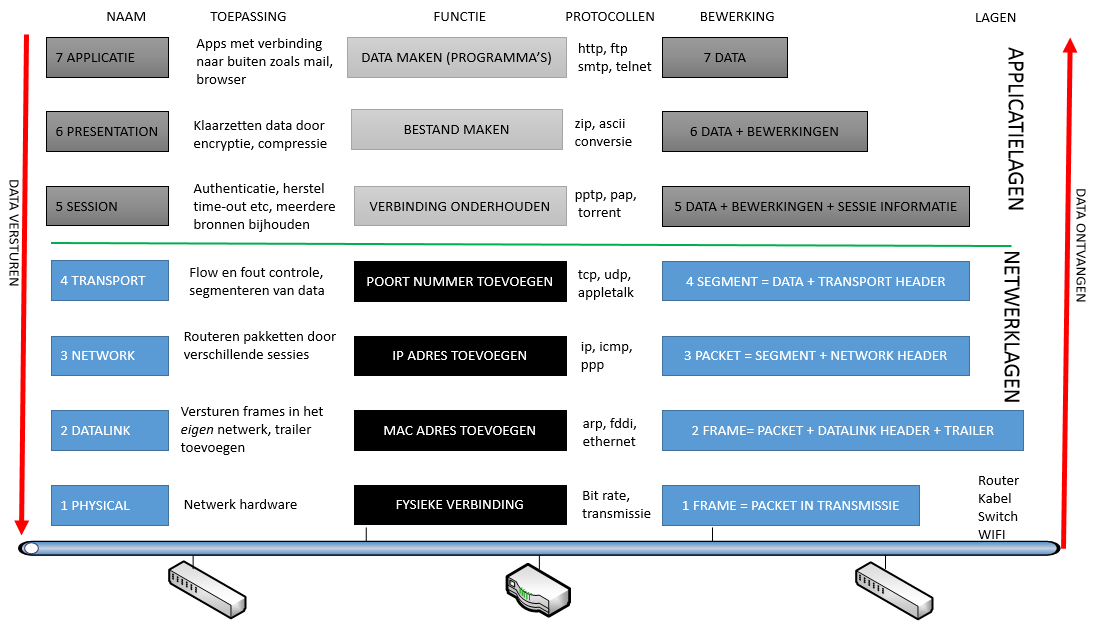
OSI Model

Hieronder zie je een overzicht van het zogenaamde OSI Model. We gaan eens nader kennismaken.

Bestudeer het onderstaande schema.



OEFENING 1 DE LAGEN

Welke lagen zijn de applicatielagen? MAAK DE NAMEN VET

7 applicatie

6 presentatie

5 sessie

4 transport

3 netwerk

2 Datalink

1 fysiek

Welke lagen zijn de netwerklagen? MAAK DE NAMEN VET

7 application

6 presentation

5 session

4 transport

3 network

2 Datalink

1 physical

Wat betekent OSI eigenlijk?

OEFENING 2 DE NAMEN VAN DE LAGEN ONTHOUDEN

Een manier om te onthouden hoe de lagen heten is er een acroniem (of heel precies een niet-dichtend acrostichon) ontwikkeld:

All People Seem To Need Data Processing.

In de zin hierboven zijn de beginletters van de woorden ook de beginletters van de lagen.

Een docent heeft het volgende acroniem bedacht: A Priest Saw The Nice Doctor Pray

VERZIN ZELF EEN ACRONIEM

## DE MTU

De lagen van het OSI model zorgt ervoor dat informatie die jij wilt verzenden in pakketjes van maximaal 1500 bytes aan data verstuurd kunnen worden.

Dat zit zo:

Computers communiceren in 0-en en1-en. Een 0 of een 1 is een bit

* 8 bits noemen we een byte
* 1 byte is dus 8 bits
* 1024 bytes noemen we een KiloByte (KB)
* 1024 KB noemen we een MegaByte (MB)
* 1024 MB noemen we een GigaByte (GB)

Zoek op het internet:

1024 GB noemen we een \_\_\_\_\_\_\_\_\_\_Byte (TB)

1024 TB noemen we een \_\_\_\_\_\_\_\_\_\_Byte (PB)

1024 PB noemen we een \_\_\_\_\_\_\_\_\_\_Byte (EB)

1024 EB noemen we een \_\_\_\_\_\_\_\_\_\_Byte (ZB)

1024 ZB noemen we een \_\_\_\_\_\_\_\_\_\_Byte (YB)

In de communicatie tussen computers werken we vooral met bytes en kilobytes. Zoals gezegd is het maximale aantal bytes wat je in een pakket kan versturen 1500 bytes.

Als je dus 4 MB aan dat moet versturen dan heb je

4 MB = 4x1024 = 4096 KB

4096 KB x 1024 = 4194304 BYTES

4194304 / 1500 = 2797 (afgerond naar boven) pakketjes die verstuurd moeten worden om 4 MB te verz.enden

OEFENING REKENEN

Bereken het aantal pakketjes wat je nodig hebt om te versturen:

Wat wordt verstuurd of ontvangen? Aantal pakketjes

Ik verstuur een mail van 1200 bytes \_\_\_\_\_\_\_\_\_

Ik mail een document van 8600 bytes \_\_\_\_\_\_\_\_\_

Ik mail een document van 1 MB \_\_\_\_\_\_\_\_\_

Ik download een film van 512 MB \_\_\_\_\_\_\_\_\_

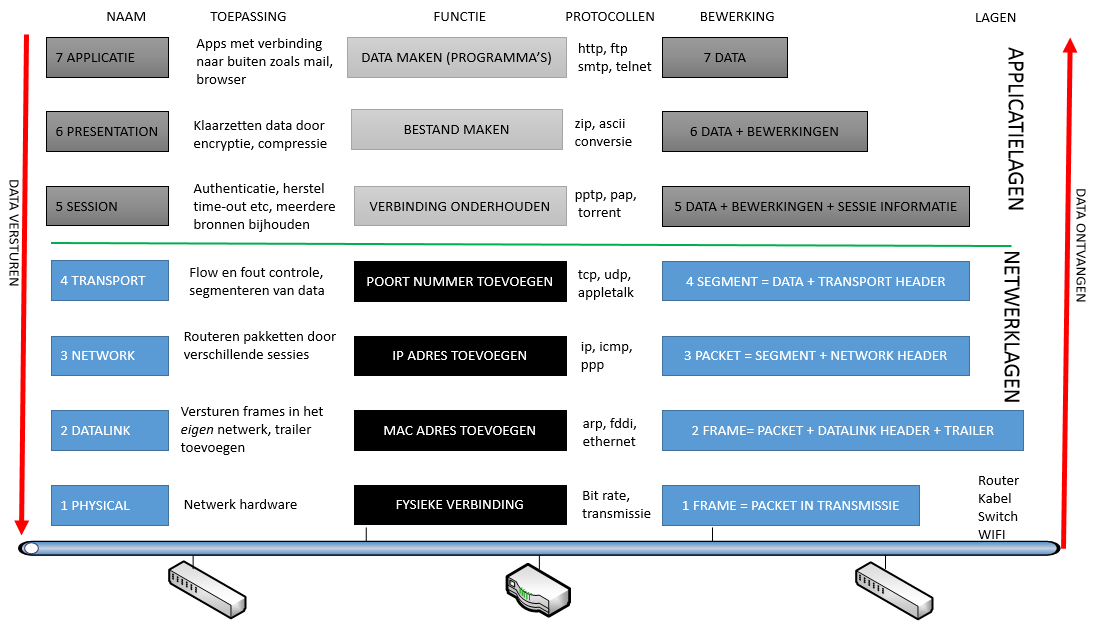
Zodra je over een netwerk verbinding data gaat sturen moet je dit in stukjes van **1500 bytes** per keer doen.

We noemen bij “gewone” netwerken (die noemen we **ethernet** netwerken) dit de MTU de **MAXIMUM TRANSMISSION UNIT**.

Officieel hebben ethernet netwerken een MTU van 1500 bytes.

# Laag 7. De applicatielaag

Hieronder zie je dat de functie van de applicatielaag het maken van de data is. Dit geldt voor applicaties die met het netwerk kunnen verbinden.



OEFENING APPLICATIE LAAG

Laten we de applicatielaag eens aan het werk zetten.

Open Notepad op je computer.

Schrijf je naam in het documentje. **Mail dit documentje rechtstreeks vanuit Notepad** naar je buurman.

JUIST. DAT GAAT NIET

Open Microsoft Word op je computer.

Schrijf je naam in het document. **Mail dit document rechtstreeks vanuit Word** naar je buurman.

DAT GAAT WEL, MAAR DE MAIL APPLICATIE WORDT EERST GEOPEND EN DAN PAS KAN JE MAILEN

Noem vier programma’s die zelf data kunnen versturen (in principe alles met een ZEND knop of een DOWNLOAD knop)

Laag zeven maakt het bestand wat je vervolgens wilt opslaan en versturen. De applicaties hierboven kunnen wel data maken, maar deze NIET zelf versturen.

Om te versturen open je in Laag 7 een andere applicatie, bijvoorbeeld MAIL of een BROWSER, en die kan wel gegevens versturen.

Veel programma’s kunnen in de applicatie laag werken. Maar niet alle programma’s kunnen de data verzenden.

De reden is dat veruit de meeste programma’s geen netwerk protocollen kunnen gebruiken. Wat protocollen zijn komt later aan de orde.

Ik noem er hier drie, welke denk je dat bij welke hoort? MAAK HET JUISTE ANTWOORD VET

Bestanden uploaden en downloaden SMTP, FTP, HTTP

Een website opvragen SMTP, FTP, HTTP

Een mail versturen SMTP, FTP, HTTP

Kan je nog even de betekenis van deze afkortingen beschrijven?

SMTP \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

FTP \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

HTTP \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Beschrijf nu in je eigen woorden (en zoek op google) waarvoor laag 7 dient.

# LAAG 6. PRESENTATIELAAG

In de presentatielaag worden data klaargezet voor verzending. Als je in de applicatielaag je data hebt gemaakt, zijn er vaak nog wat bewerkingen nodig om de data te kunnen versturen.

Een van die bewerkingen is dat de data wordt omgezet naar ASCII code. Je gaat kennismaken met de manie waarop computers met letters en tekens omgaan.

PRESENTATIELAAG TAAK 1: TEKEN CONVERSIE

Computers kunnen alleen met getallen overweg, met 1-en en 0-en. Letters en tekens kan de computer niet begrijpen, de processor kan geen H of en Y lezen alleen maar 0-en en 1-en. Alle letters en getallen moeten dan ook worden omgezet naar het binaire stelsel. Dit gebeurt in de presentatielaag.

Met getallen heeft de computer weinig moeite, die worden eenvoudig binair berekend. Hoe dit verloopt wordt later besproken.

Met tekens zoals (&^%@) en met gewone letters heeft je computer veel meer moeite. Eerst moeten de letters worden omgezet naar een getal en dan binair worden gemaakt. Het omzetten van letters naar getallen wordt conversie genoemd.

Wij mensen werken met letters en tekens, de computer heeft dan ook een methode om dit razendsnel te kunnen omzetten (converteren).

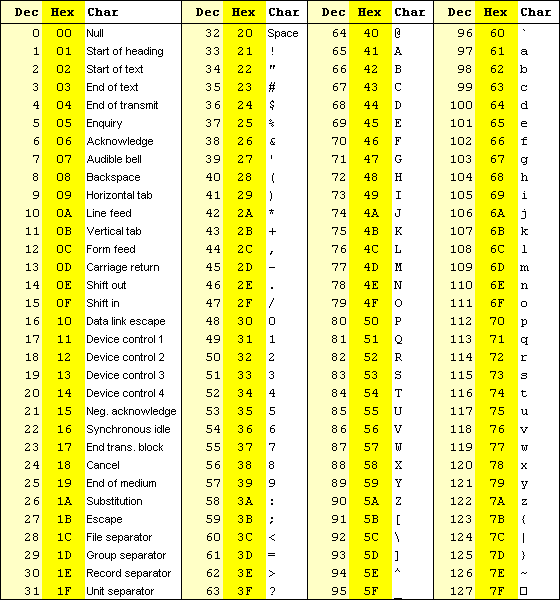
De computer maakt hierbij gebruik van de zogenaamde ASCII tabel.

## ASCII (American Standard Code for Information Interchange)

Verreweg de meest bekende methode om letters en tekens om te zetten naar getallen is de zogenaamde ASCII tabel. Deze tabel laat zien welke getallen worden gebruikt voor welke tekens in jou computer. In bijlage 1 zie je de standaard ASCII tabel.

In de tabel zie je drie kolommen.

* In de eerste zie je een decimale waarde
* In de tweede een hexadecimale waarde (belangrijk voor een onderzoek!)
* In de derde kolom het karakter

q

Oefening ASCII

Om ASCII code aan het werk te zien ga je naar notepad. Als je notepad hebt gestart activeer je num lock door rechts op het toetsenbord Num Lock te drukken.

Ga nu met je cursor in notepad staan. Typ nu rechts op het numerieke toetsenpanel 104 terwijl je ALT ingedrukt houdt. Je ziet de letter h verschijnen.

Zoek in de tabel je voor- en achternaam op en schrijf de je naam hieronder in ASCII code. Plaats spaties tussen de codes. Denk aan hoofdletters.

Naam in ASCII **: 065 108 097 101 045 101 100 100 105 110 101 032 101 108 032 072 097 110 105**

Computer slaan de ascii codes op in 1 byte per teken, dus 8 bits. Je hebt dan ook 256 (28) mogelijke combinaties en dan is de tabel vol (let op: in de ascii tabellen zie je ook dat de hoogste waarde 255 is met de waard 0 erbij 256 mogelijke combinaties.

We gaan een stukje code bekijken waarin ascii code hexadecimaal is weergeven.

Probeer met behulp van de tabellen te achterhalen wat er staat.

Hieronder zie je een typische stukje met ascii code in hexadecimale vorm. Deze code bevat 2 straatnamen en plaatsnamen.

4d61 696e 2053 742e 0000 0000 0000 0000………....rij 0-15 bytes

Maint St.

426f 7374 6f6e 0000 0000 0000 0000 0000 ......rij 16-31 bytes

Boston

1900 536f 7574 6820 5374 2e00 0000 0000 ………...rij 32-47 bytes

South St.

4e65 2059 6f726b000 0000 0000 0000 0000 ......rij 48-63 bytes

Bij deze kom ik uit op ney nogwat (fout)

Iedere rij bevat 16 bytes telkens in groepjes van twee bytes. Alle bytes die op 00 staan zijn leeg.

Let op: ieder byte is twee leestekens, dus de 4d61 code bestaat uit vier bytes en is dus twee tekens!

## TEKENCODERING

Je hebt in het bovenstaande kennis gemaakt met ASCII code. Zolang je een toetsenbord gebruikt met de normale alfabet letters heb je ruim voldoende plaats om alle tekens en letters van een ascii code te voorzien. Als je echter naar andere landen, andere talen en andere tekens kijkt dan kom je heel veel ruimte tekort. ASCII gebruikt maar 1 byte dus 8 bits om een teken op te slaan en bij 256 tekens is de tabel vol. Om dit probleem aan te pakken zijn UTF en Unicode ontwikkeld. Hier gaan we niet verder op in.

Onthoudt dat de letters en tekens op je computer ook binair moeten worden bewaard. Daarom zijn alle letters en tekens gekoppeld aan een getal. Dit getal kan je ook hexadecimaal weergeven. Deze tabellen zijn op je computer aanwezig in de CPU registers.

Als er nu door jou in de applicatielaag een tekst wordt aangemaakt ie je wilt versturen, wordt in de presentatielaag de tekens omgezet naar een format (ascii, unicode, UTF) wat ook door de ontvangende computer wordt begrepen.

COMPRESSIE

Compressie betekent dat digitale informatie wordt zo wordt verwerkt dat het aantal bits kleiner wordt. Je stuurt het dan op en de ontvanger pakt het weer uit. Het bekendste voorbeeld is waarschijnlijk het programma zip wat bestanden kan inpakken (compressie ook wel zippen) en vervolgens bij de ontvanger weer uitpakken (decompressie ook wel unzippen).

Als je continue op het internet werkt is het heel belangrijk dat je goed met je bandbreedte omgaat. Als deze te vol wordt dan wordt je verbinding langzaam. Door nu bestanden te comprimeren kan je heel wat bandbreedte winnen.

Voordat we hier naar gaan kijken gaan we zelf een compressie methode bedenken. Belangrijk is natuurlijk dat je niet alleen kunt inpakken maar vervolgens ook uitpakken.

1. Als er enen worden ingepakt dan begint en eindigt de reeks met een 1
2. Als er nullen worden ingepakt dan begint en eindigt de reeks met een 0
3. We tellen de enen en de nullen van iedere rij en geven deze hexadecimaal weer

Kijk naar dit rijtje

111111111 00000000 11111111111111111111 000000000000000 1111111111101000000000

De negen enen: wordt 191

De acht nullen wordt 080

De twintig enen wordt 1201

De vijftien nullen wordt 0150

De elf enen wordt 1111

De losse nul wordt 010

De losse een wordt 111

De laatste nullen wordt 090

Op een rijtje wordt dit

191080120101501111010111090

Dit getal is nu binair

1101010000100100001111001000110000110111000111011011100100110

Dat is flink wat korter dan de oorspronkelijke reeks

OEFENING COMPRESSIE

Kort nu zelf deze reeks in zodat je deze ook weer kunt uitpakken.

1111111111000000000001111110110111111111111111100000000001111000001111111

Hanteer de bovenstaande methode of verzin zelf een methode.

HTTP COMPRESSIE

Als verbinding met een website legt wordt deze in je browser opgebouwd door het http protocol. Ook de ontwikkelaars van dit protocol weten dat bandbreedte kostbaar is. Langzame computers zijn een verschrikking voor de bedrijfsvoering.

