generate a type of authentication that determines if a valid relationship exists between the IP address and the domain name.

12.6.3. DNS Protocol and Message Header Format

The messages are either query or reply and both have the same message format. The header consumes 12 bytes.

- **ID:** A query and its replay share the same ID.
- **Flags:** indicate whether this is a query or a reply, recursion is desired by a client query or available to a client, and the reply is from an authoritative server.
- **Questions:** The name and the type being queried
- **Answers:** provide the resource records for the queried name
- **Authority:** provides the records for another authoritative name server. A non-recursive reply contains no answer and delegates to another DNS server.
- **Additional information:** contains suggestions to ask another DNS server

32 bits, yellow is header

ID	Flags
Number of questions	Number of RRs in answer section
Number of RRs in authority records section	Number of RRs in additional section
Question section	
Answer section	
Authority records section	
Additional section	

12.7. DHCPV4

The host begins with no IPv4 settings: no IPv4 address, no mask, no default router, and no DNS server IP addresses. The host is a DHCP client.

The DHCP client protocol takes two roles: Discover a DHCP server and request to lease an IPv4 address.

The DHCPv4 server **chooses an address** from a configured range of addresses called a pool and **assigns it to the host client** for a set period. After it is leased, the client renews the lease expiration through another DHCPREQUEST. If the client is powered down or taken off the network, the address is returned to the pool for reuse.

- **Discover:** Sent by the DHCP client to find a willing DHCP server
- **Offer:** Sent by a DHCP server to offer to lease to that client a specific IP address
- **Request:** Sent by the DHCP client to ask the server to lease the IPv4 address listed in the Offer message.
- **Acknowledgment:** Sent by the DHCP server to assign the address, and to list the mask, default router and DNS server IP addresses

12.7.1. DHCPv4 Configuration Options

A router can be configured as a **DHCP server, DHCP relay agent or a DHCP client**. All these options can be configured **at the same time on the same device**. A router might be a DHCP server for a directly connected LAN, forward DHCP server request to another DHCP server for other LANs, a DHCP client and have one or more of its interfaces configured to request DHCP addresses from a remote DHCP server.

13. TCP AND UDP / LAYER 4

TCP and UDP are responsible for *establishing a temporary communication session* between two applications and delivering data between them. Link between the application layer (FTP, DNS, etc) and the lower layers that are responsible for network transmission (IP).

Characteristic / Description	UDP	TCP
General description	<i>Simple, high-speed, low-functionality</i> wrapper that interfaces applications to the network layer and does little else	<i>Full-featured protocol</i> that allows applications to send data <i>reliably</i> without worrying about network layer issues
Protocol connection setup	<i>Connectionless</i> , data is sent without setup	<i>Connection-oriented</i> , connection must be established prior to transmission
Data Interface to Application	Message-based, the application sends data in discrete <i>packages</i>	<i>Stream-based</i> , the application sends data with no structure
Reliability and acknowledgments	<i>Unreliable</i> , best-effort delivery without ACKs	<i>Reliable</i> delivery of messages, all data is acknowledged
Retransmissions	<i>Not performed</i> , application must detect lost data and retransmit if needed	Delivery of all data is managed, and lost data is <i>retransmitted automatically</i>
Features provided to manage flow of data	None	<i>Flow control</i> using sliding windows, <i>window size adjustment</i> heuristics, <i>congestion-avoidance</i> algorithms
Overhead	Very low	Low, but higher than UDP
Transmission speed	Very high	High, but not as high as UDP
Data quantity suitability	<i>Small to moderate</i> amounts of data (up to a few hundred bytes)	<i>Small to very large</i> amount of data (up to a few gigabytes)
Types of Applications that use the protocol	Applications where <i>data delivery speed matters more than completeness</i> , where small amounts of data are sent, or where multicast/broadcast are used	Most protocols and applications sending <i>data that must be received reliably</i> , including most file and message transfer protocols
Well-known applications and protocols	<i>Multimedia applications</i> , DNS, BOOTP, DHCP, TFTP, SNMP, RIP, NFS (early versions)	FTP, Telnet, SMTP, DNS, HTTP, POP, NNTP, IMAP, BGP, IRC, NFS (later versions)

13.1. TCP AND UDP ADDRESSING: PORTS AND SOCKETS

We all run many different applications simultaneously. TCP and UDP manage multiple conversations by using unique identifiers called port numbers and sockets.

13.1.1. Multiplexing and Demultiplexing

Multiplexing is the process of *combining multiple communication streams* into a single stream for transmission over a shared communication channel. Each communication stream is assigned a unique port number, which is used to identify the stream when it is transmitted over the shared communication channel.

Demultiplexing is the process of *separating a single stream* into multiple communication streams by using the identification via port numbers.

13.1.2. Ports

In UDP and TCP messages there are two addressing fields: *a source port and a destination port*. TCP and UDP port numbers *are 16 bits long*. Valid port numbers can take on values from 0 to 65'535. They identify the originating process on the source machine and the destination process on the destination machine. Both UDP and TCP use the same range of port numbers but they are independent.

Port number 77 in UDP might refer to a completely different process than port number 77 in TCP.

Port use: Sending Datagrams and Receiving Datagrams

Port Number ranges

- **Well-Known (Privileged): 0 to 1023** IANA manages these port numbers and reserves them for only the most universal TCP/IP applications.
- **Registered (User): 1024 to 49'151** There are many applications that need to use TCP/IP but are not specified in RFCs or are not as universally used as other applications. Anyone who creates a viable TCP/IP server application can request to reserve one of these port numbers, and if the request is approved, the IANA will register that port number and assign it to the application.
- **Private/Dynamic: 49'152 to 65'535** IANA neither reserves nor maintains these ports.

Well-known and registered port numbers are used for server processes. Ephemeral port numbers are used for client processes.

Client side: Ephemeral Port Number Assignment (flüchtig / kurzlebig)

Each time a client process initiates a UDP or TCP communication (with a server process), the TCP/IP software assigns it a **temporary or ephemeral port number** to use for that conversation. The client supplies the port number as the source port in the request, and then the server uses the source port as the destination port to send the reply. **Each client process** on a client needs to use a **unique ephemeral port number**, so the TCP and UDP layers must keep track of which ones are in use. *The TCP/IP software generally assigns these port numbers in a pseudo-random manner because the exact number that the software uses is not important as long as each process has a different number. (Pseudo-random to avoid immediate reallocation to another process)*

Identification of an application process

The **overall identification** of an application process uses the **combination of the IP address** of the host it runs on (or the IP address of the network interface) **and the port number** that has been assigned to it. This combined address is called a **socket**. Sockets are specified using the notation **<IP Address>:<Port Number>**. Example: 41.199.222.3:443
Each connection is uniquely identified using the combination of the client socket and server socket, which in turn contains **four elements: the client IP address and port, and the server IP address and port**.

Example: 41.199.222.3:80, 177.41.72.6:3022

13.2. UDP (USER DATAGRAM PROTOCOL)

UDP is simple and fast. The basic steps for transmission using UDP are as follows:

- **Higher-Layer Data Transfer:** An application sends a message to the UDP software
- **UDP Message Encapsulation:** The higher-layer message is encapsulated into the Data field of a UDP message. The headers of the UDP message are filled in, including the Source Port field of the application that sent the data to UDP and the Destination port field of the intended recipient. The checksum value may also be calculated.
- **Transfer Message to IP:** The UDP message is passed to IP for transmission

When the destination device receives the message, this procedure is reversed.

13.2.1. UDP Message Format

- **Source Port:** 16-bit port number of the process that originated the UDP message
- **Destination Port:** 16-bit port number of the process that is the destination of the message
- **Length:** The length of the entire UDP datagram, including both header and Data fields
- **Checksum:** Optional checksum computed over the entire UDP datagram plus a special pseudo header of fields

Bit (0)	Bit (15)	Bit (16)	Bit(31)	
Source Port (16)		Destination Port (16)		8 Bytes
Length (16)		Checksum (16)		
Application Layer Data (Size varies)				

13.2.2. Common Applications and Server Port Assignments

Applications that use UDP have the following features: *Performance is more important* than completeness (multimedia), *Exchanges of data are short*, Application uses *multicast* or *broadcast* (TCP is only for unicast)

Port #	Protocol	Comments
53	DNS	Uses a simple request/reply messaging system for most exchanges (but also uses TCP for longer ones)
67 and 68	Bootstrap Protocol (BOOTP) and Dynamic Host Configuration Protocol (DHCP)	Host configuration protocols that consist of short request and reply exchanges
69	Trivial File Transfer Protocol (TFTP)	TFTP is designed for the quick and easy transfer of small files. To avoid file corruption, TFTP application includes acknowledgments
161 and 162	Simple Network Management Protocol (SNMP)	An administrative protocol that uses relatively short messages
443	HTTPS	
520 and 521	Routing Information Protocol (RIP-1, RIP-2, RIPng)	RIP uses a simple request/reply messaging system and requires multicasts/broadcast

13.3. TCP (TRANSMISSION CONTROL PROTOCOL)

TCP Characteristics:

- *Connection-Oriented*: A process of negotiation occurs ensuring that both devices agree on how they will exchange data
- *Bidirectional*: Once a connection is established, TCP devices send data bidirectionally regardless of which one initiated the connection
- *Multiple connections*: The pair of sockets used by the two devices in the connection identifies the endpoints of the TCP connection. TCP can handle multiple connections independently without conflicts
- *Reliable/Acknowledged*: Communication using TCP is said to be reliable because TCP keeps track of data that has been sent and received to ensure that all the data gets to its destination
- *Managed Data Flow*: TCP does more than just package data and send it as fast as possible. A TCP connection is managed to ensure that data flows evenly and smoothly and dealing with possible congestion

IP is a message-oriented protocol. UDP is a stream-oriented protocol. TCP must take a *block of data* from the application and *divide it into discrete messages* for IP. These messages are called *TCP segments*. IP places them into IP datagrams and transmits them to the destination device. The recipient unpackages the segments and passes them to TCP, *which converts them back to a block* to send them to the application.

13.3.1. TCP Sliding Window Acknowledgment System

TCP provides basic reliability *using positive acknowledgment with retransmission* (PAR).

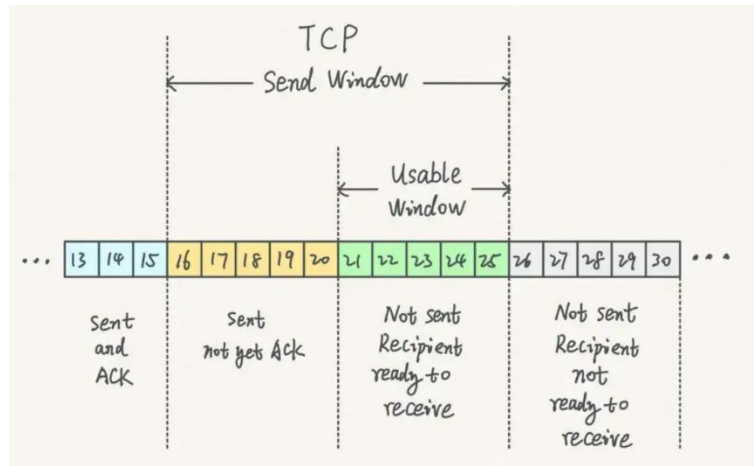
After every packet sent the sender waits for the ACK Message by the receiver. If no ACK is received in a specified amount of time, the packet is sent again, and the timer is reset.

PAR can be improved by using a message ID field in the message header or contain a field in the ACK message which specifies the maximum number of unacknowledged messages a device would be allowed to have in transit at one time.

TCP does send bytes individually but divides them into *segments*. All the bytes in a segment are sent together and received together, and thus acknowledged together. The TCP sliding window system is a variation on the enhanced PAR system with changes made to support TCPs stream orientation. Instead of a message ID field, the data is acknowledged using the sequence number of the last byte of data in the segment +1.

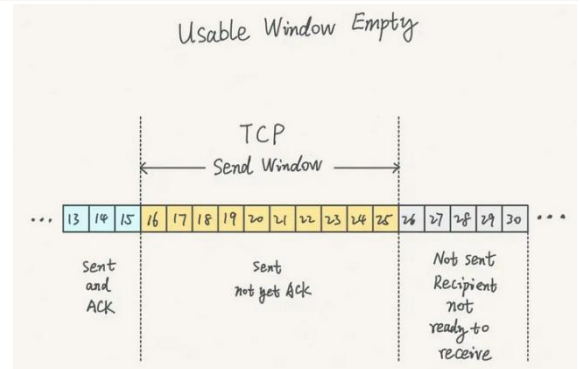
The sending device keeps track of the status of the byte stream that it needs to transmit by dividing the byte streams that the device has in its buffer into four conceptual categories:

- **Category 1:** Bytes sent and acknowledged
- **Category 2:** Bytes sent but not yet acknowledged
- **Category 3:** Bytes not yet sent for which recipient is ready
- **Category 4:** Bytes not yet sent for which recipient is not ready



The **send window** is the number of bytes that the recipient is allowing the transmitter to have unacknowledged at one time. The **usable window** is the amount of the send window that the sender is still allowed to send at any point in time (same as category 3)

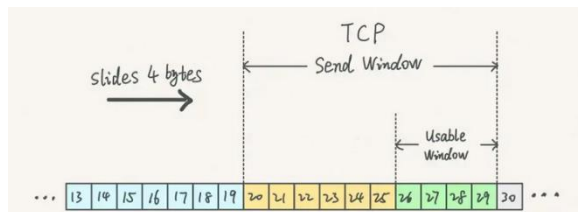
If the bytes in category 2, for example bytes 32 to 45 were transmitted in four different segments (32-34, 35-36, 37-41, 42-45) and all but the third segments arrived, the receiver will send back an ACK only for bytes 32-36. It will hold the bytes 42 to 45 but won't ACK them, because this would imply that the receiver has received bytes 37-41.



Dealing with missing acknowledgments

Until segment 3 in the example above shows up, the receiving device will not send an ACK for the following segments. The sending device will be able to send the new bytes that were added to category 3 after the last ACK (52-56). After that, the sending device will stop, and the **window will be stuck** on bytes 37-41. Eventually the TCP device will **resend the lost segments to unstuck the window**.

After the device has sent all the bytes that it is allowed to transmit, but before the ACK for the bytes in category 2 is received, the usable window is empty, which means the sender cannot send any new Bytes (Window is stuck). As soon as the sender receives an ACK, the window slides to the right. The receiver is ready to receive new bytes.



13.3.2. TCP basic operations

There are three types of messages that control transitions between states (TCP header flags set to indicate that a message is serving that function):

- **SYN (Synchronize message):** Initiates and establishes a connection. One of its functions is to synchronize sequence numbers between devices
- **FIN (Finish message):** TCP segment with the FIN bit set. It indicates that a device wants to terminate the connection
- **ACK (Acknowledgment message):** Indicates the receipt of a message such as a SYN or a FIN

Three-Way Handshake

The handshake is about exchanging information to establish a connection. The information falls into two categories: Sync initial sequence numbers (ISN) and exchange parameters.

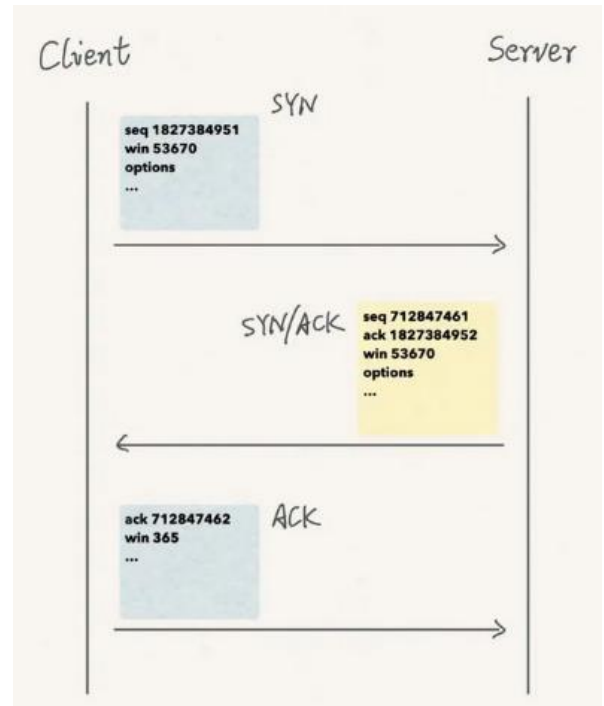
TCP needs 3 handshakes to establish the connection:

- The client sends a **SYN** message
- The server replies with an **SYN/ACK** message
- The client responds with an **ACK** message

The client initializes the **1st handshake**, sending over the **client's initial sequence number and params**. The initial sequence number is a random one, and it cannot be zero.

After receiving the message, the server responds with the **2nd handshake**. This time, we have two sequence numbers: **the server's ISN and the ACK number**. The server's ISN is a random number which is different from the client's. The **ACK number** is not a random number. It is based on the **client's sequence number + 1**.

In the **3rd handshake**, the client replies with an **ACK number** which is based on the **server's ISN + 1**.



TCP Connection Termination

The TCP Connection Termination needs a **Four-Way Handshake**.

- **Client** receives Close signal from App, **sends FIN (1)**
- **Server** receives FIN, **sends ACK (2)**, tells App to close. Then waits until App is ready to close
- **Client** receives ACK and waits for Server FIN
- When the App is ready to close, **server sends FIN (3)**
- The **client** receives the FIN and **sends an ACK (4)**
- The **server** receives the ACK and is afterwards closed.
- The **client** waits for Double Maximum Segment Life (MSL) Time and is afterwards closed.

The TCP Reset Function

TCP **includes a special connection reset feature** that allows devices to deal with problem situations, such as half-open connections or the receipt of unexpected message types. The device detecting the problem sends a TCP segment with the **RST (reset) flag set to 1**.

The TCP software generates a reset when:

- Receipt of any TCP segment from any device with which the device receiving the segment **does not currently have a connection** (other than a SYN requesting a new connection)
- Receipt of a message with an **invalid or incorrect Sequence Number or Acknowledgment Number field**, indicating that the message may belong to a prior connection
- Receipt of a SYN message **on a port where there is no process listening for connections**

13.3.3. TCP message formatting

The TCP header is 20 bytes for regular segments and more for those carrying options.

Bit(0)			Bit(15)	Bit(16)	Bit(31)
Source Port (16)				Destination Port (16)	
Sequence Number (32)					
Acknowledgment Number (32)					
Header Length (4)	Reserved (6)	Control Bits (6)		Window (16)	
Checksum (16)				Urgent (16)	
Options (0 or 32 if any)					
Application Layer Data (Size varies)					

13.3.4. TCP Maximum Segment Size (MSS)

Certain devices are *limited in the amount of space* they have for buffers to hold TCP segments and may wish to limit segment size to a relatively small value. *The MSS is a segment size that can never be exceeded*, regardless of how large the current window is. It must be chosen by balancing two competing performance issues:

- **Overhead Management:** The TCP header takes 20 bytes of data (with no options) and the IP header also uses 20 bytes of data (with no options). If we set the MSS too low, this results in very inefficient use of bandwidth. If we set it to 40 bytes, a maximum of 50 percent of each segment could actually be data and the rest would just be headers.
- **IP Fragmentation:** TCP segments will be packaged into IP datagrams that have their own size limit issues. The maximum transmission unit (MTU) of an underlying network. If a TCP segment is too large, it will lead to an IP datagram that is too large to be sent without fragmentation. Fragmentation reduces efficiency and increases the chances of part of a TCP segment being lost resulting in the entire segment needing to be retransmitted.

The standard MSS for TCP is 536 bytes (minimum MTU for IP networks is 576 bytes minus 40 bytes [2x headers]). This is as large as possible with avoiding fragmentation. If any TCP or IP options are used, the header grows and the minimum MTP will be exceeded. This leads to fragmentation.

A device can inform the other device *of the MSS it wants to use* (if not default) through parameter exchange during the connection establishment process. A device that chooses to do so includes in *its SYN message the TCP option called Maximum Segment Size*. The other device receives this option and records the MSS for the connection. *Each device in the connection may use a different MSS value*. Thanks to MTU path discovery, a device knows that the MTUs of the networks the segment will pass over are larger than the IP minimum of 576 bytes, for example Ethernet networks.

13.3.5. TCP Immediate data transfer: Push Function

TCP will generally accumulate data sent to it by an application process in a *buffer*. It chooses when and how to send data based solely on the sliding window system discussed. However, TCP includes a special *push function* to handle cases where data given to TCP needs to be sent *immediately* (for example a request for a web page).

13.3.6. TCP Priority data transfer: Urgent Function

TCP operates with a *first-in, first-out behavior*. However, TCP provides a *means for a process to prioritize* the sending of data in the form of its *urgent function* (for example when it is necessary to interrupt an applications data transfer, it must be communicated to the other end of the TCP connection immediately). It sets the *Urgent Pointer* field to an offset value that points to the last byte of urgent data in the segment.

When urgent data needs to be sent, the push function and the urgent function are usually invoked together.

13.3.7. TCP Segment Retransmission Timers and the Retransmission Queue

Retransmission Sequence:

- **Placement on Retransmission Queue, Timer Start:** As soon as a segment containing data is transmitted, a copy of the segment is placed in a data structure called the retransmission queue. A retransmission timer is started for the segment when it is placed on the queue.
- **Acknowledgment Processing:** If an acknowledgment is received for a segment before its timer expires, the segment is removed from the retransmission queue.
- **Retransmission Timeout:** If an acknowledgment is not received before the timer for a segment expires, a retransmission timeout occurs, and the segment is automatically retransmitted.

We don't want TCP to just keep retransmitting forever, so TCP will retransmit a lost segment *only a certain number of times* before concluding there is a problem and terminating the connection.

13.3.8. TCP Noncontiguous Acknowledgment Handling and Selective Acknowledgment (SACK)

Policies for dealing with outstanding unacknowledged segments, there are two approaches to handling retransmission in TCP:

- **Retransmit only timed-out segments:** This is the more conservative or optimistic approach. Only the segments whose timers are expired are retransmitted. This method is the best if the segments after the timed-out segment actually showed up. Saves bandwidth. If they did not, each segment would need to time out individually and be retransmitted. May cause performance degradation.
- **Retransmit all outstanding segments:** This is the more aggressive or pessimistic method. Whenever a segment times out, all still unacknowledged segments are resent. If many segments are lost, this method provides better performance. It may waste bandwidth on unnecessary retransmits.

SACK makes it possible to **Acknowledge** the segments which have been **received after a missing segment**. The Client sends a SACK instead of an ACK. The server knows which packets have been received after the missing packet. This way, it is not necessary to retransmit all outstanding segments.

13.3.9. TCP window size adjustment and flow control

When the server receives data from the client, it places it into the **buffer**. The server must then do two things with this data:

- **Acknowledgment:** The server must send an ACK back to the client to indicate that the data was received.
- **Transfer:** The server must process the data, transferring it to the destination application process.

In the basic sliding window system, data is acknowledged when received, but **not necessarily immediately transferred out of the buffer**. It is possible for the buffer **to fill up with received data faster than the receiving TCP can empty it**. When this occurs, the receiving device may need to **adjust the window size** to prevent the buffer from being overloaded. A device that reduces its receive window to zero is said to have **closed the window**.

13.3.10. TCP Congestion handling and congestion avoidance algorithms

When **congestion increases** on the network, **segments would be delayed** or dropped, which would cause them to be **retransmitted**. This would **increase the amount of traffic** on the network between client and server. This can lead to a vicious circle, resulting in a condition called congestion collapse. Because of this, the **TCP slow start and congestion avoidance states** are needed.

Slow start state

Slow start state is entered **at the beginning of the connection or after timeout**. Congestion window (cwnd) starts with initial cwnd and **increases by one or more MSS for each ACK received**.

Congestion avoidance

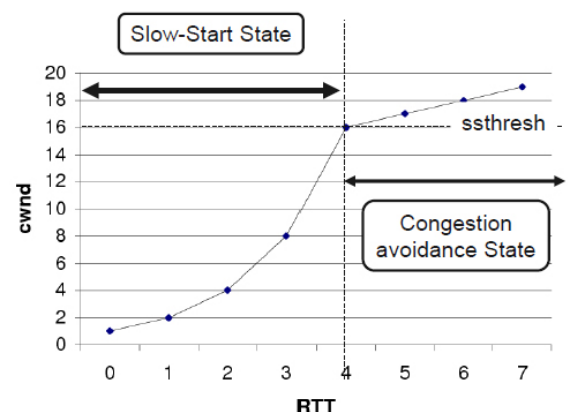
Congestion avoidance state **is entered when cwnd reaches slow start threshold (sssthresh)**. Cwnd **increases by one MSS for each RTT** (if no duplicate ACK arrives)

Fast Retransmit

Fast retransmit is a modification to the congestion avoidance algorithm. If the receiver does not receive a packet, it **keeps on sending the ACK** for the last received packet. When **the sender receives three duplicate ACKs, it assumes the packet is lost and retransmit it** without waiting for the retransmission timer to expire.

Fast Recovery

When fast retransmit is used to resend a lost segment, the device using it performs congestion avoidance, but **does not use slow start** to increase the transmission rate to back up again. The rationale for this is that since multiple ACKs were received by the sender, all indicating receipt of out-of-order segments, this indicates that several segments have already been removed from the flow of segments between the two devices. This improves performance compared to using regular Congestion Avoidance algorithm after fast retransmit.



14. NETWORK MANAGEMENT

What is Network Management for?

At least, the IT-Department *delivers services* to the customers. *These services are defined in the Service Level Agreements (SLA)* (availability, security, support, deployment). The Primary objective of service management is to *ensure that the IT services are aligned to the business needs* and actively support them. It is also increasingly important that IT acts as an *agent for change* to facilitate business transformation. So, service management and development is an ongoing dynamic process.

Key issues

- IT and business strategic planning
- Integrating and aligning IT and business goals
- Implementing continual improvement
- Measuring IT organization effectiveness and efficiency
- Optimizing costs and the Total Cost of Ownership (TCO)
- Achieving and demonstrating Return on Investment (ROI)
- Demonstrating the business value of IT
- Developing business and IT partnerships and relationships
- Improving project delivery success
- Outsourcing, insourcing and smart sourcing...

Key principles of IT Service Management

- Service Strategy
- Service Design
- Service Transition
- Service Operation
- Continual Service Improvement

14.1. FCAPS

FCAPS is a *Model/Framework* for network management systems. It has five categories: *Fault*, *Configuration*, *Accounting*, *Performance* and *Security*.

Fault Management	Configuration Management	Accounting Management	Performance Management	Security Management
Fault Detection	Resource Initialization	Track Service	Performance data collection	Selective resource access
Fault Correction	Network provisioning	Resource usage	Performance data analysis	Access logs
Network recovery	Backup & Restore	Combine costs for multiple resources	Consistent performance level	Event reporting
Alarm generation	Resource shut down	Set quotas for usage	Problem reporting	User access rights checking
Alarm handling	Change management	Audits	Capacity planning	Compliance
Alarm filtering	Pre-provisioning	Fraud reporting	Maintaining historical logs	Security related information distribution
Alarm correlation	Inventory / asset management			
Diagnostic Tests	Remote Configuration			
Error logging	Software distribution			
Error handling	Job initiation			
Error statistics	Job tracking			

14.1.1. Network Management Station

A *network operations center (NOC)* is usually a control center with many screens, clocks, depending on the subject weather report, maps, etc.

On one side is the network management station (NOC). On the other side are the agents (A process running on the clients, little helper software. Processes the data from the device and communicates with the NOC). This communication can work via pull or push:

- **Pull:** The Manager (NOC) has to ask for information, the agent is passive
- **Push:** The agent is active and sends information to the manager without it asking first.

Normally, both types are used together.

14.1.2. Protocols

There are a lot of protocols. For example:

- **SNMP**: Simple Network Management Protocol, monitoring
- **NetFlow**: Cisco application, monitors traffic
- **NETCONF**: used to request configuration data
- **IP SLA**: Used to track management performance
- **Traceroute**: Used to sniff out the different hops

14.2. DOCUMENTATION

It is useful to have a documentation. Should include:

- **Topology** (Physical Layer, Layer 2, Layer 3, ... depending on the network)
- **Devices** (Vendor, Model, SW-Release, Location, ...)
- **Connections** (Technology, Speed, Properties, Service Contract, ...)

Layer 1: Most errors happen on layer 1 (Defect, loose connections, wrong power budget, interference, ...)

Challenges: Keep it up to date, all necessary information listed, synchronization, use of the right tools, automation

14.3. OLD SCHOOL PROTOCOLS

- **SNMP – Simple Network Management Protocol**: Manager and Agents, communication via trap (push) and get/set (pull). Has a management information base (MIB). There are different Versions, v1, v2c and v3
MIB: Hierarchically organized OID (Object Identifier) collection. Types: scalar and tabular.
CPU Utilization: Cisco entry with Version, Community, Obtained CPU Value, IP Address, OID Number.
- **Syslog**: Developed in 1980ies. Protocol for sending logging messages / events. Device sends message / event to server. There are different severity levels: 0 – Emergency, 1 – Alert, 2 – Critical, 3 – Error, 4 – Warning, 5 – Notification, 6 – Informational and 7 – Debugging.
- **NetFlow**: Collecting traffic flow information. NetFlow Collector and NetFlow Exporter
Collecting Flow based. Flow: Source IP address, Destination IP address, Source port number, destination port number, layer 3 protocol type, type of service (ToS), Input logical interface.

14.4. NEW AGE

14.4.1. SDN – Software Defined Networking

The SDN-Architecture isolates the network in three different layers:

- **The application layer**: Supports the management unit with services to realize a system-independent network operation
- **The control layer**: represents the central SDN Controller Software. “the brain of the SDN”. The SDN controller is located on a server. It manages the guidelines as well as the traffic flow in the whole network.
- **The infrastructure layer**: Describes the whole physical switches in the network

These three layers communicate with each other by specific application programming Interfaces (API's) in the north and south direction.

SDN operating mode: SDN comprises some kinds of technologies.

- **Functional Isolation**: Network virtualization and automation through programmability
- **Network Control Plane**: Decides the flow of data packets through the network
- **Data Plane**: realizes the physical flow of the data packets from one point to another

Advantages of SDN: SDN allows an administrator to

- Change the routing rules of each switch (upgrade or downgrade, grade of control and security)
- See the end-to-end flow of packets
- Concentrate on only one central controller
- Virtualize even hardware and services that typically need dedicated hardware

15. BEGRIFFE | ABKÜRZUNGEN

- **ABR:** Area Border Router (OSPF) 3
- **ACK:** Acknowledgment (Control Message)
- **AfrinIC:** African Network Information Centre
- **ANSI:** American National Standards Institute
- **AP:** Access Point 2
- **API:** Application Programming Interface
- **APNIC:** Asia Pacific Network Information Centre
- **ARIN:** American Registry for Internet Numbers
- **ARP:** Address Resolution Protocol 3
- **ASBR:** Autonomous System Border Router (OSPF) 3
- **BOOTP:** Bootstrap Protocol 4
- **BPDU:** Bridge Protocol Data Unit (Used in STP) 2
- **BSSID:** Basic Service Set Identifier 2
- **CMDB:** Configuration Management database (management)
- **Cos:** Class of Service (used in networking to prioritize different types of traffic) 2
- **CRC:** Cyclic Redundancy Check (ACK Checksum) 2
- **CSMA/CA:** Carrier Sense Multiple Access / Collision Avoidance (Wireless) 2
- **CSMA/CD:** Carrier Sense Multiple Access / Collision Detection (Wired) 2
- **CTS:** Clear to Send (Wireless) 2
- **Cwnd:** Congestion window (TCP slow start) 4
- **DA:** Destination Address 2
- **DAD:** Duplicate Address Detection 3
- **dB:** Decibel 1
- **dBi:** Antenna gain compared to isotropic radiator 1
- **dBm:** Decibel ratio to 1mW 1
- **DCF:** Distributed Coordination Function (CSMA/CA - Wireless) 2
- **DHCP:** Dynamic Host Configuration Protocol 7
- **DIFS:** DCF Interframe Space (lowest priority) 2
- **DNS:** Domain Name System 7
- **DS:** Distribution System 2
- **ESSID:** Extended Service Set Identifier 2
- **FCAPS:** Fault, Configuration, Accounting, Performance, Security (management)
- **FCrDNS:** Forward Confirmed Reverse DNS 7
- **FCS:** Frame Check Sequence 2
- **FIB:** Forwarding Information Base 3
- **FTP:** File Transfer Protocol 7
- **HTTP:** Hypertext Transfer Protocol 7
- **IAB:** Internet Architecture Board
- **IANA:** Internet Assigned Numbers Authority
- **ICANN:** Internet Corporation for Assigned Names and Numbers
- **ICMP:** Internet Control Message Protocol (ICMPv4 for IPv4 and ICMPv6 for IPv6)
- **IEEE:** Institute for Electrical and Electronic Engineers
- **IETF:** Internet Engineering Task Force
- **IHL:** Internet Header Length (Length of IP header) 3
- **IMAP:** Internet Message Access Protocol 7
- **IPAM:** IP address management (management)
- **IRTF:** Internet Research Task Force
- **ISN:** Initial Sequence Number (TCP) 4
- **ISO:** International Organizations for Standardization
- **ISOC:** Internet Society
- **ISP:** Internet Service Provider
- **ITIL:** IT Infrastructure Library (management)
- **ITU:** International Telecommunications Union
- **LACNIC:** Regional Latin-American and Caribbean IP Address Registry
- **LACP:** Link Aggregation Protocol (standard defined in IEEE 802.3ad) 2
- **LAN:** Local Area Network
- **LLC:** Logical Link Control 2
- **LSA:** Link State Advertisement 3
- **LSDB:** Link State Database 3
- **MAC:** Media Access Control 2
- **MDI:** Medium Dependent Interface 1
- **MIB:** Management Information Base (management)
- **MII:** Medium Independent Interface 1
- **MIMO:** Multiple-input, multiple-output (Amount of Radios / Antenna) 1
- **MSS:** (TCP) Maximum Segment Size 4
- **MTTR:** Mean Time to Repair (management)
- **MTU:** Maximum Transmission Unit (IP) 3
- **NA:** Neighbor Advertisement Message (IPv6) 3
- **NAT:** Network Address Translation 3
- **NAV:** Network Allocation Vector 2
- **ND:** Neighbor Discovery
- **NIC:** Network Interface Card
- **NOC:** Network Operating Center (management)
- **NRZ:** No Return to Zero 1
- **NS:** Neighbor Solicitation (Message) (IPv6) 3
- **OID:** Object Identifier (management)
- **OSI:** Open Systems Interconnection
- **OSPF:** Open Shortest Path First 3
- **OSS:** Operational support system (management)
- **OUI:** Organizationally Unique Identifier (Part of MAC Address) 2
- **PAR:** Positive Acknowledgment with Retransmission (TCP) 4
- **PAT:** Port Address Translation (NAT overload) 3
- **PCS:** Physical Coding Sublayer 1
- **PDU:** Protocol Data Unit (form the data takes at each Layer)
- **PHY:** Physical Layer Device 1
- **PIFS:** PCF, Point Coordination Function IFS (middle priority) 2
- **PLCP:** Physical Layer Convergence Protocol 1
- **PMA:** Physical Medium Attachment 1
- **PMD:** Physical Medium Dependent 1
- **PoE:** Power over Ethernet 2
- **PVST:** Per-VLAN Spanning Tree 2
- **QoS:** Quality of Service (IP) 3
- **RA:** Router Advertisement Message (IPv6)
- **RAC:** Recursion Access Control (DNS) 7
- **rDNS:** Reverse DNS lookup 7
- **RFC:** Request for Comments (describes various standards in TCP/IP) 4
- **RF-MON:** Radio Frequency Monitor Modus (Wlan) 2
- **RIB:** Routing Information Base 3
- **RIP:** Routing Information Protocol 3
- **RIPE:** Réseaux IP Européans
- **ROI:** Return on Investment (management)
- **RR:** Resource Records 7
- **RSSI:** Received signal strength indication 1
- **RTS:** Request to Send (Wireless) 2

- **RTT:** Round Trip Time (TCP) 4
- **RZ:** Return to Zero 1
- **SA:** Source Address 2
- **SACK:** (TCP Noncontiguous Acknowledgment Handling and) Selective Acknowledgment 4
- **SDN:** Software defined network
- **SFD:** Start Frame Delimiter 2
- **SIFS:** Short Inter Frame Spacing (highest priority) 2
- **SISO:** Single-in, single-out (Amount of Radios / Antenna) 1
- **SLA:** Service Level Agreements (management)
- **SNMP:** Simple Network Management Protocol (management)
- **SNR:** Signal to Noise Ratio 1
- **SOA:** Start of Authority (DNS) 7
- **SPF:** Shortest Path First (Dijkstra) 3
- **Ssthresh:** Slow start threshold (TCP) 4
- **STP:** Spanning Tree Protocol 2
- **TA:** Transmitter Address 2
- **TCO:** Total Cost of Ownership (management)
- **TCP:** Transmission Control Protocol 4
- **TFTP:** Trivial File Transfer Protocol 7
- **TLD:** Top Level Domain 7
- **ToS:** Type of service (management)
- **UDP:** User Datagram Protocol 4
- **ULA:** Unique local address 3
- **UTP:** Unshielded Twisted Pair (copper cable) 1
- **VID:** VLAN identifier 2
- **VLSM:** Variable Length Subnet Masks / Masking 3
- **WAN:** Wide Area Network

16. HELPERS

Binär Umrechnungshelfer

4096	2048	1024	512	256	128	64	32	16	8 (2 ³)	4 (2 ²)	2 (2 ¹)	1 (2 ⁰)
------	------	------	-----	-----	-----	----	----	----	---------------------	---------------------	---------------------	---------------------

Hexadezimal Umrechnungshelfer

16 ⁰	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16 ¹	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
16 ²	256	512	768	1024	1280	1536	1792	2048	2304	2560	2816	3072	3328	3584	3840
16 ³	4096	8192	12 288	16 384	20 480	24 576	28 672	32 768	36 864	40 960	45 056	49 152	53 248	57 344	61 440

Sonst einfach Zahl / 16, Rest ergibt den hintersten HEX Wert, dann geteilte Zahl / 16 etc.

Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Bit Table

Term – wäre gerundet auf dez	Term (Binary)	Size in Bytes	Size in Bits
Byte (B)	Byte (B)	1 B	8 Bit
Kilobyte (KB)	Kibibyte (KiB)	1024 B (2 ¹⁰)	8192
Megabyte (MB)	Mebibyte (MiB)	1'048'576 B (2 ²⁰ or 1kB ²)	8'388'608
Gigabyte (GB)	Gibibyte (GiB)	1'073'741'824 B (2 ³⁰ or 1kB ³)	
Terabyte (TB)	Tebibyte (TiB)	1'099'511'627'776 B (2 ⁴⁰ or 1kB ⁴)	