4. The transport layer

4.1 The port number

One of the most important information contained in the header of a segment are the destination and the source port numbers.

The port numbers are necessary to identify the application to which a data unit belongs to.

The port number is a 16-bit number, so it can have the decimal values 0 to 65535 ($=2^{16}$-1). The first 1024 port numbers (0-1023) are the so called *well known port numbers*. They are used to identify the *server applications* as these port numbers have to be "well known" to the client applications.

application (protocol)	port number
web server (http)	80
web server (https)	443
FTP (ftp)	20 & 21
Email (pop3)	110
Email (imap)	143
Email (smtp)	25

Each server application receives always the same port number.

The operating system of a host is assigning one of the *other port numbers* to each *client application*. The range of dynamic port numbers is logically (1024 - 65535).

Each client application receives dynamically a different port number.

This technique is necessary to identify for example several open browsers on the same host.

4.2 TCP and UDP

The two most common protocols used on the transport layer are TCP and UDP.

4.2.1 TCP

The TCP protocol adds a 24 bytes header to the data unit. The structure of the TCP header is:

Bit 0 **Bit 15**	**Bit 16** **Bit 31**	
Source Port (16)	Destination Port (16)	24 bytes
Sequence Number (32)		
Acknowledgement Number (32)		
Header Length (4) Reserved (6) Code Bits (6)	Window(16)	
Checksum (16)	Urgent(16)	
Options (0 or 32, if any)		
Data		

The sequence and acknowledgement number allow checking if all the segments of a transmission session have reached the destination. This assures the reliability of the communication.

The windows size is the number of bytes that the sender of the TCP segment is able to receive into its buffer. This technique is called flow control.

The checksum allows detecting transmission errors.

Usually a TCP segment contains 1500 bytes of data. If many errors occur during a transmission, it can be useful to reduce the number of data bytes. Among others this can be done with the option bits. This technique is called dynamic window sizing (do not confuse the window size).

4.2.2 UDP

The UDP protocol adds a 8 bytes header to the data unit. The structure of the UDP header is:

Bit 0 **Bit 15**	**Bit 16** **Bit 31**	
Source Port (16)	Destination Port (16)	8 bytes
Length (16)	Checksum (16)	
Data		

The length field specifies the number of bytes contained in the whole segment (incl. header).

4.2.3 TCP versus UDP

The only but important advantage of UDP versus TCP is that the quantity of header information is lower. Thus the percentage of data bits within a UDP transmission is higher than with TCP. This leads to a higher net transmission speed.

TCP however allows a reliable and error free transmission.

In most applications the correctness of the transmission is more important than the speed. This is why most applications use the TCP protocol. Applications that typically use UDP are so called streaming applications like:

- video streaming
- audio streaming (i.e. voice over IP)

4.3 Firewall

Definition

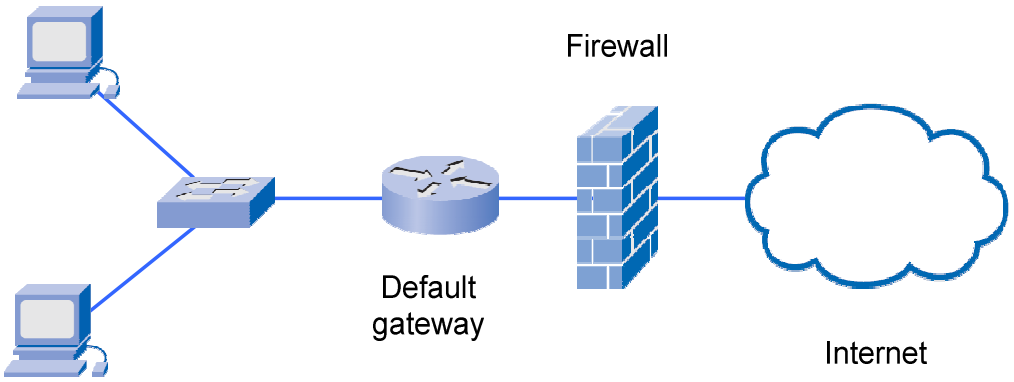
> **A firewall is a network security system that can block the incoming and the outgoing network traffic based on rules set within the firewall.**

We can classify all firewalls into two types:

1. External firewalls
2. Personal firewalls

External firewall:

An external firewall is a hardware component switched between the router and the internet. The external firewall analyzes traffic to and from a whole network.



Personal firewall:

A personal firewall is a software installed on a host. The personal firewall analyzes the traffic to and from the host it is installed on. This software can be part of the operating system.

Functionalities

Simple firewalls operate on the transport layer (layer 4).

This fact tells us that a firewall is able to analyze from which or to which software / port number a frame is sent to. Thus all firewalls are able to distinguish between incoming and outgoing traffic. Under *outgoing traffic* we understand a request going out of our network including the incoming answer to the request.

Consequently under *incoming traffic* we understand a request coming in our network including the outgoing answer to the request.

Most firewalls for private networks are configured by default such that all outgoing traffic is allowed to pass the firewall, but all incoming traffic is blocked. This configuration is fine for most private networks only consisting of hosts requesting information from outside of the network.

However if you install a *server application within the private network* which needs to be accessible for requests coming from the internet, e.g. private web servers or gaming server, then an exception to the general rule for incoming traffic has to be added. We say "*a port has to be opened in the firewall*". This exception rule will allow requests to pass the firewall that are addressed to the port number defined in the rule. The firewall rules can often be manipulated over a web-interface.

Example: