

# Computer Networks 1 | CN1

## Summary

---

## CONTENTS

|   |           |
|---|-----------|
| <b>1. Application Layer (7,6,5)</b>                                     | <b>4</b>  |
| 1.1. HTTP   | 4         |
| 1.2. Common Ports   | 5         |
| 1.3. DNS  | 5         |
| 1.3.1. Record types   | 5         |
| 1.4. E-Mail   | 5         |
| <b>2. Transport Layer (4)</b>   | <b>6</b>  |
| 2.1. Primary responsibilities   | 6         |
| 2.2. TCP  | 6         |
| 2.2.1. Glossary   | 7         |
| 2.2.2. Reliability  | 7         |
| 2.2.3. Throughput   | 8         |
| 2.2.4. Flow control   | 8         |
| 2.2.5. Congestion control   | 8         |
| 2.3. UDP  | 10        |
| 2.4. QUIC   | 10        |
| <b>3. Network Layer (3)</b>   | <b>10</b> |
| 3.1. Subnetting   | 10        |
| 3.2. IPv6   | 10        |
| 3.2.1. Header   | 10        |
| 3.2.2. Glossary   | 11        |
| 3.2.3. Special addresses  | 11        |
| 3.2.4. Neighbor Discovery Protocol (NDP)                                | 13        |
| 3.2.5. Stateless Address Autoconfiguration (SLAAC)                      | 13        |
| 3.2.6. DHCPv6   | 14        |
| 3.3. IPv4   | 14        |
| 3.3.1. Network classes (private nets)                                   | 15        |
| 3.3.2. Subnetting   | 15        |
| 3.4. Routing  | 15        |
| 3.4.1. Control plane  | 15        |
| 3.4.2. Data plane   | 16        |
| 3.4.3. Dynamic  | 16        |
| <b>4. Data Link Layer (2)</b>   | <b>17</b> |
| 4.1. Ethernet   | 17        |
| 4.1.1. Carrier Sense Multiple Access with Collision Detection (CSMA/CD) | 17        |
| 4.1.2. Half-duplex vs Full-duplex                                       | 18        |
| 4.1.3. Ethernet II Frame  | 18        |
| 4.1.4. IEEE 802.3 Frame   | 18        |
| 4.1.5. Frame Check Sequence (FCS)                                       | 18        |
| 4.2. Error detection  | 18        |
| 4.3. MAC-Address  | 18        |

|   |           |
|---|-----------|
| 4.4. Address Resolution Protocol (ARP) .....                                  | 19        |
| 4.4.1. Discovery .....  | 19        |
| 4.4.2. Spoofing .....   | 20        |
| 4.5. Switch .....   | 20        |
| 4.5.1. Flooding .....   | 20        |
| 4.5.2. Filtering .....  | 20        |
| 4.5.3. Forwarding .....   | 20        |
| 4.5.4. EtherChannel = Link Aggregation Group (LAG) .....                      | 20        |
| 4.5.5. Link Aggregation Control Protocol (LACP) .....                         | 20        |
| 4.6. VLAN .....   | 20        |
| 4.6.1. Trunking .....   | 21        |
| 4.6.2. 802.1Q .....   | 21        |
| 4.6.3. Inter-VLAN Routing .....   | 21        |
| 4.6.4. Spanning Tree Protocol (STP) .....                                     | 21        |
| 4.6.5. Rapid Spanning Tree Protocol (RSTP) .....                              | 23        |
| 4.7. Wireless .....   | 23        |
| 4.7.1. Channel bonding .....  | 23        |
| 4.7.2. Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA) ..... | 24        |
| 4.7.3. Frame .....  | 24        |
| 4.7.4. Management features .....  | 25        |
| 4.7.5. Roaming .....  | 26        |
| <b>5. Physical Layer (1) .....</b>  | <b>27</b> |
| 5.1. Encodings .....  | 27        |
| 5.1.1. Manchester encoding .....  | 27        |
| 5.1.2. RZ (Return-to-Zero) .....  | 27        |
| 5.1.3. NRZ (Non-Return-to-Zero) .....   | 27        |
| 5.1.4. 8b/10b (Clock recovery) .....  | 27        |
| 5.2. Power and dB .....   | 27        |
| 5.2.1. Law of 3s .....  | 28        |
| 5.2.2. Law of 10s .....   | 28        |
| 5.3. Modulation .....   | 28        |
| 5.4. Fiber media .....  | 28        |
| 5.4.1. Eye Diagram .....  | 28        |
| 5.4.2. Attenuation .....  | 28        |
| 5.4.3. Dispersion .....   | 29        |
| 5.4.4. Regeneration .....   | 29        |
| 5.5. Frequency .....  | 29        |
| 5.6. Calculations .....   | 29        |
| 5.6.1. Speed of Signals .....   | 29        |
| 5.6.2. Optical budget .....   | 29        |
| <b>6. Cisco .....</b>   | <b>30</b> |
| 6.1. Router setup .....   | 30        |
| 6.2. Interfaces .....   | 30        |
| 6.2.1. Static IP Assignment .....   | 30        |
| 6.2.2. DHCP Assignment .....  | 30        |
| 6.2.3. Show .....   | 30        |
| 6.3. VLAN .....   | 30        |
| 6.3.1. Assign IP .....  | 30        |
| 6.3.2. Access Port .....  | 30        |

|   |    |
|---|----|
| 6.3.3. Access Port .....                              | 30 |
| 6.3.4. VTP (Virtual Trunk Protocol) .....             | 31 |
| 6.3.5. LACP (Link Aggregation Control Protocol) ..... | 31 |
| 6.3.6. Load Balancing .....                           | 31 |
| 6.3.7. STP (Spanning Tree Protocol) .....             | 31 |
| 6.4. Routing .....                                    | 31 |
| 6.4.1. Static .....                                   | 31 |
| 6.4.2. OSPF .....                                     | 31 |
| 6.4.3. Show .....                                     | 32 |
| 6.5. DHCP .....                                       | 32 |
| 6.5.1. Create Pool .....                              | 32 |
| 6.5.2. Relay Agent .....                              | 32 |
| 6.6. NAT .....  | 32 |
| 6.6.1. IF Inside .....                                | 32 |
| 6.6.2. IF Outside .....                               | 32 |
| 6.6.3. ACL (Access Control List) .....                | 32 |
| 6.6.4. PAT (Nat overload) .....                       | 32 |
| 6.7. IPv6 .....                                       | 32 |
| 6.7.1. DHCPv6 .....                                   | 32 |
| 6.8. Troubleshooting .....                            | 32 |
| 6.8.1. Ping .....                                     | 32 |
| 6.8.2. Traceroute .....                               | 32 |

## 1. APPLICATION LAYER (7,6,5)

Combines Layers 7 (Application), 6 (Presentation) and 5 (Session).

### 1.1. HTTP

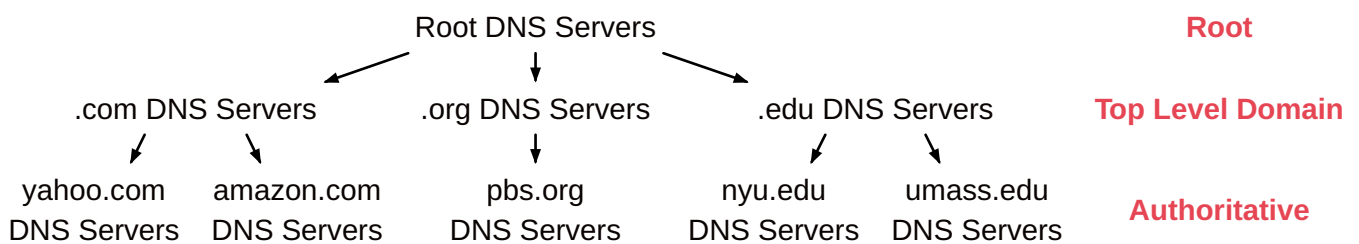
| <i>Feature</i>                  | <i>HTTP/1.0</i>            | <i>HTTP/1.1</i>                       | <i>HTTP/2</i>   | <i>HTTP/3</i>                                |
|---------------------------------|----------------------------|---------------------------------------|---|--|
| <b>Connection Management</b>    | One request per connection | Persistent connections by default     | Multiplexing allows multiple streams                                | Uses QUIC for multiplexing                   |
| <b>Request Methods</b>          | Limited (GET, POST, HEAD)  | Enhanced (PUT, DELETE, OPTIONS, etc.) | Same as 1.1   | Same as 1.1                                  |
| <b>Caching</b>                  | Basic caching support      | Improved caching with validation      | Advanced caching capabilities                                       | Same as 2 but with improved mechanisms       |
| <b>Header Compression</b>       | None                       | None                                  | HPACK (header compression)  | QPACK (header compression)                   |
| <b>Server Push</b>              | Not supported              | Not supported                         | Supported (automatic resource pushing)                              | Enhanced support for server push             |
| <b>Performance Improvements</b> | None                       | Minor improvements over 1.0           | Significant improvements in performance and latency                 | Further improvements in speed and efficiency |
| <b>SSL/TLS Support</b>          | Not inherent               | Not inherent, but commonly supported  | Built-in support with ALPN (Application-Layer Protocol Negotiation) | Uses QUIC, which incorporates TLS 1.3        |
| <b>Transport Protocol</b>       | TCP                        | TCP                                   | TCP   | QUIC   |

## 1.2. COMMON PORTS

| <i>Protocol</i> | <i>Port</i>              | <i>Layer 4</i> |
|-----------------|--------------------------|----------------|
| DNS             | 53                       | UDP, TCP       |
| HTTP            | 80                       | TCP            |
| HTTPS           | 443                      | TCP            |
| FTP             | 20, 21                   | TCP            |
| SMTP            | 25 (server) 587 (client) | TCP            |
| POP3            | 110                      | TCP            |
| DHCP            | 67 (server) 68 (client)  | UDP            |

## 1.3. DNS

Nameservers resolve domains to IP's through a distributed, hierarchical database.



| <i>Term</i>     | <i>Definition</i>   |
|-----------------|---|
| Iterated query  | Local DNS server iteratively asks one server after the other, descending the domain name hierarchy step after step.   |
| Recursive query | Local DNS server asks root server for domain, which in turn asks the TLD server, which in turn asks the authoritative server etc. until the "call stack" unwinds and returns the fully resolved domain to the query sender. |
| Caching         | Client-side temporary storage of DNS lookup information.  |

### 1.3.1. Record types

| <i>Type</i> | <i>Name</i> | <i>Value</i>                                 |
|-------------|-------------|--|
| A           | hostname    | IPv4 address                                 |
| AAAA        | hostname    | IPv6 address                                 |
| CNAME       | alias       | canonical name                               |
| NS          | domain      | hostname of authoritative NS for this domain |
| MX          | domain      | name of mailserver                           |
| PTR         | IP          | domain                                       |

## 1.4. E-MAIL

| <i>Term</i>                             | <i>Definition</i>  |
|---|--|
| SMTP (Simple Mail Transfer Protocol)    | Used to send email messages from a client to a mail server or between mail servers |
| IMAP (Internet Message Access Protocol) | Synchronizes email across multiple devices without downloading them                |

| Term   | Definition   |
|--|--|
| POP3 (Post Office Protocol version 3)        | Downloads email messages onto the user's local device, often removing them from the server afterward   |
| MIME (Multipurpose Internet Mail Extensions) | An extension of the Internet email protocol that allows multimedia content to be transmitted via email |
| SPF (Sender Policy Framework)                | An email authentication method that helps prevent spoofing   |
| MUA (Mail User Agent)                        | Allows users to send, receive, and organize their emails (eg. Mozilla Thunderbird)                     |
| MTA (Mail Transfer Agent)                    | Transfers email messages between servers (eg. Postfix)   |
| MDA (Mail Delivery Agent)                    | Processes incoming messages from the MTA and places them in the user's mailbox                         |

## 2. TRANSPORT LAYER (4)

Segment size: 1440-1480b when using IPv4, <=1460b when using IPv6

### 2.1. PRIMARY RESPONSIBILITIES

- Process-to-process delivery (distinguish between multiple applications via ports)
- Ensure reliable transfer (acknowledgments, retransmissions & reordering)
- Flow control (sender does not overwhelm receiver)
- Congestion control (network is not overloaded)

| Term           | Definition  |
|----------------|---|
| Port           | <b>16 bit long</b> numbers ( <b>0-65535</b> ) for identifying applications to send packets to.<br><b>Well-Known: 0-1023</b> for universal TCP/IP applications, managed by the IANA.<br><b>Registered: 1024-49151</b> for known applications, also managed by the IANA.<br><b>Private: 49152-65535</b> for custom applications, not managed by the IANA. |
| Socket         | Combination of <b>IP:Port</b> .   |
| Multiplexing   | Sending data from multiple sockets at sender.   |
| Demultiplexing | Delivering segments to correct socket at receiver.  |
| Checksum       | Detect errors (i.e., flipped bits) in transmitted segment.  |

### 2.2. TCP

Connection-oriented, bidirectional, reliable, managed data flow.

|                 |          |   |   |   |   |   |                        |             |  |
|-----------------|----------|---|---|---|---|---|------------------------|-------------|--|
| ← 32 bit →      |          |   |   |   |   |   |                        |             |  |
| Source Port     |          |   |   |   |   |   | Destination Port       |             |  |
| Sequence Number |          |   |   |   |   |   | Acknowledgement Number |             |  |
| Offset          | Reserved | U | A | P | R | S | F                      | Window Size |  |
|                 |          | R | C | S | S | Y | I                      |             |  |
|                 |          | G | K | H | T | N | N                      |             |  |
| Checksum        |          |   |   |   |   |   | Urgent Pointer         |             |  |
| Options         |          |   |   |   |   |   |                        | Padding     |  |
| Data            |          |   |   |   |   |   |                        |             |  |

| Field                | Definition                                 |
|----------------------|--|
| U R G (1 bit)        | Urgent Pointer                             |
| A C K (1 bit)        | Acknowledgement flag                       |
| P S H (1 bit)        | Push flag                                  |
| R S T (1 bit)        | Reset flag                                 |
| S Y N (1 bit)        | synchronize flag                           |
| F I N (1 bit)        | Finish flag                                |
| Window Size (16 bit) | Sender's receive window                    |
| Options (24 bit)     | Variable length. MSS, Window scaling, etc. |
| Padding (8 bit)      | Variable length                            |
| Data (32 bit)        | Variable length                            |

### 2.2.1. Glossary

|   | Term                 | Definition   |
|---|----------------------|--|
| <pre> sequenceDiagram     participant Client     participant Server     Note over Client: SYN     Client-&gt;&gt;Server: SYN     Note over Server: SYN+ACK     Server-&gt;&gt;Client: SYN+ACK     Note over Client: ACK     Client-&gt;&gt;Server: ACK     Note over Client: SEQ (1)     Client-&gt;&gt;Server: SEQ (1)     Note over Server: ACK (2)     Server-&gt;&gt;Client: ACK (2)     Note over Client: SEQ (2)     Client-&gt;&gt;Server: SEQ (2)     Note over Server: ACK (3)     Server-&gt;&gt;Client: ACK (3)     Note over Client: FIN     Client-&gt;&gt;Server: FIN     Note over Server: ACK     Server-&gt;&gt;Client: ACK     Note over Client: FIN     Client-&gt;&gt;Server: FIN     Note over Server: ACK     Server-&gt;&gt;Client: ACK     </pre> | Handshake            | Agreement on <b>starting sequence numbers</b> , <b>maximum segment size</b> and <b>window scaling</b> .<br>1) SYN<br>2) SYN+ACK<br>3) ACK  |
|   | FIN                  | Termination of a connection.<br>1) FIN<br>2) ACK<br>3) FIN<br>4) ACK   |
|   | Round Trip Time      | <b>RTT</b> is the time it takes for a packet to be sent to the receiver and acknowledged back to the sender.   |
|   | Buffer size          | Maximum amount of data (measured in bytes) that can be stored in memory while waiting to be processed or transmitted.  |
|   | Maximum Segment Size | <b>MSS</b> is the maximum payload size of a TCP packet. In IPv4 networks, typically, the size of the MSS is <b>1460 bytes</b> because it is encapsulated in the data link layer Ethernet frame size of <b>1500 bytes</b> . -(20 bytes for IP header + 20 bytes for TCP header) |
|   |                      |  |
|   |                      |  |
|   |                      |  |
|   |                      |  |
|   |                      |  |

### 2.2.2. Reliability

| Term                   | Definition  |
|------------------------|---|
| Sequence numbers       | <b>SEQ</b> ensures that the packets arrive or can be reassembled in order.  |
| Acknowledgement        | <b>ACK</b> ensures that the receiver gets all of the packets.   |
| Retransmission timeout | If an acknowledgment is not received before the timer for a segment expires, a retransmission timeout occurs, and the segment is <b>automatically retransmitted</b> . |
| Packet loss rate       | Measures how many packets of the ones being sent actually arrive.   |

| Term                  | Definition   |
|-----------------------|--|
| Duplicate ACKs        | A duplicate ACK occurs when a receiver receives a segment of data that is not the next expected segment, prompting it to send back the <b>same acknowledgment</b> of the <b>last correctly received</b> packet multiple times. This signals to the sender that some packets might be lost or that the data is arriving out of order.               |
| Triple Duplicate ACKs | A triple duplicate ACK specifically refers to the situation where the receiver sends <b>three duplicate ACKs</b> in a row for the same segment. This particular signal indicates to the sender that a packet has likely been lost. In response, TCP will trigger a <b>fast retransmission</b> of the missing packet without waiting for a timeout. |

### 2.2.3. Throughput

| Term                     | Definition  |
|--------------------------|---|
| Throughput               | Denoted by $T$ , is the amount of data that can be transmitted during a specified time.<br>$T = \frac{W}{R} \leq C_{L3}$  |
| Continuous sending       | Sender transmits a stream of data packets in the given window size <b>without waiting for acknowledgments</b> .   |
| Delayed / Cumulative ACK | Receiver waits for a short period to acknowledge <b>multiple segments</b> with a <b>single ACK</b> .  |
| Selective ACK            | Instead of asking for a retransmission of all missing segments, <b>SACK</b> (specified by the receiver) allows the sender to send only the lost segments, significantly improving efficiency. |

### 2.2.4. Flow control

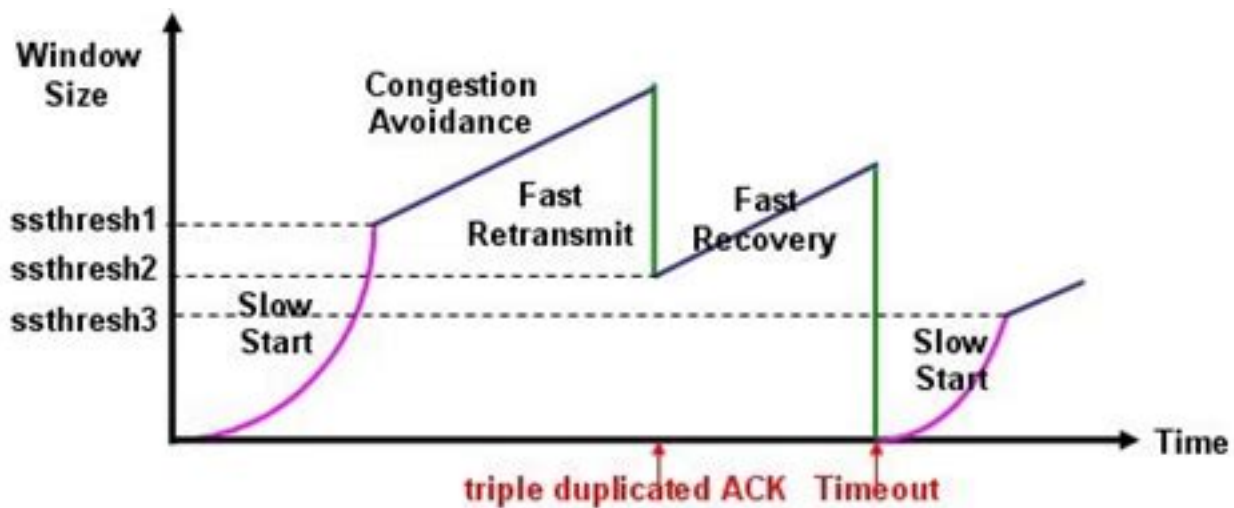
So that the sender does not **overwhelm the receiver**.

| Term            | Definition  |
|-----------------|---|
| Window Size     | Denoted by $W$ , is a <b>16 bit</b> number sent with each packet by the receiver inside of the <b>rwnd</b> header field, indicating the amount of data he still has space for. If sender receives a window size of 0, it starts a persistence timer. When the persistence timer goes off, the sending TCP sends a special segment called a window probe to query the receiver periodically and find out if the window has been increased. |
| Window scale    | Used when the TCP window size needs to be increased beyond the traditional maximum of 65,535 bytes due to the demands of high-speed networks. If the handshake header includes the <b>window scale option</b> and the packet header includes the <b>scaling factor</b> (max value of 14) then the effective window size is calculated as such: <b>window size * 2<sup>scaling factor</sup></b>  |
| Receiver Window | Managed by the receiver, who sends out window sizes to the sender. The window sizes announce the number of <b>bytes still free</b> in the receiver buffer   |

### 2.2.5. Congestion control

To prevent **network congestion**.





| Term                 | Definition  |
|----------------------|---|
| Congestion window    |   |
| Sliding Window       | Describes the process of the congestion window sliding to the right after receiving ACKs.   |
| Slow start           | Gradual growth (doubling <b>cwnd</b> every <b>RTT</b> ) within the congestion window size at the start of a connection or after a period of state of no activity.<br><b>Purpose:</b> Allows the sender to probe the available bandwidth in a controlled way.                          |
| Congestion avoidance | Transition from sluggish start to congestion avoidance segment after accomplishing a threshold.<br><b>Purpose:</b> Maintains a truthful share of the community bandwidth even as heading off excessive congestion.  |
| Fast Retransmit      | Detects packet loss through duplicate acknowledgments and triggers speedy retransmission without waiting for the <b>retransmission timeout</b> .<br><b>Purpose:</b> Speeds up the recuperation method with the aid of retransmitting lost packets without looking ahead to a timeout. |
| Fast Recovery        | Enters a quick healing state after detecting packet loss, lowering congestion window and transitioning to congestion avoidance.<br><b>Purpose:</b> Accelerates healing from congestion by way of avoiding a complete go back to slow begin after packet loss.                         |

| Term | Definition  |
|------|---|
| AIMD | Adjusts the congestion window size based on network situations following the <b>Additive Increase, Multiplicative Decrease</b> principle.<br><b>Purpose:</b> Provides a balanced approach by way of linearly growing the window all through congestion avoidance and halving it on packet loss. |

### 2.3. UDP

| ← 32 bit →  |                  |
|-------------|------------------|
| Source Port | Destination Port |
| Length      | Checksum         |
| Data        |                  |

### 2.4. QUIC

Actually a layer 7 Protocol, running on top of UDP

## 3. NETWORK LAYER (3)

Packet size: **1500b**

### 3.1. SUBNETTING

Dividing a **/X** network into **n** amount of **/Y** subnets:  $2^{Y-X} = n$ .

Eg: Dividing a **/16** network into **/24** subnets will yield **256** subnets, because  $2^{24-16} = 2^8 = 256$

### 3.2. IPV6

#### 3.2.1. Header

| ← 32 bit →                     |               |            |
|--------------------------------|---------------|------------|
| Version                        | Traffic class | Flow label |
| Payload length                 | Next header   | Hop limit  |
| Source address (128 bits)      |               |            |
| Destination address (128 bits) |               |            |

| Field                   | Definition  |
|-------------------------|---|
| Version (4 bit)         | Version of IP Protocol (always 6)   |
| Traffic class (8 bit)   | Priority + Class. Used for differentiating level of service and packet types        |
| Flow label (20 bit)     | Used to identify packets belonging to the same flow                                 |
| Payload length (16 bit) | Size of the data payload in bytes that follows the IPv6 header. max is 65'535 bytes |
| Next header (8 bit)     | Type of optional header following the IPv6 header.                                  |
| Hop limit (8 bit)       | Maximum number of hops a packet can take before being discarded.                    |

### 3.2.2. Glossary

| Term  | Definition   |
|---|--|
| Extension Header                                    | Additional headers used in IPv6 to provide optional information. These can define aspects like payload size, routing, or fragmentation.  |
| DHCPv6  | Dynamic Host Configuration Protocol for IPv6; this allows servers to assign IPv6 addresses dynamically from a pool, similar to DHCP for IPv4.                                      |
| NAT64   | Network Address Translation from IPv6 to IPv4 and vice versa; it facilitates communication between IPv6 and IPv4 networks.   |
| Neighbor Discovery Protocol ( <b>NDP</b> )          | A protocol in IPv6 for discovering other network nodes, determining their link-layer addresses, and ensuring that addresses are valid and reachable.                               |
| Internet Control Message Protocol ( <b>ICMPv6</b> ) | A crucial part of IPv6 that handles error messages and operational queries, with an expanded role compared to ICMP in IPv4.  |
| MTU   | Maximum Transmission Unit; the size of the largest packet that can be sent in a single frame over a network medium. IPv6 can handle larger MTUs compared to IPv4. Default is 1500b |
| Jumbo frame   | MTU of 9000b   |
| Multicast Listener Discovery ( <b>MLD</b> )         | IPv6 multicast routers can use MLD to discover multicast listeners on a directly attached link.  |
| Path MTU Discovery ( <b>PMTUD</b> )                 | Protocol for determining the Maximum Transmission Unit (MTU) size on the network path between two hosts, usually with the goal of avoiding IP fragmentation.                       |

### 3.2.3. Special addresses

| Term                                | Definition   |
|-------------------------------------|--|
| Link-local Address                  | <b>FE80::/10</b> Used for local communication between devices on the same network segment.   |
| Global Unicast Address              | <b>2000::/3</b> A globally routable address, these addresses are equivalent to public IPv4 addresses and can be reached over the internet.                           |
| Unique Local Address ( <b>ULA</b> ) | <b>FC00::/7</b> An address for local communication that is not routable on the global internet, similar to private addresses in IPv4.                                |
| Multicast Address                   | <b>FF00::/8</b> An address that enables a single packet to be sent to multiple destinations simultaneously.  |
| Anycast Address                     | An address assigned to multiple interfaces, where a packet sent to an anycast address is routed to the nearest (in terms of routing distance) interface.             |
| Reserved Address                    | Certain ranges in IPv6 are reserved for future use or specific functions. For example, addresses starting with <b>::/128</b> are reserved for unspecified addresses. |
| Documentation Address               | <b>2001:DB8::/32</b> Designated specifically for use in documentation and examples, ensuring it does not conflict with real-world addresses.                         |

| <i>Term</i>                  | <i>Definition</i>  |
|------------------------------|--|
| Link-local Multicast Address | <b>FF02::/16</b> Part of the link-local address range; it enables devices to communicate within a local network without requiring an external routing address. |

| <i>Addresses</i>   | <i>Range</i> | <i>Scope</i> |
|--------------------|--------------|--------------|
| Unspecified        | ::/128       | n/a          |
| Loopback           | ::1          | Host         |
| IPv4-Embedded      | 64:ff9b::/96 | n/a          |
| Discard-Only       | 100::/64     | n/a          |
| Link-Local         | fe80::/10    | Link         |
| Global Unicast     | 2000::/3     | Global       |
| Unique Local (ULA) | fc00::/7     | Global       |
| Multicast          | ff00::/8     | Variable     |

### 3.2.3.1. Multicast

| <i>Term</i>       | <i>Definition</i>  |
|-------------------|--|
| ff02::1           | All nodes, within scope 2 (link-local).  |
| ff02::2           | All routers, within scope 2 (link-local).  |
| ff02::1:ffxx:xxxx | The IPv6 node joins a solicited multicast address group from all the interfaces where unicast and anycast addresses are configured. Its scope is the link-local. |

### 3.2.3.2. DHCPv6

| <i>Term</i> | <i>Definition</i>   |
|-------------|---|
| ff02::1:2   | A link-scoped multicast address used by a client to communicate with neighboring (i.e., on-link) relay agents and servers. All servers and relay agents are members of this multicast group.                            |
| ff05::1:3   | A site-scoped multicast address used by a relay agent to communicate with servers, either because the relay agent wants to send messages to all servers or because it does not know the unicast address of the servers. |

### 3.2.3.3. IPv6 Extension Headers currently defined

| <i>Term</i>                    | <i>Definition</i>  |
|--------------------------------|--|
| Routing                        | Used by an IPv6 source to list one or more intermediate nodes to be “visited” on the way to a packet’s destination. There are different types of routing headers defined for different uses. |
| Fragmentation                  | Used by an IPv6 source to send a packet larger than would fit in the path MTU to its destination.  |
| Authentication                 | Used by IPsec to provide security services like integrity and data origin authentication to IPv6 traffic.  |
| Encapsulating Security Payload | Used by IPsec to provide security services like confidentiality and/or integrity to IPv6 packets.  |
| Hop-by-Hop Option              | Used to carry optional information that may be examined and processed by every node along a packet’s delivery path.  |

| <i>Term</i>         | <i>Definition</i>   |
|---------------------|---|
| Destination Options | Optional information to be examined by the destination node |

### 3.2.4. Neighbor Discovery Protocol (NDP)

#### 3.2.4.1. Host - Router Discovery Functions

| <i>Term</i>               | <i>Definition</i>  |
|---------------------------|--|
| Router discovery          | Hosts can locate routers residing on attached links.                     |
| Prefix discovery          | Hosts can discover address prefixes that are on-link for attached links. |
| Parameter discovery       | Hosts can find parameters (e.g., MTU).                                   |
| Address autoconfiguration | Stateless configuration of addresses of network interfaces.              |
| Redirect                  | Provide a better next-hop route for certain destinations.                |

#### 3.2.4.2. Host - Host Communication Functions

| <i>Term</i>                             | <i>Definition</i>   |
|---|---|
| Address resolution                      | Mapping between IP addresses and link-layer addresses. This is equivalent to ARP for IPv4. This function allows to resolve the link-layer address of another node in the link when only the IPv6 address of that node is known. |
| Next-hop determination                  | Hosts can find next-hop routers for a destination.  |
| Neighbor unreachability detection (NUD) | Determine that a neighbor is no longer reachable on the link.   |
| Duplicate address detection (DAD)       | Nodes can check whether an address is already in use.   |

#### 3.2.4.3. Packet types

| <i>Name</i>                          | <i>Type</i> | <i>Description</i>   |
|--------------------------------------|-------------|--|
| Router Solicitation ( <b>RS</b> )    | 133         | To locate routers on an attached link.   |
| Router Advertisement ( <b>RA</b> )   | 134         | Used by routers to advertise their presence periodically or in response to a RS message. |
| Neighbor Solicitation ( <b>NS</b> )  | 135         | To find the MAC-address of the neighbor or to check if the neighbor is still reachable.  |
| Neighbor Advertisement ( <b>NA</b> ) | 136         | To respond to a Neighbor Solicitation message.   |
| Redirect                             | 137         | To point the host to a better first hop router for a destination.                        |

### 3.2.5. Stateless Address Autoconfiguration (SLAAC)

A method for automatically configuring IPv6 addresses without a DHCP server, relying on local network information.

#### 3.2.5.1. Autoconfigure link-local address

Mac address: **70:07:12:34:56:78**

- 1) Flip **7th** bit: **72:07:12:34:56:78**. If it is "0", the address is locally administered and if it is "1", the address is globally unique.
- 2) Insert **FFFE** in the middle: **7207:12FF:FE34:5678**
- 3) Combine with link-local prefix: **FE80::7207:12FF:FE34:5678**

New address: **FE80::7207:12FF:FE34:5678**

### 3.2.5.2. Perform Duplicate Address Detection (DAD)

To make sure that the address is actually unique in the local segment.

Upon configuring an IPv6 address, every node joins a **multicast group** identified by the address **FF02::1:FFxx:xxxx** where xx:xxxx are the **last 6 hexadecimal values** in the IPv6 unicast address, eg. FF02::1:FF**34:5678**

- 1) The host sends a Neighbor Solicitation message from the Unspecified Address (::) to the Solicited Node multicast address.
- 2) If the generated address is in use, the host using that address sends a Neighbor Advertisement back. The sending host then knows the tentative address can not be used.
- 3) The host then proceeds to generate a new address and sends a new Neighbor Solicitation message to the link.
- 4) If there is no reply after some time, the host informs all the other hosts that it uses this address and it sends a Neighbor Advertisement message to the All Nodes address.
- 5) The host assigns the address to the interface and now has an active IPv6 link. This is the so-called Link-local Address Assignment.

### 3.2.5.3. Router search

- 1) Router solicitation
- 2) Router advertisement

### 3.2.5.4. Generating global unicast address

Based on the information from the Router Advertisement, the host generates a global unicast address and wants to know if it is available to use, so it does the DAD process again. If it is not a duplicate, the host will use it.

## 3.2.6. DHCPv6

### 3.2.6.1. Flags

| Term | Definition   |
|------|--|
| A    | Host can perform SLAAC to generate its own IPv6 address based on the prefix(es) contained in the RA message. |
| O    | Host can fetch additional options from the DHCPv6. The DHCPv6 does not provide IPv6 addresses in this case.  |
| M    | Host will get its IP address and additional options from a DHCPv6 server.                                    |
| L    | The prefix shared in the RA is reachable on the link.  |

## 3.3. IPV4

|                        |     |          |     |                 |    |                 |  |
|------------------------|-----|----------|-----|-----------------|----|-----------------|--|
| ← 32 bit →             |     |          |     |                 |    |                 |  |
| Version                | IHL | DSCP     | ECN | Total Length    |    |                 |  |
| Identification         |     |          | RS  | DF              | MF | Fragment Offset |  |
| Time to Live           |     | Protocol |     | Header Checksum |    |                 |  |
| Source IP Address      |     |          |     |                 |    |                 |  |
| Destination IP Address |     |          |     |                 |    |                 |  |
| Options (if IHL > 5)   |     |          |     |                 |    |                 |  |
| Data                   |     |          |     |                 |    |                 |  |

| <i>Field</i>                  | <i>Definition</i>  |
|-------------------------------|--|
| Version (4 bit)               | Version of IP Protocol (always 4)  |
| IHL (4 bit)                   | Internet Header Length (in 32bit words. min 5)   |
| DSCP (6 bit)                  | Differentiated Services Code Point   |
| ECN (2 bit)                   | Explicit Congestion Notification   |
| Total Length (16 bit)         | Header + Data in bytes. max is 65'535  |
| Identification (16 bit)       | Used to uniquely identify each packet sent from a source host. It helps in reassembling fragmented packets at the destination. |
| RS (1 bit)                    | Reserved (must be zero)  |
| DF (1 bit)                    | Don't Fragment   |
| MF (1 bit)                    | More Fragments   |
| Fragment Offset (13 bit)      | Position of the fragment. Must be multiplied by 8 to extrapolate the position of the fragment inside of a packet.              |
| Time to Live (8 bit)          | Hop limit  |
| Protocol (8 bit)              | Protocol of data portion   |
| Options (if IHL > 5) (32 bit) | Variable in length. Can specify timestamps, record route or other settings   |
| Data (32 bit)                 | Variable in length   |

### 3.3.1. Network classes (private nets)

| <i>Term</i> | <i>Definition</i>  |
|-------------|--|
| A           | <b>10.0.0.0</b> - 10.255.255.255 (10/ <b>8</b> prefix)           |
| B           | <b>172.16.0.0</b> - 172.31.255.255 (172.16/ <b>12</b> prefix)    |
| C           | <b>192.168.0.0</b> - 192.168.255.255 (192.168/ <b>16</b> prefix) |

### 3.3.2. Subnetting

#### 3.3.2.1. Calculating subnet mask

/24 = **1111 1111 . 1111 1111 . 1111 1111 . 0000 0000** = 255.255.255.0

/10 = **1111 1111 . 1100 0000 . 0000 0000 . 0000 0000** = 255.192.0.0

#### 3.3.2.2. Calculating increment

Address increment =  $\frac{\text{amount of addresses}}{256}$  or  $2^{8-(\text{mask mod } 8)}$

Let there be 4 subsequent networks starting with 10.0.0.0, each being /20

Amount of addresses =  $2^{32-20} = 2^{12} = 4096$ . Increment =  $\frac{4096}{256} = 16$

Alternatively:  $2^{8-(20 \bmod 8)} = 2^{8-4} = 2^4 = 16$

Networks = 10.0.0.0/20, 10.0.16.0/20, 10.0.32.0/20, 10.0.48.0/20

## 3.4. ROUTING

### 3.4.1. Control plane

Includes functions and processes that determine which path to use to send the packet or frame. The control plane is responsible for populating the routing table, drawing network topology, forwarding table, and hence enabling the data plane functions. This means here the router makes its decision.

- Network-wide logic
- Determines how datagram is routed among routers along end-to-end path from source to destination host

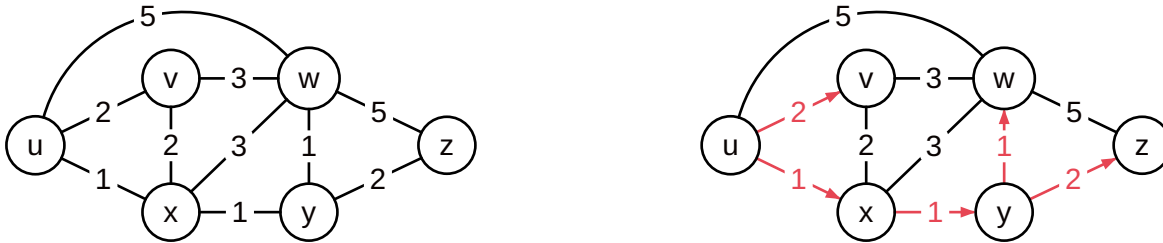
### 3.4.2. Data plane

Includes functions and processes that forward packets/frames from one interface to another based on control plane logic. Routing table, forwarding table and the routing logic constitute the data plane function. Data plane packet goes through the router and incoming and outgoing of frames are done based on control plane logic.

- Local, per-router function
- Determines how datagram arriving on router input port is forwarded to router output port

### 3.4.3. Dynamic

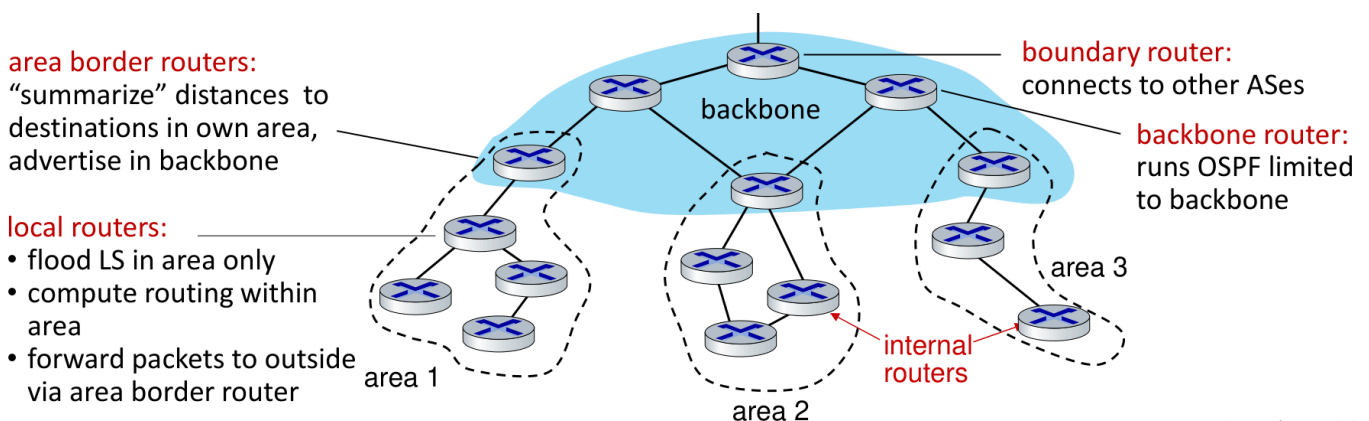
#### 3.4.3.1. Dijkstra's algorithm (Link State)



| Step | $N'$   | $D(v), p(v)$ | $D(w), p(w)$ | $D(x), p(x)$ | $D(y), p(y)$ | $D(z), p(z)$ |
|------|--------|--------------|--------------|--------------|--------------|--------------|
| 0    | u      | 2, u         | 5, u         | 1, u         | $\infty$     | $\infty$     |
| 1    | ux     | 2, u         | 4, x         |              | 2, x         | $\infty$     |
| 2    | uxy    | 2, u         | 3, y         |              |              | 4, y         |
| 3    | uxyv   |              | 3, y         |              |              | 4, y         |
| 4    | uxyvw  |              |              |              |              | 4, y         |
| 5    | uxyvwz |              |              |              |              | 4, y         |

#### 3.4.3.2. OSPF (Open Shortest Path First)

- Each router floods OSPF link-state advertisements (directly over IP rather than using TCP/UDP) to all other routers in entire AS
- Multiple link costs metrics possible: bandwidth, delay
- Each router has full topology, uses Dijkstra's algorithm to compute forwarding table



#### 3.4.3.3. RIP (Routing Information Protocol)

Not used anymore, uses distance vector algorithm to calculate shortest route.



## 4. DATA LINK LAYER (2)

Functions:

- Error detection
- Flow control
- Addressing

### 4.1. ETHERNET

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.

| <i>Logical Link Control (LLC)</i>   | <i>Media Access Control (MAC)</i>   |
|---|---|
| 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. |

#### 4.1.1. 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.

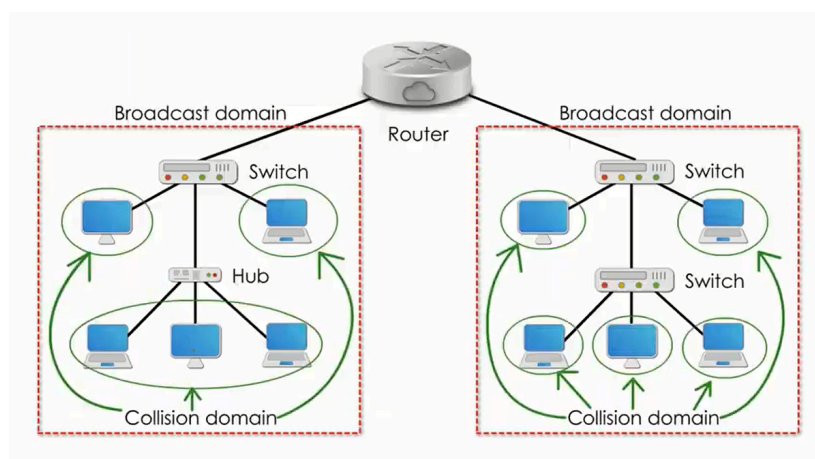
- **Carrier Sense**: Listen to the medium
- **Multiple Access**: Sending if medium is free, else waiting for a random time and try again
- **Collision**: The amplitude of the signal increases because a collision occurs.
- **Collision Detection / Backoff algorithm**: The nodes stop transmitting for a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

##### 4.1.1.1. 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.

##### 4.1.1.2. Collision domain vs broadcast domain



#### 4.1.2. Half-duplex vs Full-duplex

Full-duplex 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. Half-duplex needs CSMA/CD for collision detection.

#### 4.1.3. Ethernet II Frame

| ← 64B (1518 Bytes) → |       |         |  |        |
|----------------------|-------|---------|--|--------|
| DA 6B                | SA 6B | Type 2B | DATA (MAC SDU) 0 (+64 padding) ... 1500B | FCS 4B |

| Term | Definition                       |
|------|----------------------------------|
| DA   | Destination Address              |
| SA   | Source Address                   |
| Type | EtherType protocol, eg. IPv4/ARP |
| FCS  | Frame Check Sequence             |

Most common type in use today. Also called the DIX frame.

MAC PDU must be at least 64B to guarantee that all collisions can be detected. If it's smaller, the frame must be filled with Padding Bytes.

#### 4.1.4. IEEE 802.3 Frame

| ← 64B (1518 Bytes) → |       |           |           |         |        |
|----------------------|-------|-----------|-----------|---------|--------|
| DA 6B                | SA 6B | Length 2B | LLC 802.2 | LLC SDU | FCS 4B |

#### 4.1.5. Frame Check Sequence (FCS)

- 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.

#### 4.2. ERROR DETECTION

| Term                          | Definition   |
|-------------------------------|--|
| EDC (Error Detection Code)    | A generic term for various methods used to identify errors in transmitted data. Various data communication protocols |
| CRC (Cyclic Redundancy Check) | A specific type of EDC that uses polynomial division to detect changes to raw data.                                  |

#### 4.3. MAC-ADDRESS

| ← 48 bit →                               |              |
|--|--------------|
| Organizationally Unique Identifier (OUI) | NIC specific |

| Field   | Definition   |
|---|--|
| Organizationally Unique Identifier (OUI) (24 bit) | It is assigned by the Institute of Electrical and Electronics Engineers (IEEE) to specific manufacturers or organizations. The OUI uniquely identifies the organization that produced the network interface. |

| Field                 | Definition  |
|-----------------------|---|
| NIC specific (24 bit) | This portion ensures that each device produced by the same organization has a unique MAC address. |

7th bit: Globally unique (0) or locally administered (1)

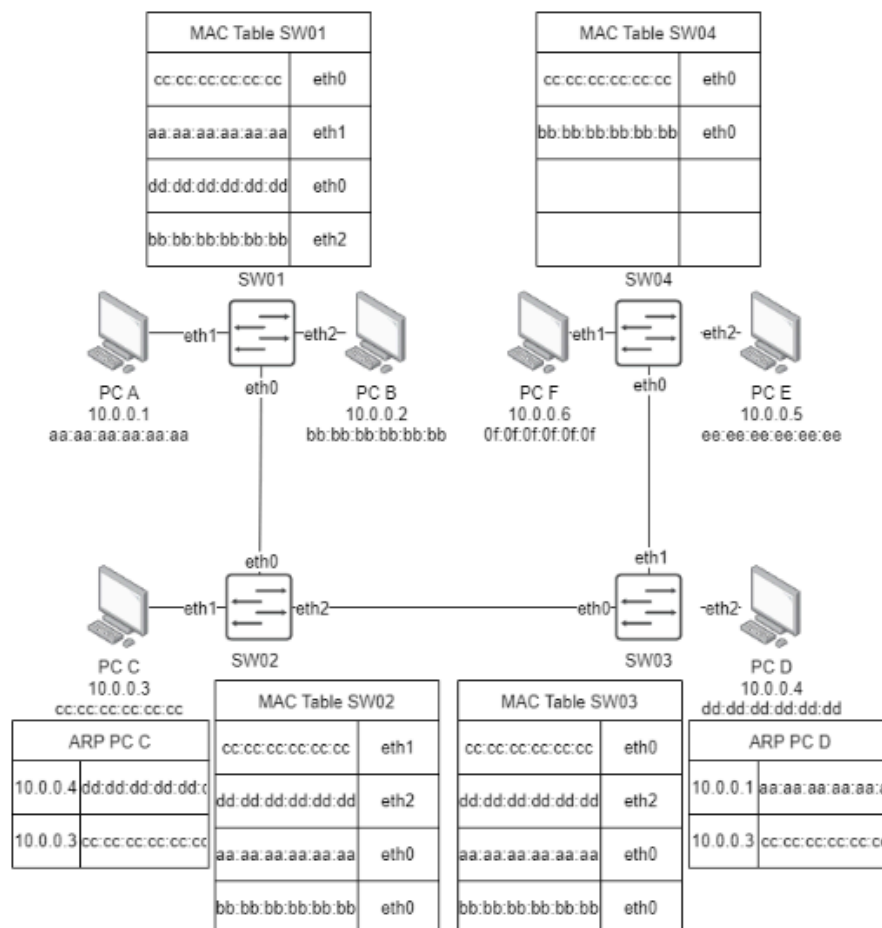
8th bit: Unicast (0) or multicast (1)

#### 4.4. ADDRESS RESOLUTION PROTOCOL (ARP)

Maps network addresses to data link layer addresses / resolves IPv4 addresses to MAC addresses. Entries in the ARP table are time stamped and can time out.

Entries are added by monitoring the traffic and adding source IP and MAC addresses of the incoming packets to the table. If no entry is found inside of the ARP table, then the node launches an ARP discovery process by sending an ARP broadcast request and receiving an ARP reply from the requested MAC addresses' host. When a node receives a packet with a destination IP address where no cached entry for the MAC address can be found, the encapsulation of the IPv4 packet fails and the packet gets dropped.

IPv6 does not need ARP because it uses the Neighbor Discovery Protocol (NDP).



- 1) PC C sends a packet to 10.0.0.4, PC D responds.
  - SW02 sends broadcast
  - all switches receive C's address
  - SW02 and SW03 receive D's address from the reply
- 2) PC A sends a frame to dd:dd:dd:dd:dd:dd, PC D responds.
  - SW01 sends broadcast
  - SW02 & SW03 already know D's address
  - SW01-SW03 receive A's address
- 3) PC B sends an ICMP ping to 10.0.0.4
  - PC B sends ARP request to find D's MAC address
  - all switches receive B's address

##### 4.4.1. Discovery

- 1) PC A sends a broadcast: "Who has the IP 10.10.10.30?"
- 2) The ARP Request is flooded
- 3) 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.
- 4) Now the PC A knows the MAC address of 10.10.10.30 and can send its Packet.

#### 4.4.2. Spoofing

ARP has no validation if the sender of a frame is correct. ARP spoofing, also called ARP poisoning, refers to the method of inserting the wrong MAC address into ARP requests and responses by the node. An attacker can lead sent frames to the wrong destination and has the ability to read the traffic (MITM attack). Configuring static ARP entries is one way to prevent ARP spoofing.

#### 4.5. SWITCH

- All devices connected to the switch ports form a **broadcast domain**
- All ports are full-duplex

##### 4.5.1. Flooding

When a switch 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)

##### 4.5.2. Filtering

When a switch 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 switches do not need to know. This reduces traffic.

##### 4.5.3. Forwarding

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

##### 4.5.4. EtherChannel = Link Aggregation Group (LAG)

EtherChannel is a technology used in networking to group several physical Ethernet links into a single logical link. This approach increases bandwidth and provides redundancy.

##### 4.5.5. Link Aggregation Control Protocol (LACP)

IEEE specification (802.3ad) open-standard protocol for EtherChannel Configurations. It dynamically adds and manages ports in the EtherChannel.

###### 4.5.5.1. Load balancing

The hashes created from source or destination IP addresses determine which link in the EtherChannel will carry the traffic.

| Term   | Definition  |
|--------|---|
| src-ip | Use when: Many devices with different IPs send to one device with a single IP address |
| dst-ip | Use when: A device with a single IP sends to many devices with different IP addresses |

#### 4.6. VLAN

LAN: all devices in the same broadcast domain

VLAN: Virtual separation of LAN on a switch

Reasons for using VLANs:

- To reduce CPU overhead on each device by reducing the number of devices that receive each broadcast frame
- To reduce security risks by reducing the number of hosts that receive copies of frames that the switches flood (broadcasts, multicasts, and unknown unicasts)
- To improve security for hosts that send sensitive data by keeping those hosts on a separate VLAN
- To create more flexible designs that group users by department, or by groups that work together, instead of by physical location
- To solve problems more quickly, because the failure domain for many problems is the same set of devices as those in the same broadcast domain
- To reduce the workload for the Spanning Tree Protocol (STP) by limiting a VLAN to a single access switch

#### 4.6.1. Trunking

With **trunking**, only a single cable is needed to carry traffic for all VLANs. VLAN trunking works by applying **VLAN tagging**, where the sending switch adds an extra header to each frame before sending it across the trunk link. This trunking header contains a VLAN Identifier (VLAN ID), allowing the receiving switch to determine the VLAN to which each frame belongs. Switch ports that are assigned to a single VLAN and carry traffic for only that VLAN are referred to as **access ports**. Ports that carry traffic for multiple VLANs using VLAN tagging are called **trunk ports**.

#### 4.6.2. 802.1Q

The standard of how to tag an ethernet frame in a trunk is defined in IEEE 802.1Q. 802.1Q inserts an extra 4 byte 802.1Q VLAN header into the original frame's Ethernet header.

#### 4.6.3. Inter-VLAN Routing

##### 4.6.3.1. Attaching a Router

A router can be added to a switch using multiple VLANs. The cable from the switch to the router gets configured as a trunk. The router then can simply perform its usual routing logic between the subnets. This concept is called **Router-on-a-Stick**.

##### 4.6.3.2. Using a Layer 3 Switch

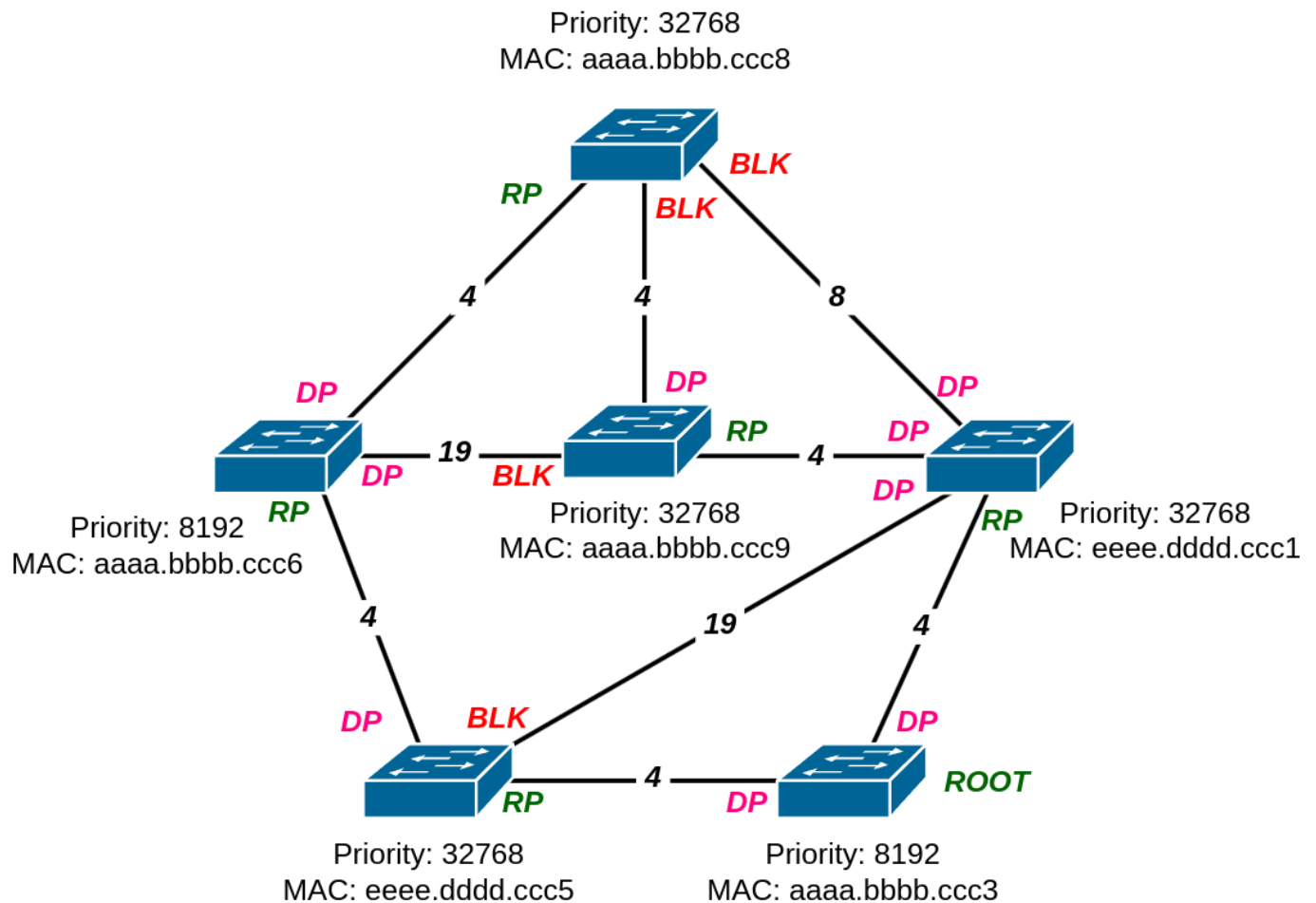
With the use of a switch with layer 3 capabilities, the need for a separate router is omitted, as the switch brings the ability for routing by itself. Routing can be turned on that switch and packets between the VLANs get routed.

#### 4.6.4. Spanning Tree Protocol (STP)

[cisco docs](#)

Prevents loops in the network (eg. broadcast).

| <b>Term</b>     | <b>Definition</b>   |
|-----------------|---|
| Root device     | Bridge on the network that serves as a central point in the spanning tree   |
| Root port       | Port on each device that provides the most efficient path to the device   |
| Designated port | Lowest path cost when forwarding packets from that LAN to the spanning-tree root  |
| Disabled port   | Port is disabled to prevent loops   |
| BPDU            | Bridge Protocol Data Unit. Destination address is multicast: 01:80:c2:00:00:00<br>Types:<br>– Hello / configuration BPDU. Sent by the root bridge<br>– Topology Change Notification (TCN). Sent by a different switch to the root |
| CAM table       | MAC address table, maps MAC addresses to ports. Entries have an aging limit   |



#### 4.6.4.1. Procedure

When the bridges in a network are powered up, each bridge functions as the STP root. The bridges send configuration BPDUs and compute the spanning-tree topology.

When a bridge receives a configuration BPDU that contains superior information (lower bridge ID, lower path cost, and so forth), it stores the information for that port. If this BPDU is received on the root port of the bridge, the bridge also forwards it with an updated message to all attached LANs for which it is the designated bridge.

If a bridge receives a configuration BPDU that contains inferior information to that currently stored for that port, it discards the BPDU.

#### 4.6.4.2. Determining bridge priority

- 1) Lowest root bridge ID (BID) – Determines the root bridge.
- 2) Lowest cost to the root bridge – Favors the upstream switch with the least cost to root
- 3) Lowest sender bridge ID – Serves as a tiebreaker if multiple upstream switches have equal cost to root
- 4) Lowest sender port ID – Serves as a tiebreaker if a switch has multiple (non-EtherChannel) links to a single upstream switch, where:
  - Bridge ID = priority (4 bits) + locally assigned system ID extension (12 bits) + ID [MAC address] (48 bits); the default bridge priority is 32,768, and
  - Port ID = priority (4 bits) + ID (Interface number) (12 bits); the default port priority is 128.

#### 4.6.4.3. Manual calculation steps

- 1) Identify the Root Bridge with the lowest BID (only one per network)
- 2) Identify link costs (per link)
- 3) Select root ports (1 per switch) with the lowest total cost to the Root Bridge
- 4) Select designated ports (1 per link)

5) Identify blocked ports (1 per redundant link)

#### 4.6.4.4. Port states

| State      | Definition   |
|------------|--|
| Disabled   | Administratively disabled for various reasons. Does not participate in STP/PVST operation.   |
| Blocking   | After excluding disabled ports, the switch starts all ports in the blocking state. In this state, the port does not accept user frames. It accepts only BPDUs.   |
| Listening  | The first transitional state after the blocking state, in which the spanning tree determines that the interface should participate in frame forwarding   |
| Learning   | In this state, the switch builds the CAM table entries. The port accepts user frames but does not forward them. From the incoming frames, it learns the MAC addresses of the connected devices. It saves the learned MAC addresses in the CAM table. |
| Forwarding | Accepts and forwards user frames.  |

#### 4.6.4.5. Timers

| Timer               | Definition   |
|---------------------|--|
| hello interval (2s) | The interval at which a bridge sends out configuration BPDUs.  |
| forward delay (15s) | The time a port remains in the Listening and Learning states before transitioning to the Forwarding state. |
| max age (20s)       | The maximum age of a received BPDU before it is considered stale.  |

#### 4.6.4.6. Topology change

- 1) Link goes down
- 2) Switch with changed link will send a BPDU of type **Topology Change Notification** (TCN) on its root port
- 3) Next switch in the hierarchy forwards the TCN to its root port and sends a configuration BPDU with the **Topology Change Acknowledgement** (TCA) flag set back to the previous bridge
- 4) As soon as root bridge receives TCN it sends a configuration BPDU with the TCA and **Topology Change** (TC) flags set
- 5) Root bridge continues to set TC flag on its configuration BPDUs for a duration of Max Age + Forward Delay (35s) and the other bridges forward them
- 6) As soon as a switch receives a BPDU with TC set, it shortens its MAC address aging timer to Forward Delay (15s)

#### 4.6.5. Rapid Spanning Tree Protocol (RSTP)

RSTP provides significantly faster spanning tree convergence after a topology change, introducing new convergence behaviors and bridge port roles to accomplish this. While STP can take 30 to 50 seconds to respond to a topology change, RSTP is typically able to respond to changes within 3 \* hello times (default: 3 \* 2 seconds) or within a few milliseconds of a physical link failure.

#### 4.7. WIRELESS

Uses different MAC address.

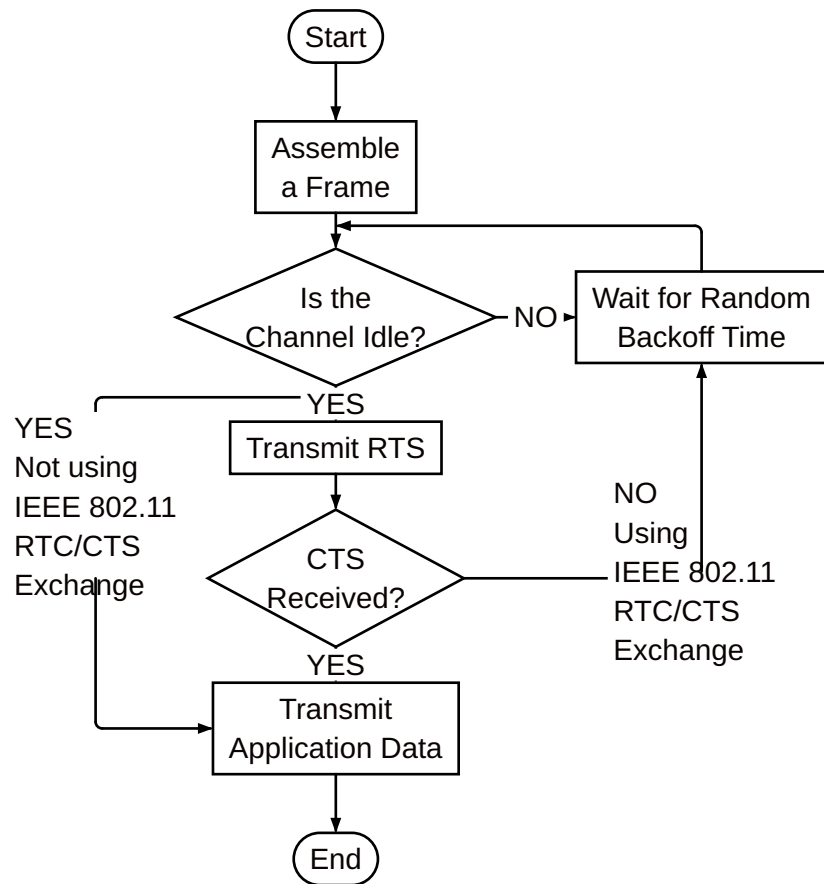
| Term         | Definition                             |
|--------------|--|
| BSSID        | Every AP has a unique BSSID            |
| ESSID / SSID | Every WLAN has an ESSID. Isn't unique. |

#### 4.7.1. Channel bonding

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



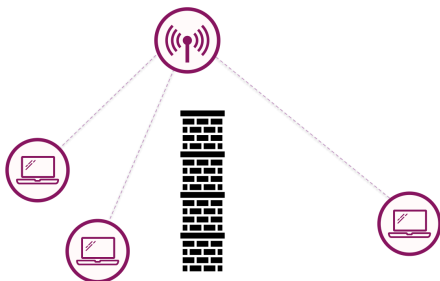
#### 4.7.2. Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA)



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

##### 4.7.2.1. 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.

##### 4.7.2.2. 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.

##### 4.7.2.3. Network Allocation Vector (NAV)

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

#### 4.7.3. Frame

| ← 34-2346B →    |                       |              |              |              |                |              |      |          |
|-----------------|-----------------------|--------------|--------------|--------------|----------------|--------------|------|----------|
| Frame ctrl (2B) | Dura-<br>tion ID (2B) | Addr. 1 (6B) | Addr. 2 (6B) | Addr. 3 (6B) | Seq. ctrl (2B) | Addr. 4 (6B) | Data | CRC (4B) |

##### 4.7.3.1. Frame Control

| ← 2B (16b) → |
|--------------|
|--------------|



| Pro-<br>tocol<br>ver.<br>(2b) | Type<br>(2b) | Sub-<br>Type<br>(4b) | To DS<br>(1b) | From<br>DS<br>(1b) | More<br>Frag<br>(1b) | Retry<br>(1b) | Pwr<br>Mgt<br>(1b) | More<br>Data<br>(1b) | WEP<br>(1b) | Rsvd<br>(1b) |
|-------------------------------|--------------|----------------------|---------------|--------------------|----------------------|---------------|--------------------|----------------------|-------------|--------------|
|-------------------------------|--------------|----------------------|---------------|--------------------|----------------------|---------------|--------------------|----------------------|-------------|--------------|

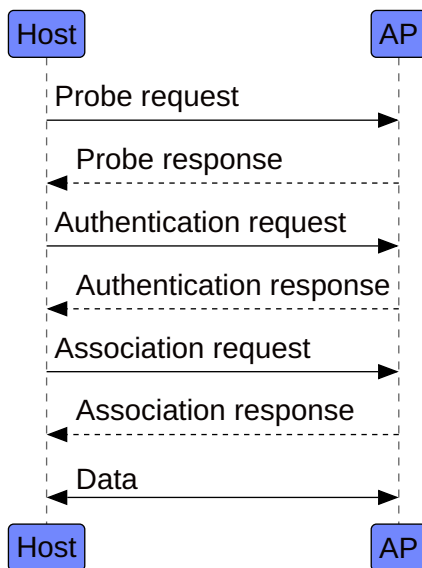
| <i>Term</i> | <i>Definition</i>  |
|-------------|--|
| To DS       | Whether the frame is being sent to the Distribution System |
| From DS     | Whether the frame originated from the Distribution System  |

#### 4.7.4. Management features

Frame types:

| <i>Term</i>                     | <i>Definition</i>   |
|---------------------------------|---|
| Probe Request                   | Frame sent by a client to discover available networks by querying nearby APs for their information.   |
| Probe Response                  | Frame sent by an AP in reply to a Probe Request, providing details about the AP, including its network name (SSID) and capabilities.                        |
| Authentication Request/Response | Frame used in the initial setup process, where a client requests authentication from the AP before being allowed to access the network.                     |
| Association Request             | Frame sent by a client to a wireless access point (AP) requesting to join a specific network.   |
| Association Response            | Frame sent by the AP in response to the Association Request, indicating whether the association was successful and providing parameters for the connection. |
| Reassociation Request           | Frame sent when a client moves from one AP to another within the same network, requesting to re-establish a connection.                                     |
| Reassociation Response          | Frame sent by the new AP in response to the Reassociation Request, confirming the re-establishment of the connection.                                       |
| Timing Advertisement            | Frame used in power-saving modes to inform clients about the timing of beacon frames, enabling better synchronization and energy efficiency.                |
| Beacon                          | Periodic frame broadcasted by an AP that provides information about the network, including the SSID, supported data rates, and security protocols.          |
| Disassociation                  | Frame sent by either the client or the AP to terminate an association, indicating that the client is leaving the network or the connection is lost.         |
| Deauthentication                | Frame used to terminate the authentication between the client and the AP, often when the client disconnects or is forcibly removed from the network.        |

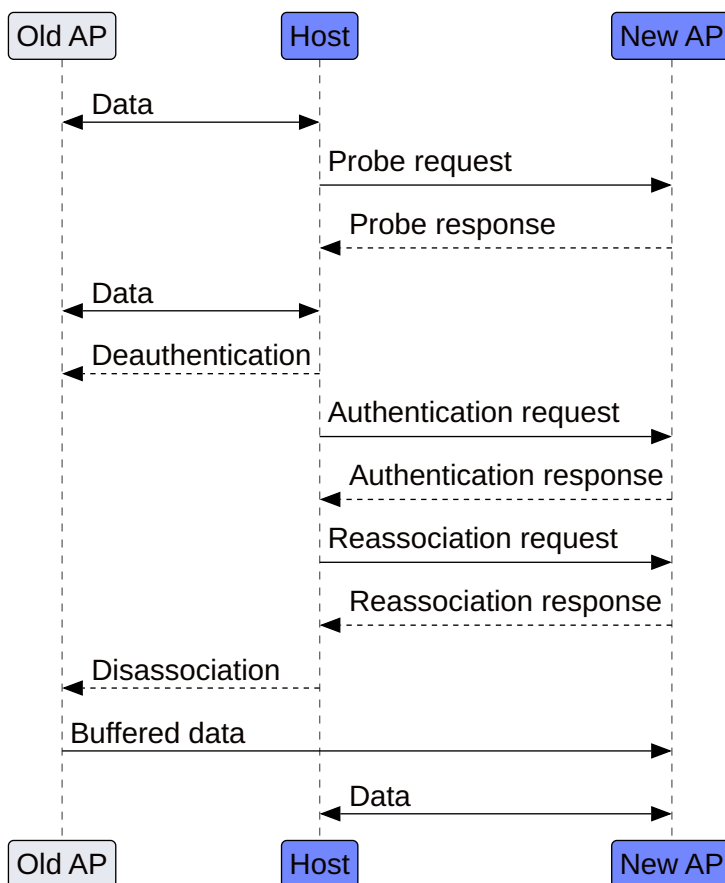
#### 4.7.4.1. Association / Reassociation



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

#### 4.7.5. 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) Either old AP or client authenticate to new AP
- 5) Client sends a reassociation request to the new AP
- 6) New AP sends a reassociation response
- 7) Client sends a disassociation request to the old AP
- 8) 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).

#### 4.7.5.1. Handoff Thresholds

Parameters that determine when a device should switch from one access point to another during movement. These might include:

| Term                             | Definition  |
|----------------------------------|---|
| Signal Strength Threshold        | Sets a minimum signal level below which the device should consider roaming to another AP.                     |
| Quality of Service (QoS) Metrics | Based on the performance characteristics, like latency or packet loss, thresholds can dictate when to switch. |

## 5. PHYSICAL LAYER (1)

### Responsibilities

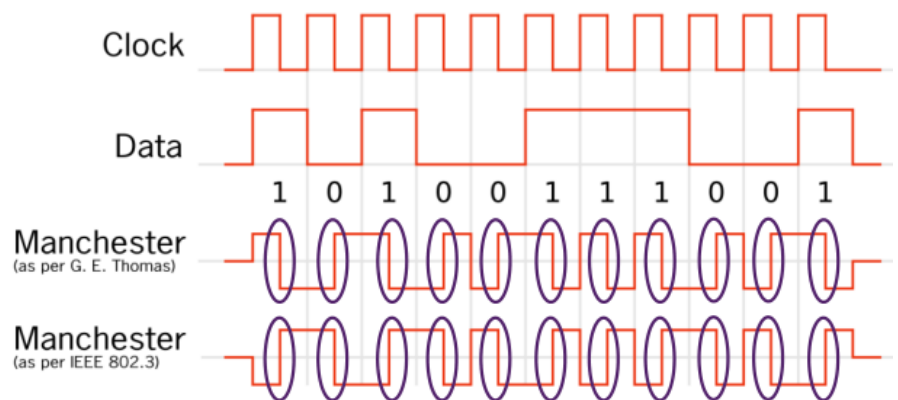
- Representing bits as physical signals (electrical voltage, light pulses, radio waves)
- Defining cables, connectors, modulation methods, and wireless frequencies
- Synchronization of transmitter and receiver
- Data rates and physical medium characteristics

### 5.1. ENCODINGS

Encoding converts the stream of bits into a format recognizable by the next device in the network path.

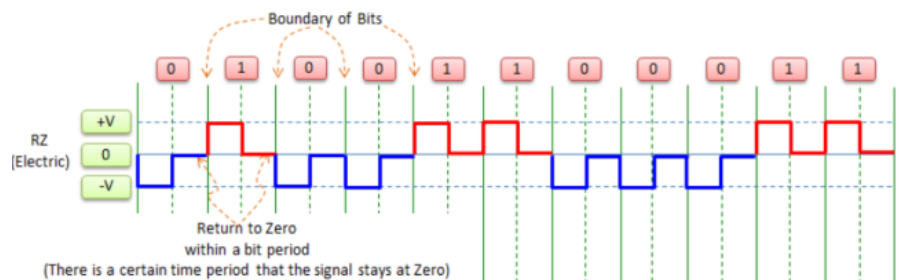
#### 5.1.1. Manchester encoding

- Self-clocking
- 1 = Falling
- 0 = Rising
- Requires twice as much bandwidth as binary encoding



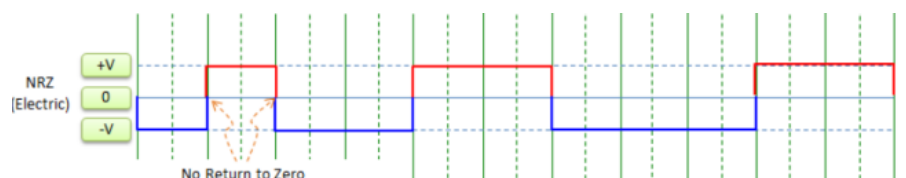
#### 5.1.2. RZ (Return-to-Zero)

- Self-clocking
- Refinement of NRZ



#### 5.1.3. NRZ (Non-Return-to-Zero)

- 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 (relevant for NRZ).

### 5.2. POWER AND DB

| Term | Definition           |
|------|----------------------|
| dB   | decibel              |
| dBm  | decibel ratio to 1mW |

| Term                 | Definition   |
|----------------------|--|
| dBi                  | antenna gain compared to isotropic radiator            |
| RSSI                 | Received signal strength indication                    |
| SNR                  | Signal to Noise Ratio                                  |
| Receiver Sensitivity | up to which level signals can be received successfully |

$$\text{decibels} = 10 \cdot \log_{10}(\text{milliwatts})$$

$$\text{milliwatts} = 10^{\frac{\text{decibels}}{10}}$$

### 5.2.1. Law of 3s

- A value of 3 dB means that the power value of interest is double the reference value
- A value of –3 dB means the power value of interest is half the reference

### 5.2.2. Law of 10s

- A value of 10 dB means that the power value of interest is 10 times the reference value
- A value of –10 dB means the power value of interest is 1/10 of the reference

### 5.3. MODULATION

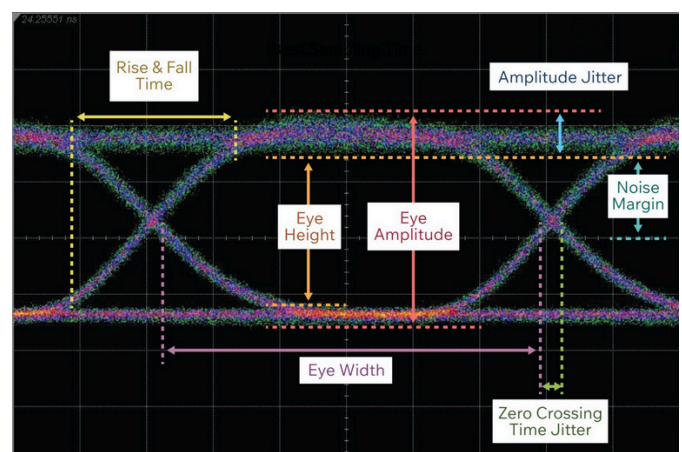
Altering the carrier signal.

### 5.4. FIBER MEDIA

| Single-Mode   | Multimode                         |
|---|-----------------------------------|
| Very small core   | Larger core                       |
| Expensive lasers  | Less expensive LEDs               |
| Long-distance applications (⇒ more susceptible to chromatic dispersion) | up to 10Gbps over 500 meters      |
|   | LEDs transmit at different angles |

### 5.4.1. Eye Diagram

- An eye diagram results from superimposing the “0”s and “1”s of a high-speed digital data stream.
- An eye diagram shows a relative performance of the signal
- 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



### 5.4.2. Attenuation

- Absorption by the fiber material
- Scattering of the light from the fiber

| Term       | Definition   |
|------------|--|
| Microbends | Caused by small distortions of the fiber in manufacturing                |
| Macrobends | Caused by wrapping fiber around a corner with too small a bending radius |

| Term                   | Definition   |
|------------------------|--|
| Back reflections       | Caused by reflections at fiber ends, like connectors |
| Fiber splices          | Caused by poor alignment or dirt                     |
| Mechanical connections | Physical gaps between fibers                         |

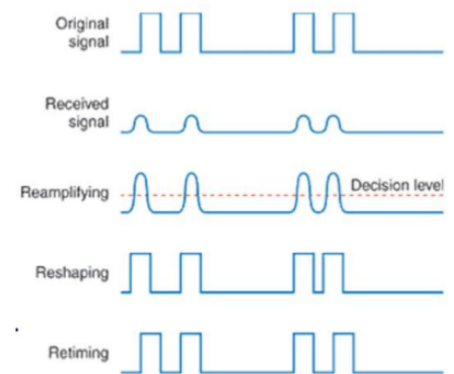
#### 5.4.3. Dispersion

| Term                               | Definition  |
|------------------------------------|---|
| Chromatic Dispersion               | <ul style="list-style-type: none"> <li>– Different wavelengths travel at different speeds</li> <li>– Causes spreading of the light pulse</li> </ul>   |
| Polarization Mode Dispersion (PMD) | <ul style="list-style-type: none"> <li>– Single-mode fiber supports two polarization states</li> <li>– Fast and slow axes have different group velocities</li> <li>– Causes spreading of the light pulse</li> </ul> |

#### 5.4.4. Regeneration

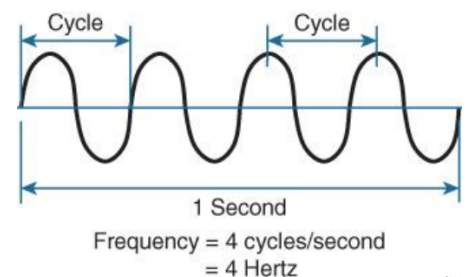
Fixes the dispersion.

| Term          | Definition   |
|---------------|--|
| Re-amplifying | Makes the analog signal stronger (i.e. makes the light brighter)   |
| Reshaping     | Restores the original pulse shape that is used to distinguish 1's and 0's.   |
| Retiming      | Restores the original timing between the pulses. Usually involves an Optical-Electric- Optical (O-E-O) conversion. |



#### 5.5. FREQUENCY

| Term       | Definition   |
|------------|--|
| Hertz (Hz) | Number of cycles per second  |
| Bandwidth  | Width of frequency space required within the band  |
| Wavelength | Measure of the physical distance that a wave travels over on cycle. Increases as the frequency decreases |



#### 5.6. CALCULATIONS

##### 5.6.1. 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:

$$T(s) = \frac{\text{Length of cable (km)}}{\text{Speed of signal (km/s)}}$$

##### 5.6.2. Optical budget

Transmission power - Receiver sensitivity

---

## 6. CISCO

### 6.1. ROUTER SETUP

```
Router> enable
Router# configure terminal
Router(config)#
```

### 6.2. INTERFACES

#### 6.2.1. Static IP Assignment

```
Router(config)# interface GigabitEthernet 0/0/1
Router(config-if)# ip address 172.16.0.0 255.255.255.252
Router(config-if)# no shutdown
Router(config-if)# exit
```

#### 6.2.2. DHCP Assignment

```
Router(config)# interface GigabitEthernet 0/1/1
Router(config-if)# ip address dhcp
Router(config-if)# no shutdown
Router(config-if)# exit
```

#### 6.2.3. Show

```
Router(config)# do show ip interface brief
Router# show ip interface brief
Router# show ip interface GigabitEthernet 0/0/1
```

### 6.3. VLAN

```
Switch(config)# vlan 120
Switch(config-if)# name vlan-server
Switch(config-if)# exit
```

#### 6.3.1. Assign IP

```
Switch(config)# interface vlan 120
Switch(config-if)# ip address 10.120.0.10 255.255.255.0
```

#### 6.3.2. Access Port

```
Switch(config)# interface GigabitEthernet 0/0/1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 120
Switch(config-if)# exit
Switch(config)# interface GigabitEthernet 0/0/1-5
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 120
Switch(config-if)# exit
```

#### 6.3.3. Access Port

```
Switch(config-if)# switchport mode trunk
```

### 6.3.4. VTP (Virtual Trunk Protocol)

#### 6.3.4.1. Server

```
Switch(config)# vtp domain ins  
Switch(config)# vtp mode server
```

#### 6.3.4.2. Client

```
Switch(config)# vtp domain ins  
Switch(config)# vtp mode client
```

### 6.3.5. LACP (Link Aggregation Control Protocol)

```
Switch(config-if)# channel-group 5 mode active  
Switch(config-if)# channel-group 5 mode passive
```

### 6.3.6. Load Balancing

```
Switch(config)# port-channel load-balance <strategy>
```

### 6.3.7. STP (Spanning Tree Protocol)

#### 6.3.7.1. Bridge priority

```
Switch(config)# spanning-tree vlan 1 priority <priority>
```

#### 6.3.7.2. Interface costs

```
Switch(config-if)# spanning-tree cost 100
```

#### 6.3.7.3. PortFast mode

```
Switch(config-if)# spanning-tree portfast
```

#### 6.3.7.4. Show

```
Switch# show spanning-tree  
Switch# show spanning-tree root
```

## 6.4. ROUTING

### 6.4.1. Static

#### 6.4.1.1. IPv4

```
Router(config-if)# ip route <destination_network_id> <subnet_mask>  
<next_hop_router> <administrative_distance>?  
Router(config-if)# ip route 10.0.0.0 255.0.0.0 192.168.1.1
```

#### 6.4.1.2. IPv6

```
Router(config-if)# ip route <ipv6_prefix> <outgoing_interface> <next-  
hop> <administrative_distance>?  
Router(config-if)# ipv6 route 2001:db8:2103:a::/64  
GigabitEthernet1/0/1 fe80::ba27:ebff:fea8:3e50
```

### 6.4.2. OSPF

#### 6.4.2.1. IPv4

```
Router(config)# router ospf <process-id>  
Router(config-if)# ip ospf <process-id> area <area-nr>
```

#### 6.4.2.2. IPv6

```
Router(config)# ipv6 router ospf <process-id>  
Router(config-if)# ipv6 ospf <process-id> area <area-nr>
```

### 6.4.3. Show

Router# show ip route

Router# show ip ospf route

## 6.5. DHCP

### 6.5.1. Create Pool

```
Router#(config-if) ip dhcp pool DEV
network 192.168.1.0 255.255.255.0
default-router 192.168.1.1
dns-server 1.1.1.1 8.8.8.8
lease 5
domain-name enterprise.com
```

### 6.5.2. Relay Agent

```
Router(config-if)# ip helper-address 176.16.12.10
```

## 6.6. NAT

### 6.6.1. IF Inside

```
Router(config-if)# ip nat inside
```

### 6.6.2. IF Outside

```
Router(config-if)# ip nat outside
```

### 6.6.3. ACL (Access Control List)

```
Router(config-if)# access-list 1 permit 192.168.1.0 0.0.0.255
```

### 6.6.4. PAT (Nat overload)

```
Router(config-if)# ip nat inside source list 1 interface
GigabitEthernet0/1 overload
```

## 6.7. IPV6

```
Router(config-if)# ipv6 enable
```

### 6.7.1. DHCPv6

```
Router(config-if)# ipv6 dhcp client pd MY_PREFIX
```

```
Router(config-if)# ipv6 address autoconfig default
```

## 6.8. TROUBLESHOOTING

### 6.8.1. Ping

```
Router# ping <destination-ip> source <interface-name>
```

### 6.8.2. Traceroute

```
Router# traceroute <destination-ip> source <interface-name> numeric
```