Simon Vetter og Hermann Elton

Nettverksprogrammering

1 Computer networks and the internet

1.1 What's the internet?

The internet is a computer network that interconnects billions of computing devices throughout the world. Not too long ago, these computing devices were primarily traditional desktop pcs, Linux workstations, and so-called servers that store and transmit information such as Web pages and email messages. Due to the increase in laptop smartphones and tables as internett things, the word computer network is a bit dated. It's basically a lot of so called computers that are connected to each other, and share data with each other.

1.2 The Network

Protocol

Protocol defines the format and the order of messages exchanged between two or more communicating entities, as well as the actions taken on the transmission and/or receipt of a message or other event. A protocol is basically a conversation or exchange between systems, units, etc. Something is sent and something is received and something is returned. A protocol is a standard set of rules that allow electronic devices to communicate with each other. These rules include what type of data may be transmitted, what commands are used to send and receive data, and how data transfers are confirmed.

You can think of a protocol as a spoken language. Each language has its own rules and vocabulary. If two people share the same language, they can communicate effectively. Similarly, if two hardware devices support the same protocol, they can communicate with each other, regardless of the manufacturer or type of device. For example, an Apple iPhone can send an email to an Android device using a standard mail protocol. A Windows-based PC can load a webpage from a Unix-based web server using a standard web protocol.

Packet switching

Packet switching transmits data across digital networks by breaking it down into packets for more efficient transfer using various network devices. When one device sends a file to another, the file is broken down into packets that are sent across the network using the most efficient route. The network devices route the packets for the given end system. There are two predominant types of packet switching routers and link-layer switches. Both types of

switches forward packets toward their ultimate destinations. Link-layer switches are typically used in access networks, while routers are typically used in the network core. Packet switching also grants security features considering, if one network node were to be unavailable/destroyed, the network devices could simply redirect the packets using another path.

The sequence of communication links and packet switches traversed by a packet from the sending end system to the receiving end system is known as a route or path through the network

Router switching requires some additional information, addresses for receiver and sender and a checksum to control error detection.

Link-layered switching however is a dedicated line between two different end systems, and is not shared with anyone else.

Most packet switches use store-and-forward transmission at the inputs to the links. Store-and-forward transmission means that the packet switch must receive the entire packet before it can begin to transmit the first bit of the packet onto the outbound link.

Types of Delay

Processing Delay

The time required to examine the packet's header and determine where to direct the packet is part of the processing delay. The processing delay can also include other factors such as the time needed to check for bit-level errors in the packet that occurred in transmitting the packet bits from the upstream node to router A. Processing delays in high speed routers are typically on the order of microseconds or less.

Queuing Delay

Queuing delay is the delay that is experienced when waiting to be transmitted to a link. The length of the queuing delay of a specific packet will depend on the number of earlier-arriving packets that are queued and waiting for transmission onto the link.

Transmission Delay

Assuming that packets are transmitted in a first come first server manner as is common in packet switched networks, our packet can be transmitted only after all the packets that have arrived before it have been transmitted. This can cause a transition delay if you can't receive all the packets needed fast enough.

Propagation Delay

The time required to propagate from the beginning of the link to router B is the propagation delay and relies on the medium that is being used.

1.3 Layered Architecture

To provide structure to the design of network protocols, network designers organize protocols. Each protocol belongs to one of the layers. The protocols of the various layers are called the protocol stack. Distributed applications have an interface to the communication stack, and the top layer in the simplified 5-layer OSI model is the application layer. System developers are responsible for developing the applications according to interfaces to protocols on the application layer. The protocols, on the other hand, are the responsibility of various standardization organizations to define.The main principle is that applications and protocols communicate with each other on an equal level, but use services from the team below to transfer data (PDU, Protocol Data Unit) between the parties. Because there are defined interfaces between the layers, new protocols can be developed against these without the "whole stack" having to be rtbeplaced. For example, you can switch between wireless and wired data transfer without affecting the network layer.

The Internet protocol stack consists of five layers.

Task for the different layers are:

Application layer: Interface with distributed applications.

Transport layer: End-to-end transmission of messages.

Network layer: Ensures that each packet is routed through the network

Link layer: Ensures that packets are transferred between two adjacent nodes (between two network cards)

Physical layer: Sends signals over a transmission medium (air, copper, fiber)

Protocol Layering

A protocol layer can be implemented in software, in hardware, or in a combination of the two. App layer protocols such as HTTP and SMTP are almost always implemented in software in the end systems. So are transport-layer protocols. Because the physical layer and data link layers are responsible for handling communication over a specific link, they are typically implemented in a network interface card associated with a given link. The network layer is often a mixed implementation of hardware and software. Protocol layering has conceptual and structural advantages. Layering provides a structured way to discuss system components. Modularity makes it easier to update systems components. Some researchers and networking engineers are vehemently opposed to layering. One potential drawback of layering is that one layer may duplicate lower layer functionality. For example, many protocol stacks provide error recovery on both a per-link basis and an end to end basis. A second potential drawback is that functionality at one layer may need info that is only present in another layer, this violates the goal of separation of layers.

Application layer

The application layer is where network applications and their application-layer protocols reside (The interface towards the distributed application). The Internet's application layer includes many protocols, such as the HTTP protocol (which provides for Web document request and transfer), SMTP (which provides for the transfer of e-mail messages), and FTP (which provides for the transfer of files between two end systems).

An application-layer protocol is distributed over multiple end systems, with the application in one end system using the protocol to exchange packets of information with the application in another end system. A packet of information at the application layer is often called a message.

Transport layer

The Internet's transport layer transports application-layer messages between application endpoints. In the Internet there are two transport protocols, TCP and UDP, either of which can transport application-layer messages. TCP provides a connection-oriented service to its applications. TCP includes guaranteed delivery of the messages to the destination and flow control (breaks messages into shorter segments, source can throttle transmission rate when the network is congested). The UDP protocol provides a connectionless service to its applications, can think as only dumping messages, not concerning if destination receives or not. A transport-layer packet is often called a *segment.*

Network layer

The Internet's network layer is responsible for moving network-layer packets known as *datagrams* from one end system to another. TCP or UDP (from the transport-layer protocol) in the source end system, passes the segment alongside a destination address to the network layer. The Internet's network protocol also includes the IP Protocol which defines the fields in the datagram as well as how end systems and routers should act. Lastly, this layer's protocol also includes routing protocols that determine the routes the datagrams take (the path).

Link layer

The link layer ensures that packets transfer between adjacent network elements (network cards), a series of routers between the source and destination. When a packet moves from one node to the next node in the route, the network layer passes the datagram down to the link layer, which delivers the datagram to the next node along the route. At this next node, the link layer passes the datagram back up to the network layer (The network and link layer work intermittently together). Examples of link-layered protocol include Ethernet and WiFi. Link-layer packets are often referred to as frames.

Physical layer

The job of the physical layer is to move individual bits within the frame from one node to another, through a transmission medium. The medium may be guided (copper wire, fiberoptics) or un-guided (satellite, through atmosphere).

The OSI Model

The iso model implements seven layers and not five like the IPS does. The layers are Application layer, presentation layer, session layer, transport layer, network layer, data link layer, and physical layer. Five of these layers are basically the same as in IPS, but the two new session layer and presentation layer. The presentation layer is to provide services that allow communicating applications to interpret the meaning of data exchanged. These services include data compression and data encryption as well as data description. The session layer provides for delimiting and synchronization of data exchange, including the means to build checkpointing and recovery schemes.

1.3 Encapsulation

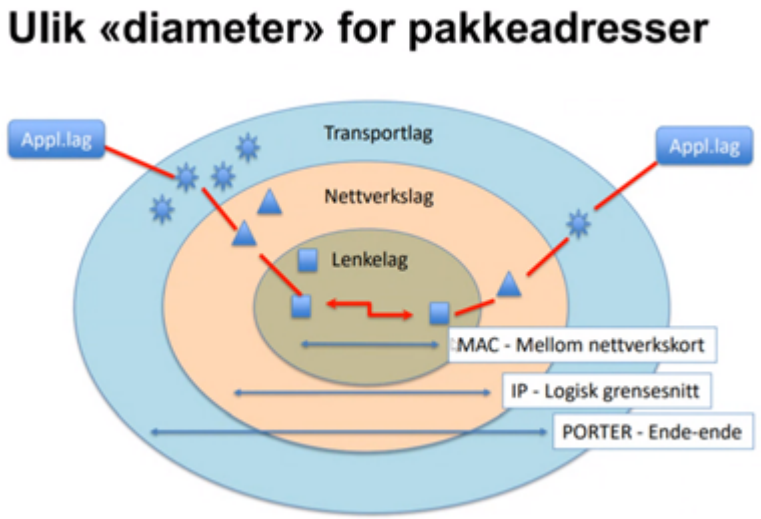
Each of the different layers' protocols appends headers to the packet or payload. These headers contain control information the next layer needs (ex. Receiver and sender of the given layer). For each layer, the size of the total packet increases for each layer. The action of the next protocols are only given from the information in the header and rules of handling these. For example, here is what happens when you send an email using your favourite email program (such as Outlook or Thunderbird):

1. the email is sent from the Application layer to the Transport layer.
2. the Transport layer encapsulates the data and adds its own header with its own information, such as which port will be used and passes the data to the Internet layer
3. the Internet layer encapsulates the received data and adds its own header, usually with information about the source and destination IP addresses. The Internet layer than passes the data to the Network Access layer
4. the Network Access layer is the only layer that adds both a header and a trailer. The data is then sent through a physical network link.

Each packet (header + encapsulated data) defined by a particular layer has a specific name:

- **Frame** – encapsulated data defined by the Network Access layer. A frame can have both a header and a trailer.
- **Packet** – encapsulated data defined by the Network layer. A header contains the source and destination IP addresses.

- **Segment** – encapsulated data as defined by the Transport layer. Information such as the source and destination ports or sequence and acknowledgment numbers are included in the header.



1.4 Encryption

A secure transfer may happen throughout multiple layers simultaneously and independently of each other. information in the different layers may be encrypted independently of each other. Examples of encryption in different layers are; Transport layer - TLS/SSL, Network layer - VPN and Link layer - WiFi through WPA2.

As modern systems for sending and receiving data contain a great deal of sensitive information, it is important that this information certainly arrives unchanged, and that only the correct recipient receives the information. Using different encryption methods will ensure this well, and using layered encryption, will ensure that as much information as possible is protected before it is forwarded. In layered encryption, the payload is encrypted in different layers that are independent of each other. If a layer fails or someone manages to decrypt it, there are still several layers of encryption protection left.

Transport Layer Security (TLS) Layer 6

Successor of Secure Sockets Layer (SSL), and are protocols that will provide secure communication in a network of computers such as the Internet. TLS is usually used in layer 6 (the presentation layer), but can also be part of layers 7 and 5 in an application that is to send out data (ie TLS is used in the application layer in a 5-layer simplified OSI model). It is often used to encrypt HTTP requests and responses without messing with the TCP protocol, but can also be used in email and other Internet services.

In short, TLS works by using encryption between server and client. If we go more in depth, we can divide the behavior of TLS into several parts:

- First, the client and server must agree to use TLS. Either by using a separate port for TLS connections (eg port 443 for HTTPS) or by the client sending a request to the server to use TLS.

The server and client perform a handshake to determine which variables to use to create a secure connection. The client presents a list (Cipher suites) with encryption algorithms and hash functions, such as SHA256.

- The server selects the most secure of the client's encryption algorithms and hash functions, which the server also has available, and notifies the client of this choice.

- Server sends back a digital certificate that contains info such as name and public encryption key

- To encrypt the communication, the client encrypts a randomly selected number with the server's public key. This is sent to the server who can decrypt it with his private key.

- Based on the randomly selected number, new keys are created for encryption and decryption.

Virtual Private Network (VPN) Layer 3

VPN is usually part of several layers in the OSI model, but much of the work takes place in the network layer (layer 3). Then the payload to be sent out has often been through TLS encryption from the presentation team already. Users of a wireless network that uses VPN must authenticate themselves, and communication takes place via an encrypted key. No unauthorized users can access what is happening wirelessly on the network, and eavesdropping is useless as the data is encrypted.

Communication between devices that use VPN seems to be connected to the same private network, which gives the name Virtual Private Network. The network receives the same benefits as private networks, including security. VPN is often used to do administrative work on a network, even if you cannot be directly connected to it. For example, a teacher may need to edit grade files on the school network from home, and may do so by connecting to the school's VPN and authenticating themselves.

Wifi Protected Access II (WPA2) Layer 2

Used on the link layer (layer 2), which means that it is used for both TLS and VPN when a computer sends out data. WPA2 is the successor to WPA and the predecessor to WPA3, which was created to correct the vulnerabilities and security holes of their predecessors. These techniques are used to allow devices to connect to a wireless network without intruders. WPA2 support is required for all newer wifi devices in order for them to carry the wifi trademark (logo) on the box / product. The WPA protocols took over for the very weak WEP security that was used until the beginning of the 2000s. WPA2 supports the use of Temporary Key Integrity Protocol (TKIP) and Counter Mode Cipher Block Chaining Message

Authentication Code Protocol (CCMP) cryptography. CCMP is more secure than TKIP and should theoretically be almost unshakable, but early implementation in WPA2 proved to have weaknesses that could be exploited.

After going through the link layer, the payload passes to the physical layer to be transferred to another computer. Then encryption with TLS, VPN and WPA2 may have been used to ensure secure transmission.

1.5 What is an IP?

The network layer: The third layer in the simplified 5-layer layer model. Ensures that each packet to be sent will be routed through the network.

Router: Forwards data packets between different IP networks through the different data networks it is connected to.

The IP network is a logical network that is part of the layer model and belongs to the network layer. The IP network is one of the most important when it comes to data communication. All data traffic and transmission of data packets takes place over IP networks and routers. An IP network is a collection of machines that have the same web address and thus have the same web masks. This also means that they have the same broadcast domain, since they are in the same network.

When machines are on the same IP network, it means that they can send IP packets to each other without going through a router to have it sent. It will be less work to send packages when it is within the same system.

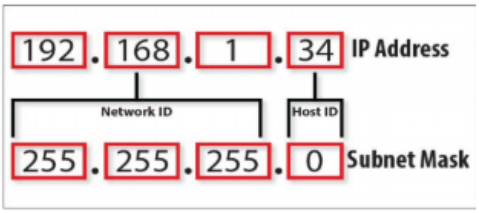
If the data is to be sent to places outside the IP network, it is routed through routers. A router has at least two inputs. Each of these inputs points to IP addresses in different IP networks and therefore becomes the link between the two IP networks. Then data will be sent from one IP network via the router and to the other IP network.

An IP address is a unique identifier or address that is assigned to a device such as a PC or printer. the address must be unique because it is used to get the package to the right machine.

A machine can have several interfaces, for example a laptop can be connected to both wirelesom har
samme domenenavn.
7. TCP
Hvordan oppnår man pålitelig overføring i TCP?
Hvordan får TCP indikasjon om metning i nettet? Hva er tiltaket?
Pålitelig overføring: opprette forbindelse 3W-HS. Sending har sekvensnummer for første byte i nyttelast. Kvitteringsnummer for neste forventede mottatte byte. Retransmisjon hvis feil i overføring
Indikasjon metning: metning betyr pakketap. Pakkekvittering uteblir eller ny kvitss and wired LAN at the same time, and each of the interfaces must have its unique IP address.

The IP address is 32 bit in version 4 (IPv4). This means that there are 2 ^ 32 possible combinations, which give about 4 billion IP addresses.
Although it sounds like a big number, IPv4 is not enough for all the computers used today.

Therefore, a new version (IPv6) has been released that uses 128 bits in the address field that gives 2 ^ 128 possible combinations.
For example, dubbed IVI is used to enable machines, at different addresses (IPv4 and IPv6), to communicate with each other and still maintain end-to-end address visibility / transparency.
There is a close and important connection between the web address and the machine's IP address. The URL is embedded as part of the IP address. The IP address consists of two parts, a web address (web ID) and a node address (host ID). These two parts cannot be read directly from the 32 bit of the IP address. It is the web mask that tells us where the distinction in the IP address goes between the web address and the node address.



Subnetting is the practice of dividing a network into two or more smaller networks. This mainly gives 3 benefits, which are:

1. Increases "routing" efficiency, by simply sending the information to those who need it, instead of sending it to everyone, which creates much more traffic.
2. Increases network security, because in a network, all devices have the ability to reach each other, which can lead to potential security issues.
3. This also allows you to group different departments together, and only provide the ability to receive the potentially confidential packages to the users who will have them.

IP-adressen er **10.24.21.58** med Nettmaske **255.255.252.0**

- **252 = 1111 1100**
- 254 = 1111 1110
- 255 = 1111 1111

252	1111 1100
21	0001 0101
OG	0001 0100 = 20

Nettadressen blir da **10.24.20.0/22**

1.6 Packet switching

Packet switching is a method of connecting receiver and transmitter to enable data exchange. This is the most widely used transmission method in connection with data traffic and local area networks. The data to be transmitted is divided into several packets, each with a tail and a head, as well as the actual data to be sent. Furthermore, these are sent over the network via so-called switches. Where the packet is to be sent, from switch to switch, is determined by a recipient's address which is placed in the packet's header together with the sender's address. During packet transfer, packet loss can occur, as well as changes to data along the way. As a security check to help against this, all packets have a checksum in the packet header that helps identify whether the data has been transferred correctly or not.

1.7 IETF

Summary

Standardization of the internet has been a process that has been going on ever since we got the internet in the 90s. Various volunteers and organizations have taken on the work in this standardization process. This has led to different organizations taking shape, and has been given different areas of responsibility. One of these organizations is the Internet Engineering Task Force (IETF). IETF develops standards for application, transport, internet and the link layer. Their main focus is TCP / IP standards. Their primary form of work is via Request for Comments (RFC). We have, for example, IPv4, which is a well-known RFC, numbered as RFC 791. There is a close collaboration between computer engineers and other professionals before these standards are published. IETF meetings have an entrance fee of around 650 US dollars (2014) and are held three times a year, but there is no formal requirement for membership.

1.7.1 Tasks

The main task of the IETF is to develop, promote and maintain Internet standards. IETF has focused on communication protocols such as TCP / IP.

The standards developed by the IETF are optional and have no "formal" recognition.
Their tasks are to ensure that technology:

- can perform necessary function.
- distributed correctly and is scalable.
- is safe and can be operated safely.
- is affordable to use

1.7.2 Working methods

The IETF is an open standardization body that does not require any formal membership or requirements for the members. The organization consists of a relatively large collection of people who are interested in the further development of the Internet. All participants and leaders in the IETF are volunteers, but as a rule their work is funded by either employers or sponsors. The IETF has eight work areas that have between 3 and 30 different work groups each.

IETF is divided into eight different work areas which are: applications, internet, operation, routing, security, real-time applications and infrastructure, transport and general. The working groups have a chosen leader and a "charter", which describes the group's focus area and what is expected of them. The working groups are expected to complete their tasks in order to be closed down or, in some cases, to receive new tasks. The area managers' responsibility is to inspect and follow up the work of the working groups within the area.

Most of the work of the various working groups is done via email, as the IETF holds meetings only three times a year. The work then takes place through discussions on an open mailing list.

The IETF works on the principle of "rough consensus", which means that consensus with most of the members is enough to make a decision, which is well emphasized through a quote from David Clark on the IETF: "We reject kings, presidents and votes. We believe in broad consensus and working code. " This means that a rough consensus between the members of the mailing list has final authority in decisions. The IETF thus has no polls or leaders who decide, but rather consensus-based decisions are made. In this way, the IETF thus bases itself in many ways on trust with its participants. Finally, we end up with new internet standards, numbered with what we call Request for Comments and one serialized number. Not all RFCs end up as internet standards, but they are published and available to everyone. An RFC has a specific number, and if it is to be updated, it is made rather than new. Then the old standard is "deprecated", or written off in favor of the new RFC.

1.8 IEEE

Summary

The Institute of Electrical and Electronics Engineers (IEEE) is a non-profit organization that aims to promote the engineering activities associated with creating, developing, integrating, sharing and applying knowledge of electrical engineering, information technology and science. The organization mainly works to form professional networks that will help to influence technological development, but they also have a number of different offers for students, such as research articles and online television. The organization has a large membership group, about 430,000 members. All members must either be affiliated with a recognized institution with an education in engineering or technology or have relevant education and experience in a relevant field. In some rarer cases, one may be nominated to receive the highest degree of membership in the organization.

IEEE is constantly working to develop, and therefore has a number of goals for the future that focus on inspiring, promoting and improving the understanding of technology and global innovation as well as offering various career opportunities and information. Subordinate to IEEE is the organization IEEE-SA. They create standards for different industries where the main focus is to create a set of common guidelines for things such as measurement units, digital interfaces and the like.

1.8.1 Standards

The Institute of Electrical and Electronics Engineers Standards Association (IEEE-SA) is an organization in the IEEE that creates global standards for many different industries, such as: IT, robotics, nanotechnology, telecommunications, etc. The 4 most well-known standards are:

- IEEE 260: Standard Symbols for Units of Measure
- IEEE 488: Standard digital interface for programmable instruments.
- IEEE 610: Standard Glossary of System Development Terminology
- IEEE 754: Floating-point arithmetic specifications

IEEE has many groups of standards. One of the most important is 802.11, which is a series of standards that deal with wireless local area networks in the 2.4, 3.6 and 5 GHz frequency bands. The most popular standard in this family is 802.11b which is based on the 802.11-1997 standard. 802.11b was the first widely accepted wireless networking standard, followed by IEEE 802.11g and 802.11n.

1.8.2 Working methods

IEEE mainly works to form professional networks and through these influence technological development. The IEEE organizes several hundred conferences annually and publishes over 30% of the articles in electronics and computer technology. Students can get help from other IEEE members to work on articles and assignments. They can also submit articles to get feedback. IEEE maintains a very high standard of journals. IEEE also offers a good research

site for students and companies in technology and engineering, where they can read up on information. IEEE also offers a good collection of research articles, magazines, online TV and a job search site.

1.8.3 Future

For the next five years, the IEEE has the following objectives:

● To promote global innovation through broad collaboration and knowledge sharing,

● To improve public understanding of engineering work and technology, as well as apply standards for their practical application.

● To be a reliable source of educational services and resources to support lifelong learning.

● Offer opportunities for career and professional development

● Inspire a worldwide audience by building societies that promote technical interests, informing about public policy and expanding the level of knowledge for the benefit of humanity.

In the long term, the IEEE wants to cultivate a collaborative environment that is open, inclusive and impartial, that will maintain the organization's strength, reach and vitality for future generations.

2 Application layer

2.1 Principles of Network Applications

Architectures

Client-server architecture

There is an always-on host, called the server which services requests for many other hosts, called clients. Note that with client server architecture, clients do not directly communicate with each other for example in web apps, two browsers do not directly communicate. There isn't always one host, because one server might not be able to handle all the load that might come. That's why larger apps have multiple servers that form one super server that the clients communicate with.

Peer to Peer

In a P2P architecture, there is minimal reliance on dedicated servers in data centers. Instead the app exploits direct communication between pairs of intermittently connected hosts called peers. The peers are not owned by the service provider, but instead desktops and laptops

controlled by users with most of the peers residing in homes, unis, and offices. Because the peers communicate without passing through a dedicated server, the architecture is called peer to peer. One of the most compelling features of P2P architectures is their self-scalability. P2P architectures are also cost effective, since they normally don't require significant server infrastructure and server bandwidth.

Transport services provided by the internet

TCP services

The TCP service model includes a connection-oriented service and a reliable data transfer service. When an application invokes TCP as Its Transport protocol the application receives both these services from TCP

Connection-oriented service

TCP has the client and server exchange transport-layer control information with each other before the application-level messages begin to flow. This so-called handshaking procedure alerts the client and server, allowing them to prepare for an onslaught of packets. After the handshaking phase, a TCP connection is said to exist between the sockets of the two processes. The connection is a full-duplex connection in that the two processes can send messages to each other over the connection at the same time. When the app finishes sending messages, it must tear down the connection.

Reliable data transfer service

The communicating processes can rely on tcp to deliver all data sent without error and in the proper order. When one side of the application passes a stream of bytes into a socket, it can count on TCP to deliver the same stream of bytes to the receiving socket, with no missing or duplicate bytes.

UDP services

UDP is a no-frills, lightweight transport protocol, providing minimal services. UDP is connectionless, so there is no handshaking before two processes start to communicate. UDP provides an unreliable data transfer service that is, when a process sends a message into a UDP socket, UDP provides no guarantee that the message will ever reach the receiving process. Furthermore, messages that do arrive at the receiving process may arrive out of order. UDP does not include a congestion-control mechanism, so the sending side of UDP can pump data into the layer below at any rate it pleases.

Application-Layer Protocols

An application-layer protocol defines:

- The types of messages exchanged, for example, request messages and response messages.
- The syntax of the various messages types, such as the fields in the message and how the fields are delineated
- The semantics of the fields, that is, the meaning of the information in the fields

- Rules of determining when and how a process sends messages and response messages

One of these protocols are HTTP, and another is SMTP(Simple Mail transfer Protocol)

2.2 The Web and HTTP

Overview of HTTP

The Hypertext Transfer Protocol is a stateless application level request/response protocol that uses extensible semantics and self-descriptive message payloads for flexible interactions with network-based hypertext information systems.

HTTP is described as being a stateless protocol. This is because the protocol is memoryless and [HTTP server] maintains no information about the clients.

Extensible semantics means that the packet is extensible/flexible, packet header is not constant, may add new headers such as cookies (store data at each end).

Self-descriptive message payloads. Payloads are in cleartext, not encrypted.

The HyperText Transfer protocol (HTTP), the Web's application-layer protocol, is at the heart of the Web. HTTP is implemented in two programs: a Client program and a server program. The client program and server program, executing on different end systems, talk to each other by exchanging HTTP messages. HTTP defines the structure of these messages and how the client and server exchange the messages. HTTP defines how Web clients request Web pages from web pages from Web servers and how servers transfer Web pages to clients. When a user requests a Web page, the browser sends HTTP request messages for the objects in the page to the server. The server receives the requests and responds with HTTP response messages that contain the objects. HTTP uses TCP as its underlying transport protocol. The HTTP client first initiates a TCP connection with the server, Once the connection is established, the browser and the server processes access TCP through their socket interfaces. On the client side the socket interface is the door between the client process and the TCP connection, on the server side it is the door between the server process and the TCP connection. The client sends HTTP request messages into its socket interface and receives HTTP response messages from its socket interface. Similarly, the HTTP server receives request messages from its socket interface and sends response messages into its socket interface. When the client sends a message into its socket interface, the message is out of the client's hands and is in the hands of TCP. TCP provides a reliable data transfer service to HTTP . This implies that each HTTP request message sent by a client process eventually arrives intact at the server similarly, each HTTP response message sent by the server process eventually arrives intact at the client.

HTTP with non-persistent

Each request/response transaction is handled by one TCP connection. Each TCP connection is closed after the server sends an object—the connection does not persist for other objects. Most browsers can handle often 5-10 parallel TCP connections.

The non-persistent connection takes the connection time of 2RTT + file transmission time. It takes the first RTT (round-trip time) to establish the connection between the server and the client. The second RTT is taken to request and return the object. This case stands for a single object transmission.

HTTP with persistent connections

With persistent connections, the server leaves the TCP connection open after sending a response. Subsequent requests and responses between the same client and server can be sent over the same connection. Implementing this method counters some of the problems non-persistent connections encounter.

Firstly, a brand-new connection has to be established for each requested object. For each of these connections, TCP buffers must be allocated and TCP variables must be stored in both the client and server. This can result in a significant burden on the web server which may be serving hundreds of requests. Secondly, each object sent over the TCP connections suffers a delivery delay of two RTTs (round-trip time), one to establish connection and one to request/receive object. Web server can self state max objects transferred over one connection and time limit before closing connection.

RTT stands for the round-trip time taken for an object request and then its retrieval. In other words, it is the time taken to request the object from the client to the server and then retrieve it from the server back to the client.

HTTP message format

An HTTP message constructed of three different parts; the request line, header lines and entity body.

The **request line** is the first line of the HTTP request message. The request line has again three different fields, method field, URL field and the HTTP version field. Method field can take different values such as GET, POST, HEAD, DELETE. In the URL field you define which object to request (path to object). Version field is self explanatory.

In the **header line**, certain information about the connection is noted. Such as the user agent (browser) that is making the request, if the TCP connection should be persistent or not or which language the user prefers to recieve.

The **entity body** may contain information that the user has filled in. An HTTP can for example use the POST method when a user fills out a form.

User-Server Interaction: Cookies

As previously mentioned, the HTTP server is stateless. However, sometimes it may be beneficial for a Web site to identify users. Cookie technology is used for this. An HTTP cookie is a small piece of data stored on the user's computer by the web browser while browsing a website. Cookies were designed to be a reliable mechanism for websites to remember stateful information (such as items added in the shopping cart in an online store) or to record the user's browsing activity (including clicking particular buttons, logging in, or recording which pages were visited in the past). They can also be used to remember pieces of information that the user previously entered into form fields, such as names, addresses, passwords, and payment card numbers.

windows format: (1) a cookie header line in the HTTP response message; (2) a cookie header line in the HTTP request message; (3) a cookie file kept on the user’s end system and managed by the user’s browser; and (4) a back-end database at the Web site.

With the Set-cookie response, a Web site may give the user a unique number to remember the certain user. During the subsequent sessions, the browser passes a cookie header to the server, thereby identifying the user to the server.

Cookies perform essential functions in the modern web. Perhaps most importantly, authentication cookies are the most common method used by web servers to know whether the user is logged in or not, and which account they are logged in with. Without such a mechanism, the site would not know whether to send a page containing sensitive information, or require the user to authenticate themselves by logging in. The security of an authentication cookie generally depends on the security of the issuing website and the user's web browser, and on whether the cookie data is encrypted. Security vulnerabilities may allow a cookie's data to be read by a hacker, used to gain access to user data, or used to gain access (with the user's credentials) to the website to which the cookie belongs (see cross-site scripting and cross-site request forgery for examples)

Tracking cookies, and especially third-party tracking cookies, are commonly used as ways to compile long-term records of individuals' browsing histories — a potential privacy concern that prompted European and U.S. lawmakers to take action in 2011. European law requires that all websites targeting European Union member states gain "informed consent" from users before storing non-essential cookies on their device.

2.3 Web caching

A Web cache—also called a proxy server—is a network entity that satisfies HTTP requests on the behalf of an origin Web server. The Web cache has its own disk storage and keeps copies of recently requested objects in this storage.

When a browser establishes a TCP connection to the Web cache and requests a certain object, if the cache does not contain this object the cache will establish a new TCP connection to the main Web server, fetch the object then return it to the first browser. The object is then stored in the cache for later hits.

2.4 …...

2.4.1 HTTP/2

HTTP/2 is a second version of HTTP that enables a more efficient use of network resources using for example header compression and multiple concurrent exchanges on the same connection. Around 30% of web servers use HTTP/2.

2.4.2 DNS

The Domain Name System (DNS) is a hierarchical and decentralized naming system for computers, services, or other resources connected to the Internet or a private network. It associates various information with domain names assigned to each of the participating entities. Most prominently, it translates more readily memorized domain names to the numerical IP addresses needed for locating and identifying computer services and devices with the underlying network protocols. By providing a worldwide, distributed directory service, the Domain Name System has been an essential component of the functionality of the Internet since 1985.

The Domain Name System delegates the responsibility of assigning domain names and mapping those names to Internet resources by designating authoritative name servers for each domain. Network administrators may delegate authority over sub-domains of their allocated name space to other name servers. This mechanism provides distributed and fault-tolerant service and was designed to avoid a single large central database.

The Domain Name System also specifies the technical functionality of the database service that is at its core. It defines the DNS protocol, a detailed specification of the data structures and data communication exchanges used in the DNS, as part of the Internet Protocol Suite.

The Internet maintains two principal namespaces, the domain name hierarchy and the Internet Protocol (IP) address spaces. The Domain Name System maintains the domain name hierarchy and provides translation services between it and the address spaces. Internet name servers and a communication protocol implement the Domain Name System. A DNS name server is a server that stores the DNS records for a domain; a DNS name server responds with answers to queries against its database.

2.5 Email

2.5.1 SMTP

Overview
Simple Mail Transfer Protocol (SMTP) is a protocol for sending electronic mail. The protocol has been the standard protocol for sending e-mails since the beginning of the 80's, and gradually came a further development of the protocol, called Extended SMTP. This allowed other data as files as attachments in addition to the usual ASCII characters.

What you want from SMTP is a protocol that stands for efficient, secure, and reliable sending of electronic mail. Note that SMTP is only responsible for sending email, not retrieval, which must be handled by other protocols. SMTP must be able to send email from client to server, as well as from server to server.

Description
To start an SMTP session, an initialization must first be performed. This initialization depends on whether it is SMTP or ESMTP, but this will be explained in more detail in the operation. To send an e-mail, you have to execute these three commands in the following order: MAIL, RCPT and DATA. These commands will be described later.

1. Initialization
To initiate an SMTP session, the client must first establish a connection between itself and the server. Then the server will respond to the client with a welcome message.
After the welcome message has reached the client, the client must use and send the command "HELO" to begin the session. There are also systems that use ESMTP, where the E stands for Extended. Therefore, an initialization of such sessions will not use the "HELO", but the "EHLO" command. This command provides the client with information about which extensions the server supports.

Once the session has started, three commands are executed to send e-mail. The commands must be executed in the correct order, otherwise the server will give error messages. For example, if an RCPT command appears without a previous MAIL command, a "503 bad sequence of commands'' error message is returned. After the session has started, three commands will be executed to be able to send an e-mail. It is important that these three commands are executed in the correct order or the client will receive error messages from the server. An example of an error message if you execute the commands in the wrong order is a 503 error message. This is a "bad sequence of commands" error message that is sent to the client if an RCPT command is executed before a MAIL command.

2. MAIL
An example of a MAIL command is as follows:

>> MAIL FROM: <reverse-path> [SP <mail-parameters>] <CRLF>
>> MAIL FROM: <test@test.no>

This command says that a new e-mail transaction starts and all data left on the server from previous e-mails must be removed.

Essentially, the command from the example above will initiate a new email transaction, but the command will also remove all of the data from previous emails on the server.

The section: "<reverse-path>" is the e-mail address sent from, but it is also the e-mail where the error messages are sent. If the command is accepted, the client will receive the message returned:
"250 Ok" from the server, otherwise an error message will be returned.

3. RCPT

"RCPT TO" is a command that specifies the recipient of the email. This command can be repeated several times if you want to add many recipients. An example of an RCPT command is as follows:

>> RCPT TO: <forward-path> [SP <rcpt-parameters>] <CRLF>

Here is the "<forward-path>" email address of the recipient. If the address is accepted, the server returns "250 OK" and the address is stored. If the address is not accepted, a 550 error message will be returned as well as an error message of the type "no such user -" and the address.

4. DATA

The DATA command starts transferring the content of the message (text, attachments, etc.). After the DATA command has been sent to the server from the client, the server will respond with a 354 feedback. The message content can then be forwarded to the server. Once the content has been submitted, a period (“.”) Must be submitted on a separate line. If the e-mail is accepted, it will be sent and the server will return "250 OK" to the client.

Ending

To end an SMTP session, the client must first send QUIT, and wait for a response from the server. If the answer is "221 OK", then the session ends. The session can also be ended in other ways, for example by a crash that occurs, or a timeout after DATA. In such cases, servers must be careful not to reject messages that were accepted during the current session.

2.5.2 MIME

Summary
Multipurpose Internet Mail Extensions (MIME) is a technology developed to enable the sending of more than just US-ASCII characters by e-mail. With this technology it is possible to send photos as attachments. MIME is an extension of the Simple Mail Transfer Protocol (SMTP).

This enhancement makes email sending more advanced. Emails contain header fields with the necessary information that describes any attachments so that the recipient's email client can view them.

US-ASCII (7-bit) is standard, but it is often necessary to send 8-bit data. Then Base64 is used, which is a simple transcoding. It is used on binaries, such as images. It makes it possible to send characters outside the US-ASCII, such as æ, ø and å. The data volume increases by 33%, but you do not have to develop a new standard.

MIME and SMTP

A standard e-mail system consists of a set of different protocols for handling the reception and sending of e-mails. For transmission, SMTP is used, which was completed in 1982, ie it is an old standard (RFC821, 1982). MIME is an extension of SMTP and offers options such as sending attachments, and supports a wider character set.

Why we use MIME

The protocol for sending mail, SMTP, was prepared at the same time as ARPANET used TCP and IP to form an Internet consisting of several networks. At this time, the format to be used for emails was determined. The format was called the Standard for the Format of ARPA Internet Text Messages. It only allowed text with characters included in the original ASCII, which is 7-bit. In the MIME context, this character set is called US-ASCII.

The format was sensible for use at the time, since SMTP was to be used to send simple messages. But as SMTP established itself to a greater extent with the general public, and newer technology allowed faster transfers, it was clear that a new solution was needed. Instead of creating a new format and updating all e-mail servers that were in operation, it was decided to make an extension of the existing forma.

RFC 2045 states that the purpose of MIME is to enable characters not included in US-ASCII, to allow a large number of formats of attachments such as images and audio files, and to allow multiple different attachments in the same email. MIME is thus created to be flexible and will ensure that SMTP can be used further, despite its weaknesses.

Header fields

Each email contains header fields that provide information about the MIME version and any attachments. These operate in the same way as those in HTTP. Previously, e-mail was limited to technology in SMTP format. Some limitations are that one can not transfer multimedia and 8-bit US-ASCII format. Along with general information about sender, recipient and date, you have a header field that tells which MIME version is used. Attachments and text with characters outside US-ASCII are placed under header fields that define, among other things, file type and character set. This information is used to decode the data when it arrives at the recipient. The e-mail client in which the e-mail is opened can interpret the data and present it in the right file format.

Base64 coding

MIME uses coding to transmit data. The coding can be done in many different ways, depending on the type of data to be sent. Base64 encoding is used on binary files, such as images. Base64 also enables the transmission of national characters such as æ, ø and å - which do not really exist in US-ASCII.

With base64 coding, you first deal with 3 bytes, ie 24 bits (3 x 8 bits). Then you read 6 bits at a time. Then you end up with 4 new characters (4 * 6 bit). Furthermore, a base64 table is used to convert each character to a US-ASCII character. Finally, the new 8-bit US-ASCII character is transmitted

3. Transport Layer

3.1 Transport-layer services

The transport layer works as the end-system part of the end-to-end system communication.

A transport layer protocol provides for logical communication between application processes running on different hosts. By logical communication, we mean that from an app's perspective, it is as if the hosts running the processes were directly connected in reality the host may be on opposite sides of the planet, connected via numerous routers and a wide range of link types. Application processes use the logical communication provided by the transport layer to send messages to each other, free from the worry of the details of the physical interface structure used to carry these messages.

Whereas transport layer protocol provides logical communication between *processes* running on different hosts a network-layer protocol provides logical communication between *hosts*.

The internet makes two distinct transport layer protocols available to each application layer. One of these protocols is UDP, which provides an **unreliable**, connectionless service to the invoking app. The second of these protocols is TCP which provides a **reliable**, **connection oriented** service to the invoking apps. UDP is unreliable which means that it doesn't not guarantee that the data sent by one process will arrive intact. TCP on the other hand is reliable and uses flow control, sequence numbers, acknowledgements and timers.

3.2 Socket

A socket is one endpoint of a two-way communication link between two programs running on the network. A socket has a port number of 16-bit, ranging from 0 – 65535. Port numbers ranging from 0 – 1023 are called *well-known port-numbers* and are restricted (reserved) for well known application-protocols such as HTTP – 80, FTP – 21.

Source port number field and destination port number field.

3.3 Multiplexing and Demultiplexing

In telecommunications and computer networks, multiplexing is a method by which multiple analog or digital signals are combined into one signal over a shared medium. The aim is to share a scarce resource. For example, in telecommunications, several telephone calls may be carried using one wire.

Demultiplexing (Demuxing) is a term relative to multiplexing. It is the reverse of the multiplexing process. Demultiplex is a process of reconverting a signal containing multiple analog or digital signal streams back into the original separate and unrelated signals.

Although demultiplexing is the reverse of the multiplexing process, it is not the opposite of multiplexing. The opposite of multiplexing is inverse multiplexing (iMuxing), which breaks one data stream into several related data streams. Thus, the difference between demultiplexing and inverse multiplexing is that the output streams of demultiplexing are unrelated, while the output streams of inverse multiplexing are related.

3.3.1 Connectionless multiplexing and demultiplexing

A UDP contains a source number and destination number in a data-transfer, source number serves a part as a “return address”.

3.3.2 Connection-oriented multiplexing and demultiplexing

TCP socket is identified by a four-tuple: (source IP address, source port number, destination IP address, destination port number). Thus, when a TCP segment arrives from the network to a host, the host uses all four values to direct (demultiplex) the segment to the appropriate socket.

3.4 UDP

UDP (User Datagram Protocol) is a communications protocol that is primarily used for establishing low-latency and loss-tolerating connections between applications on the internet.

UDP is called a connectionless protocol. This means that the protocol does not require to establish a connection before the protocol can begin to transfer data. As soon as an application process passes data to UDP, UDP will package the data inside a UDP segment and immediately pass the segment to the network layer.

UDP does also not contain a congestion control mechanism like TCP (throttling the transport layer, when one or more links between source and destination become excessively congested). TCP will also continue to resend a segment until the receipt of the segment has been acknowledged by the destination. Because UDP does not contain all these reliable

data-transfer principles, real-time applications, who often require a minimum sending rate and who can tolerate data loss, may find UDP favourable to TCP

- No connection establishment can “blast” away data once instructed.
- No connection state, does not have to set up buffers, congestion control parameters, acknowledge and sequence parameters.
- Small packet header overhead (8 bytes).

Unlike TCP, UDP doesn't guarantee that the packets will get to the right destinations. That means that UDP doesn't connect to the receiving computer directly -- which TCP does. Rather, it sends the data out and relies on the devices in between the sending and receiving computers to correctly get the data where it's supposed to go.

3.5 TCP

The current version of the TCP protocol allows two endpoints in a shared computer network to establish a connection that enables a two-way transmission of data. Any data loss is detected and automatically corrected, which is why TCP is also called a reliable protocol. TCP also provides congestion control.Congestion control is a function for preventing any one TCP connection from “clogging” the links and routers between communicating hosts with an excessive amount of traffic.
TCP allows for transmission of information in both directions. This means that computer systems that communicate over TCP can send and receive data at the same time, similar to a telephone conversation.

The TCP software is controlled by the various network applications, such as web browsers or servers, via specific interfaces. Each connection must always be identified by two clearly defined endpoints (client and server). It doesn’t matter which side assumes the client role and which assumes the server role. All that matters is that the TCP software is provided with a unique, ordered pair consisting of IP address and port (also referred to as "2-tuple" or "socket") for each endpoint.

Other information about TCP

- Header on 20 bytes (large, can have optional data on additional 40 bytes)
- Splits message in several slices before sending
- Requires a stateful connection
- Needs to establish a connection, 3 way handshake
- Data transfer with receipt
- Ability to retransmit “lost” packets
- Receiver feedback (Acknowledge flags such as ACK, NAK)
- Error detection (contains a checksum)

3.5.1 Retransmission principles of TCP

Go-Back-N

A Go-Back-N (GBN) protocol, the sender is allowed to transmit multiple packets (when available) without waiting for an acknowledgment, but is constrained to have no more than some maximum allowable number, N, of unacknowledged packets in the pipeline

Selective Repeat

As the name suggests, selective-repeat protocols avoid unnecessary retransmissions by having the sender retransmit only those packets that it suspects were received in error (that is, were lost or corrupted) at the receiver. A single packet error can thus cause GBN to retransmit a large number of packets, many unnecessarily

3.6 TLS

Transport Layer Security (TLS), and its now-deprecated predecessor, Secure Sockets Layer (SSL), are cryptographic protocols designed to provide communications security over a computer network. Several versions of the protocols are widely used in applications such as email, instant messaging, and voice over IP, but its use as the Security layer in HTTPS remains the most publicly visible.

The TLS protocol aims primarily to provide privacy and data integrity between two or more communicating computer applications. When secured by TLS, connections between a client (e.g., a web browser) and a server (e.g., wikipedia.org) should have one or more of the following properties:

The connection is private (or secure) because a symmetric-key algorithm is used to encrypt the data transmitted. The keys for this symmetric encryption are generated uniquely for each connection and are based on a shared secret that was negotiated at the start of the session. The server and client negotiate the details of which encryption algorithm and cryptographic keys to use before the first byte of data is transmitted. The negotiation of a shared secret is both secure (the negotiated secret is unavailable to eavesdroppers and cannot be obtained, even by an attacker who places themself in the middle of the connection) and reliable (no attacker can modify the communications during the negotiation without being detected).
The identity of the communicating parties can be authenticated using public-key cryptography. This authentication is required for the server and optional for the client.
The connection is reliable because each message transmitted includes a message integrity check using a message authentication code to prevent undetected loss or alteration of the data during transmission.
In addition to the above, careful configuration of TLS can provide additional privacy-related properties such as forward secrecy, ensuring that any future disclosure of encryption keys cannot be used to decrypt any TLS communications recorded in the past.

TLS supports many different methods for exchanging keys, encrypting data, and authenticating message integrity. As a result, secure configuration of TLS involves many configurable parameters, and not all choices provide all of the privacy-related properties.

Attempts have been made to subvert aspects of the communications security that TLS seeks to provide, and the protocol has been revised several times to address these security threats. Developers of web browsers have repeatedly revised their products to defend against potential security weaknesses after these were discovered (see TLS/SSL support history of web browsers).

The TLS protocol comprises two layers: the TLS record and the TLS handshake protocols.

4. Network Layer

4.1 The tasks of the network layer

The main responsibility of the network team is routing. The network team determines the route a packet should take from the sending machine to the receiving machine. Structured addresses are used. This is why you can use a start address to represent the entire route to a group of units and up to the destination. Application data, i.e. what is to be sent, is grouped into segments / packets and further the network layer transfers these data packets between the machines in the network where the application is running. It is also this team that addresses messages and translates logical addresses into physical ones.

It is on the network layer that machines operate in networks. Which machines are in the network does not matter; on the network layer, all machines are equal and are represented by a logical address. Networks can be connected, including in the Internet, so that machines in different networks can communicate. Then a network on a machine will have a common network address towards the Internet, which everyone on the network uses when communicating outside their network.

Routers make sure all the packages get where they need to go. Routers connect all the networks by having two or more interfaces, one for each network. Among other things, a home router will have two interfaces, for the home network and one for the Internet. They receive a packet, process it to make sure it can be forwarded, read the address and then decide which output the packet should be forwarded to.

4.1 Forwarding and Routing

The role of the network layer is simply to move packets from a sending host to receiving host. Two important network-layer functions are:

- Forwarding: When a packet arrives at a routers input link, the router must move the packet to the appropriate output link.
- Routing: The network layer also needs to determine the route or path taken by the packets.

4.1.1 Forwarding

Forwarding refers to the router-local action of transferring a packet from an input link interface to the appropriate output link interface. Every router has a **forwarding table**. Before the router forwards a packet, it examines the value of a field in the arriving packet's header. The value stored in the forwarding table entry for that header indicates the router's outgoing link interface to which that packet is to be forwarded

Datagram network
Each time an end system wants to send a packet, it stamps the packet with the address of the destination end system and then pops the packet into the network. Each router has a forwarding table to appropriately forward the packet to the correct output link.

4.1.2 Routing

Routing refers to the network-wide process that determines the end-to-end paths that packets take from source to destination.

The network service model defines the characteristics of end-to-end transport of packets between sending and receiving end systems. Example of this could be Guaranteed delivery, Guaranteed delivery with bounded delay, Guaranteed minimum bandwidth etc.

4.1.3 Routing table

When routers are used in an interconnected network, each router will build a routing table that contains information about the preferred route from one machine to another.
A routing table consists of:

- Network Destination - The destination address of the network.
- Subnet mask - the part of the network address that must match if the route is to be used.
- Gateway - the address to which the data packet may be forwarded. These are network cards or nearest routes.
- Interface - the address of the network card that the data packets must pass through.
- Metric (M) - number of jumps to destination network.
- You can see your own routing table by using the command: netstat or sudo route -n

4.2 Router

A router consists of four main parts

- **Input ports**:
  - Performs the physical layer function of terminating an incoming physical link at a router
  - Needs to interoperate with the link layer.
  - Lookup function/ forwarding table
- **Switching fabric**
  - The switching fabric connects the router's input ports to its output ports.
- **Output ports**
  - An output port stores packets received from the switching fabric and transmits these packets on the outgoing link by performing the necessary link-layer and physical-layer functions
- **Routing processor**
  - The routing processor executes the routing protocols
  - computes the forwarding table for the router
  - network management functions (path calculations)

Packet loss may happen within the router because of exhausted memory when storing packets. This can both happen at input port and output port. Implementation of packet schedulers may help with this.

There are mainly 3 different methods of implementing switching. Switching via memory, switching via bus and switching via an interconnection network.

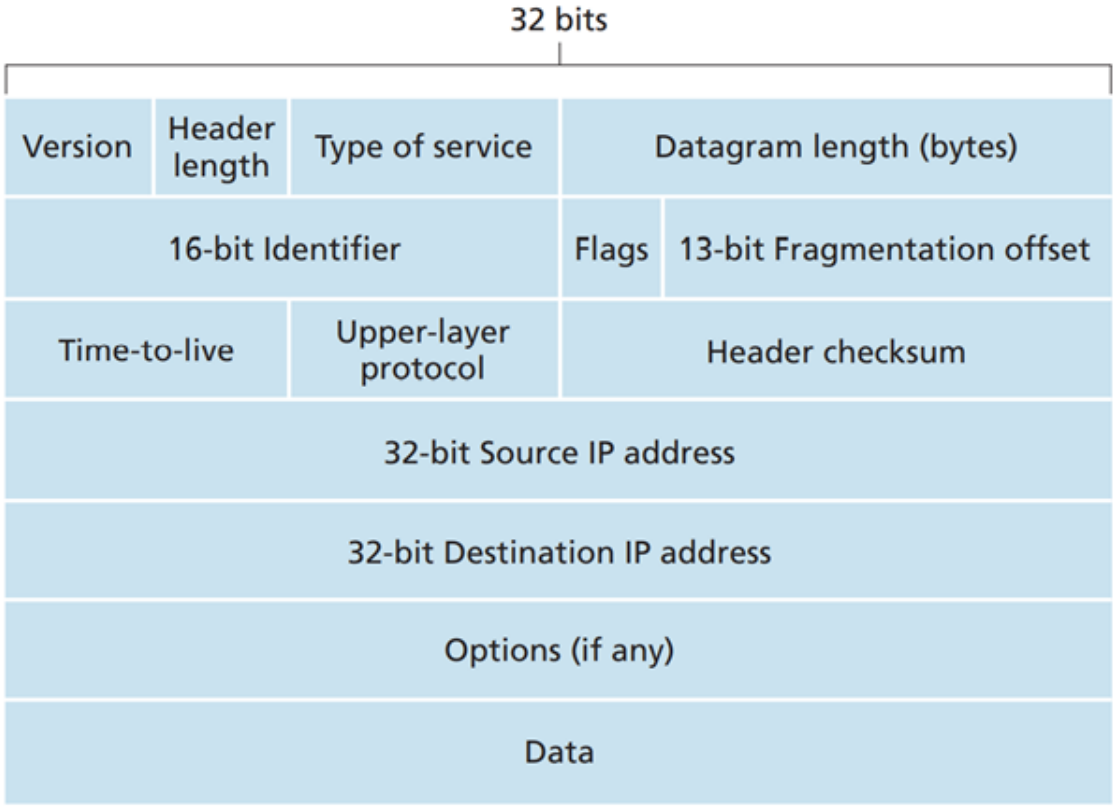
4.3 The Internet Protocol

4.3.1 IPv4

IPv4 is an abbreviation for Internet Protocol version 4, which was the first widespread version of the IP protocol. It was first used by ARPANET in the early 1980s. IPv4 has been the main system for all communication over the internet since it was introduced. In recent times, it has slowly but surely begun to be replaced by IPv6, but is still used in most areas. The IPv4 system has 32-bit addresses. It provides 4,294,967,296 different address options. Parts of these are reserved for special data such as private addresses. To communicate, IPv4 uses communication between routers to send the data packets. The system uses "best effort delivery". This means that it tries as best it can to get all the packages sent correctly,

but it has no systems or checks that guarantee this. These checks will take place on other teams in the team model.

4.3.1.1 Datagram format



Version number: These 4 bits specify the IP protocol for the datagram.
Header length: 4 bits determining where the data of the datagram begins.
Type of service: TOS bits allows datagrams to be distinguished from each other, such as real-time datagrams.
Datagram length: Total length of IP datagram, theoretical max size is 2^16 = 65355
Identifier, flags, fragmentation offset: IP fragmentation at router
Time-to-live: ensures that datagrams do not circulate forever, decrements by one for each router passed
Protocol: This field is used only when an IP datagram reaches its final destination. Determines which transport layer protocol the datagrams data should go to, 6 – TCP, 17 – UDP
Header checksum: The header checksum aids a router in detecting bit errors in a received IP datagram.
Source and destination IP addresses: When a source creates a datagram, it inserts its IP address into the source IP address field and inserts the address of the ultimate destination into the destination IP address field.

Options: The options fields allow an IP header to be extended.
Data: Contains the transport layer segment (TCP, UDP).

4.3.2 Addressing

Subnet
To determine the subnets, detach each interface from its host or router, creating islands of isolated networks, with interfaces terminating the end points of the isolated networks. Each of these isolated networks is called a subnet.

4.3.2.1 CIDR

CIDR is a method used to interpret IP addresses. This syntax replaces the previous system with classes, Classful IP addressing, used before 1993. This makes it easier to divide larger blocks of IP addresses when it is to be distributed over smaller networks. CIDR allows IP addresses to be shared exactly where you want, which in practice gives 32 different "classes", which of course gives more freedom of choice than just 5.

Classful IP addressing is that the IP addresses are divided into 5 different classes: A, B, C, D and E. This was done to be able to diversify the subnet so that if it is a small subnet it can use class C where it is only 256 sub-IPs, but in return several 2 million different subnets to distribute. It turned out that there was too little variation between the classes and CIDR thus replaced classful IP addressing to provide more freedom of choice.

IPV4 example (Norwgian)

Oppgave 4: Nettverk subnetting (20%)

Du har gitt et IPv4-nettverk a.b.c.0/24. Dette skal deles i subnett med antatt følgende behov:

A: 80 ansatte, B: 10 tjenermaskiner, C: 20 gjestebrukere

For å utnytte subnettene mest mulig og samtidig gi dem så lave adresser som mulig plasseres de forskjellige gruppene i subnett der de fyller mest mulig av kapasiteten. Siden 80 ansatte ikke får plass i nett B med 62 plasser så plasseres de i det største nettet med 126 plasser og lavest adresser, hvorpå det da blir 46 ledige plasser. Det samme prinsippet går for tjenermaskinene og gjestebrukerne, vi plasserer de 20 gjestebrukerne i neste nett med 30 plasser og 10 plasser som blir stående ledig. Tjenermaskinene havner i nettet etter dette med 14 plasser og det blir 4 ledige plasser der. Totalt blir det da 60 ikke utnyttede plasser i de utvalgte subnettene. Kaller de to resterende ubrukte nettene D og E, og de vil ha totalt 76 ledige plasser som alle vil ha høyere adresse enn de ansatte, tjenermaskinene og gjestebrukerne.

Nett	Nettadresse	CIDR	Min host	Max host	Broadcast	Tilgjengelige	Behov	Ledig
A	a.b.c.0	/25	a.b.c.1	a.b.c.126	a.b.c.127	126	80	46
C	a.b.c.128	/27	a.b.c.129	a.b.c.158	a.b.c.159	30	20	10
B	a.b.c.160	/28	a.b.c.161	a.b.c.174	a.b.c.175	14	10	4

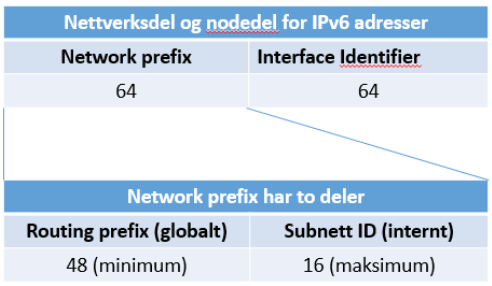
D	a.b.c.176	/28	a.b.c.177	a.b.c.290	a.b.c.291	14	0	14
E	a.b.c.192	/26	a.b.c.290	a.b.c.254	a.b.c.255	62	0	62

IPV6 example (Norwgian)

1. Les forklaring på engelsk Wikipedia om IPv6 address (2.1) om Unicast adresseformat. Lag en figur som viser sammenhengen mellom Network prefix, Routing prefix, subnett ID og Interface Identifier for en IPv6 adresse. Hvilken størrelse kan Routing prefix og Subnett ID ha?
2. En virksomhet disponerer 2001:db8::/60 og har behov for fire subnett med mellom 25 og 150 brukere på hvert nett. Sett opp en adresseplan for disse subnettene slik at subbnett med de laveste adressene tas i bruk. Hvor mange ledige subnett vil virksomheten fortsatt ha?

1. Network prefix er 64 byte og består av Routing prefix for Internett-ruting og subnett ID for adressering i virksomhetens lokalnett. Antall bit i Network prefix er 64 bit, som fordeles på Routing prefix med minimum 48 bit og subnett ID med maksimum 16 bit.
2. En Routing prefix på 60 bit gir 4 bit til interne subnett ID. 2^4 gir rom for inntil 16 subnett. De fire laveste tas i bruk

- 2001:db8:0:0::/64 (som også kan skrives 2001:db8::/64)
- 2001:db8:0:1::/64
- 2001:db8:0:2::/64
- 2001:db8:0:3::/64



Virksomheten vil fortsatt ha inntil 12 ledige subnett med samme nettmaske-lengde, eventuelt disse kan samles i færre, men større Subnett. (Merk: Interface-ID er alltid 64 bit)

The IP address

The Internet is made up of hundreds of thousands of IP networks, each of which has unique URLs. An IP address consists of two parts, a web address (web ID) and a node address (host ID) and is 32-bit. The web address in the IP address tells us which IP networks the machine belongs to, and is important when the router is to forward IP packets. Routers only look at the URL when IP packets are to be forwarded, and leave it to the last router to make sure that the IP packet arrives at the right machine.

Today, there are two versions of IP, IPv4 and IPv6. IPv6 is a longer 128-bit IP address, as opposed to IPv4's 32-bit. The reason why this IP protocol was introduced is because IPv4 does not have enough unique IP addresses for today's consumption. In this section we talk about IPv4, but the principle is the same for IPv6.

The subnet mask

The problem is that we can not distinguish what is the URL and node address just by looking at the 32-bits that make up the IP address. To solve this, we have a subnet mask that tells us where the difference goes. the subnet mask tells us how many bits from left to right make up the URL. The sum of the URL and the node address must always be 32-bit, so that we get an IP address. Each machine must know its own subnet mask in order to identify computers belonging to the same IP network. the subnet mask is 32-bit, where the number of 1-bits corresponds to the length of the URL.

Internet's address assignment strategy is known as Classless Interdomain Routing (CIDR—pronounced cider) [RFC 4632]. CIDR generalizes the notion of subnet addressing.

As with subnet addressing, the 32-bit IP address is divided into two parts and again has the dotted-decimal form a.b.c.d/x, where x indicates the number of bits in the first part of the address. The x most significant bits in this format is called the prefix or network prefix

Only the x bits are the ones that are significant for outer networks. The remaining 32 – x bits identifies the specific hosts in the subnet.

Broadcasting

When a host sends a datagram with destination address 255.255.255.255, the message is delivered to all hosts on the same subnet.

4.3.3 DHCP - Obtaining a Host Address

A DHCP server automatically assigns (allocate) a computer an:

- IP address
- Subnet mask
- Default gateway
- DNS server

The DHCP server can assign IP addresses within its scope (the blocks it has). When the DHCP server assigns an IP, it assigns it as a lease. The server can create reservations, meaning a given host receives the same IP address each time it connects to the network, or a host may be assigned a temporary IP address that will be different each time the host connects to the network.

4.3.4 Network address translation (NAT)

The NAT router behaves to the outside world as a single device with a single IP address. The NAT device performs a technique called IP masquerading. This technique hides an entire IP address space, usually consisting of private IP addresses (your own subnet), behind a single IP address in another, usually public address space. The hidden addresses are changed into a single (public) IP address as the source address of the outgoing IP packets so they appear as originating not from the hidden host but from the routing device itself.

Remaps an IP address space into another by modifying network address information in the IP header of packets while they are in transit across a traffic routing device.

4.3.4 ICMP

ICMP, specified in [RFC 792], is used by hosts and routers to communicate network-layer information to each other. Primarily used for troubleshooting networks (error checking and reporting functionality). Ping command (performs Echo request – eche reply).

Traceroute

The tracert command is implemented by using ICMP messages. To determine the names and addresses of the routers between source and destination, the function sends a series of pings, incrementing the TTL for every contact point. When each router observes that the TTL of the datagram (ping) has expired, it sends back an ICMP message (type 11 code 0) including router name and IP address.

4.3.5 IPv6

IPv6 is a new standard for IP addresses. It took over for the previous standard IPv4 to deal with the lack of IP addresses. From IPv4 to IPv6 there have been several changes. Among other things, the address space has been expanded from 32 to 128 bits and simplification of packet headers. Fragmentation of packages is also done differently.

An address is made up of several parts so that the data sent over the network will reach the correct recipient. How such an address is formatted can be done in several ways where the most common are "Unicast" and "Multicast".

Some of the IP addresses are reserved for special purposes.

One of the main reasons why IPv4 should be replaced by a new standard was to address the need for a lack of IP addresses. With the development of IPv6, you got expanded address space from 32 to 128 bits. In addition, packet header was simplified in IPv6 with a fixed size of packet header, and there was also better support for time-critical and real-time applications. Furthermore, there were also new extensions for confidentiality, integrity and authentication (encryption).

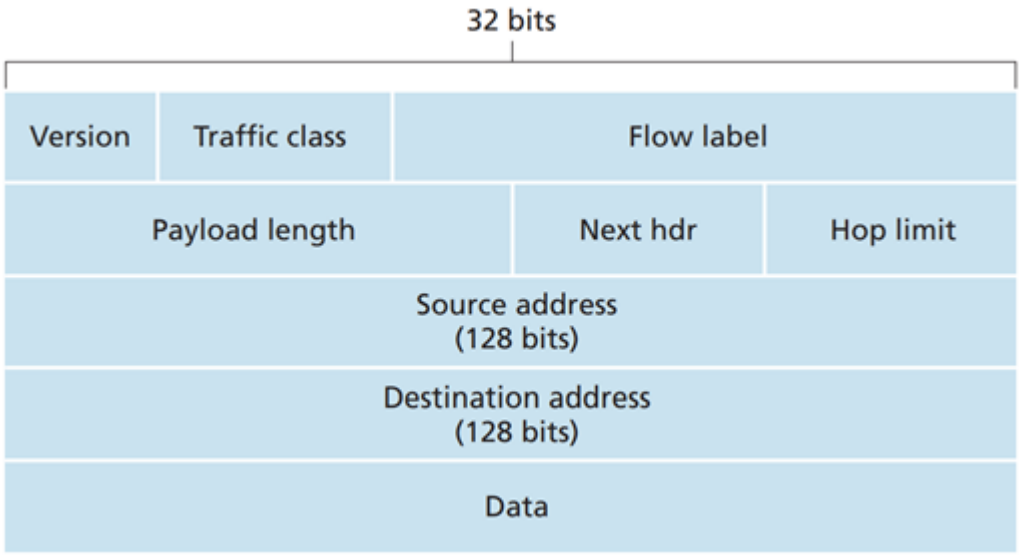
Communication between IPv4 and IPv6

There are mainly two different methods for IPv4 and IPv6 nodes to communicate with each other.

Dual-stack nodes: Capable of sending/receiving both version

Tunneling: Works by using an intervening set of IPv4 routers. The routers construct an IPv4 datagram, taking in the entire v6 datagram into its own field. If the now IPv4 datagram is received by an IPv6 node, it can simply extract the data version 6, if received by v4.

4.3.5.1 Datagram format



- **Expanded addressing capabilities**: Increased size of the IP address from 32-bit to 128-bit, not leaving “every grain of sand on the planet to be IP-addressable”.
- **Anycast address**: In addition to multicast and unicast, the anycast address allows a datagram to be delivered to any one of a group of hosts.
- **Streamlined 40-byte header**: a 40-byte fixed-length header allows for faster processing of the IP datagram. A new encoding of options allows for more flexible options processing.
- **Flow labelling and priority**: Allowing labelling of packets belonging to different flows. A video or audio stream might be treated as a flow. Also enables prioritizing certain datagrams within a flow. Ex ICMP over network news.
- **Version**: This 4-bit field identifies the IP version number
- **Traffic class**: This 8-bit field is similar in spirit to the TOS field we saw in IPv4.
- **Flow label**: As discussed above, this 20-bit field is used to identify a flow of datagrams.
- **Payload length**: 16-bit value stating the number of bytes in the datagram.
- **Next header**: Identifies the protocol to which the contents (data field) of this datagram will be delivered (for example, to TCP or UDP).

- **Hop limi**t: The contents of this field are decremented by one by each router that forwards the datagram. If the hop limit count reaches zero, the datagram is discarded.
- **Source** and destination addresses: The various formats of the IPv6 128-bit address are described in RFC 4291.
- **Data**: This is the payload portion of the IPv6 datagram. When the datagram reaches its destination, the payload will be removed from the IP datagram and passed on to the protocol specified in the next header field.
- **Fragmentation/reassembly**: Not allowed at intermediate routers. Fragmentation and reassembly is only allowed at source and destination. If a IPv6 datagram is to large to be
- **Header checksum**
- **Options**: options field is one of the possible next headers pointed to from within the IPv6 header

4.3.6 Autonomous systems (AS)

Imagine an AS as being like a town's post office. Mail goes from post office to post office until it reaches the right town, and that town's post office will then deliver the mail within that town. Similarly, data packets cross the Internet by hopping from AS to AS until they reach the AS that contains their destination Internet Protocol (IP) address. Routers within that AS send the packet to the IP address.

Every AS controls a specific set of IP addresses, just as every town's post office is responsible for delivering mail to all the addresses within that town. The range of IP addresses that a given AS has control over is called their "IP address space."

Most ASes connect to several other ASes. If an AS connects to only one other AS and shares the same routing policy, it may instead be considered a subnetwork of the first AS.

Typically, each AS is operated by a single large organization, such as an Internet service provider (ISP), a large enterprise technology company, a university, or a government agency.

4.7 NIX - NORWEGIAN INTERNET EXCHANGE

4.7.1 Summary

Norwegian Internet Exchange (NIX) is Norway's largest Internet Exchange Point and the Norwegian interconnection point for exchanging traffic between service providers (ISP). Edge routers from each ISP are connected in a switch through which all traffic flows.

The main task of an Internet Exchange Point, or exchange point, is to allow 2 or more networks to communicate directly with each other.

4.7.2 Tasks

In Norway, most autonomous systems are connected to NIX. This is a switch - at first it was a single machine located in Oslo - now such interconnection points have been established in several places in Norway to increase capacity and safety.

Several different autonomous systems can have connections between them without going through NIX, and one system can communicate with another via an intermediate system, ie a transit network. There are financial interests related to agreements and exchange of traffic between autonomous systems (peering). If two systems have the same amount of traffic in and out, you can make an agreement without payment. NIX is based on this principle. If you want to connect to transit networks, you have to pay.

NIX operates as an exchange point for IP interconnection in Norway. Most internet service providers in Norway use this to distribute their services to users in Norway with both good availability and speed. NIX also offers data exchange for large broadcasting companies such as NRK and TV2.

In short, NIX is Norway's largest exchange point for data traffic, and operates as an alternative for the middle layer between systems when they are to communicate.

4.7.3 Behavior

NIX has 6 different nodes in Norway:

- NIX1 and NIX2 in Oslo spread over different locations.
  - NIX1 is the main point spread over 3 different locations.
  - NIX2 is the backup point in Oslo and is located at St. Olav's place.
- BIX in Bergen is located at the University of Bergen in the IT Department there.
- TRDIX in Trondheim is located at NTNU Gløshaugen (Realfagbygg A, 3rd floor).
- TIX in Tromsø is located at UiT Norway's Arctic University.
- SIX in Stavanger is located at the University of Stavanger.

Where the 4 local points BIX, TRDIX, TIX and SIX are intended mainly for local traffic to reduce the pressure on the main points in Oslo.

None of these IXPs are connected, but act as standalone points.

5 Link layer

Summary

The link layer is layer 2 in the simplified OSI model. The team is divided into LLC (Logical Link
Control) and MAC (Media Access control). LLC is above MAC and provides a common interface to protocols on the network layer, thus hiding the various media characteristics.
MAC provides adaptation to different transmission media that have different electrical and signaling properties

The task of the link layer is to transfer packets between the network layers on adjacent ones nodes. It prepares packages through framing and other mechanisms, before the packages transferred to the physical layer. In addition, the link team can detect and correct errors that can
occur on the physical layer.

MAC has mechanics to prevent collision during transmission of packets being sent over the net. A MAC address is unique to each network card, this does not change (not 100% correct to say) and makes it possible to identify items. This address will be sent to the frame head during parcel shipping. MAC is also responsible for the check amount in the frame tail during transmission. During reception, these values will be validated, which leads to that the shipment is accepted or rejected.

IEEE 802 is a collection of LAN protocols. IEEE 802.3 defines the Ethernet protocol, and used in wired networks. WLAN is used for wireless local area networks, and defined by IEEE 802.11. Each of these utilizes different protocols to avoid

collisions, such as CSMA / CD and CSMA / CA. The ARP protocol translates IP addresses to
MAC addresses.

5.1 Description

The link layer is next to the bottom of the simplified OSI model. Over this team the network layer is located. The network layer sends data down to the link layer where it stays
processed and forwarded to the physical layer which ensures that signals are sent out to internet or a local network. The link layer is divided into two sub-layers, Logical Link Control (LLC) and Media Access Control (MAC). The link layer is divided into sub-layers to hold transmission technology hidden from the network layer. This is because the protocols on The network layer is general, and should work across different network technologies such as for example ADSL and Ethernet.

5.2 Tasks

The task of the link layer is to transfer packets between adjacent nodes in local (LAN, Local Area Network) and Wide Area Network (WAN). The link team receives packets from the network layer, then prepares the packets through framing and other mechanisms, before being transferred to the physical layer. The link team can too detect and correct any defects that may occur on the physical layer.

5.2.1 LLC (Logical Link Control)

LLC controls the logic of the link layer, this includes synchronization and error checking. Synchronization involves determining the start and end of each data frame and
error checking is described earlier in the article.
LLC acts as an interface between the network layer and the MAC. It is used to handle addressing and multiplexing (sending) / demultiplexing (receiving) of protocols over the MAC layer. This functionality allows protocols to operate in parallel

in the same network. Multiplexing is a method of combining separate signals a joint. This method is beneficial because it maximizes transmission capacity.

5.2.2 MAC (Media Access Control)

Controls how data is transferred between transmission technologies (wireless, wired, fiber etc.).

MAC secures transfers between two components for network collision. To achieve this allocates network access to the devices and prevents them from transferring simultaneously. To identify if another component is trying to transfer using the Carrier Sense algorithm Multiple Access with Collision Detection, abbreviated CSMA / CD. A collision occurs if multiple devices are trying to send information simultaneously on the same network.

A MAC address consists of six groups of two hexadecimal numbers. An example is: "00-13-22-02-23-45". Everything connected to the network has a MAC address. This the address is unique and can therefore be used as an identifier for the devices on the network.

Some systems allow this to be changed. Some vendors reuse MAC addresses, but usually this is not a problem unless two or more devices with the same address are connected to the same network. This rarely happens, but if it happens, the system will not work properly.

The MAC team is responsible for creating the frame head and frame tail. A frame is one kind of package in the link layer. The frame header consists of the MAC address and

control bytes. The frame tail contains a checksum for the entire frame. It's MAC
who is responsible together with the physical team to transfer the frame.
When the MAC receives a frame from the physical layer, it is responsible for accepting the frame.
This is done by checking the MAC address in the frame header. To check if the frame does not
is corrupt, the check amount is verified in the frame tail. Ramming and framing requires a lot processing power and the NIC card implement this.

5.3 Protocols

Since the link layer is divided into two sub-layers, you will see that different protocols are dedicated to
each layer. One group of standards that is central in this context is IEEE 802
and contains LAN protocols. One of them is IEEE 802.2, which defines the behavior and the rules of LLC. The protocols used on the MAC layer depend on the hardware
which is in use. For example, IEEE 802.3 is a standard that describes Ethernet, and IEEE 802.11 defines the WLAN protocol. These protocols have different formats
their frames. The ARP protocol is used to enable devices to determine which ones
MAC addresses of other network devices.

5.3.1 Ethernet (IEEE 802.3)

Ethernet is a widely used technology for transmitting data over a local area network.
Network devices are connected through a router or switch with physical cables.
Earlier, the CSMA / CD algorithm was mentioned. Before a device starts sending data, it checks if there is traffic on the line. If two frames collide, both senders will wait
each at random before they start sending again. The algorithm was used in the elderly ethernet technologies where the cables were half-duplex. In newer cables that are full-duplex, this is no longer an issue as data can be sent in both directions simultaneously.

5.3.2 WLAN (IEEE 802.11)

IEEE 802.11 defines the protocol used to implement a local wireless
network between devices. This standard consists of several revisions, the new ones often offers better speed and longer distances. These are identified by letters where it
The latest to date is 802.11ax. (IEEE 802.11, 2020).

WLAN uses a CSMA variant called CSMA / CA, where CA stands for Collision Avoidance. CSMA / CA works by a device that wants to communicate over one channel, will check at random time intervals if it is available before sending its data. CSMA / CA also uses a protocol called RTS / CTS to avoid collisions. If
a device A wants to communicate with a router over a high-traffic channel,
A can send a Ready-To-Send signal to the router. The router asks all other devices to stop before sending a Clear-To-Send signal to A. (PowerCert, CSMA / CD and CSMA / CA, 2019)

5.3.3 ARP

ARP stands for Address Resolution Protocol and is a protocol that in the link layer is used to translate from IP addresses to MAC addresses. If computer A wants to communicate with computer B on a local network, A must obtain the MAC address of B. If A does not have this stored, but knows the IP address, A will send out a broadcast message as requests the corresponding MAC address for the IP address. With WireShark you can see how a device sends out a message FF: FF: FF: FF: FF: FF and requests that those who know the MAC address of 10.0.0.138 to take contact.

6 Links

Video recordings https://sites.google.com/view/tdat2004-datakom/video

Subject summary website: https://sites.google.com/view/tdat2004-datakom/