Computernetwerken 1
Theorie

M. Vandemaele R. Swennen
Rudi.Swennen@ucll.be

13.09.2018

Voorwoord

Deze tekst is de cursus bij het vak Computernetwerken - deel 1 van het 1° jaar Bachelor Toegepaste Informatica aan de UC Leuven-Limburg. In dit vak bestuderen we de netwerken die computers met mekaar verbinden. Veel computergebruikers werken (misschien onbewust) met data die via een computernetwerk verkregen of opgeslagen worden. Deze data bevinden zich op een andere computer dan deze waarvan ze het toetsenbord bedienen. Denken we maar aan toepassingen als het web, email, netwerkschijven (dit zijn eigenlijk folders die zich op een andere computer bevinden maar die er uitzien als een harde schijf van de eigen computer), ...
Het doel van het vak Computernetwerken is te leren:

- welke vaktermen er gebruikt worden
- wat deze termen betekenen
- welke apparatuur er vereist is om computers te laten communiceren
- welke de werkingsprincipes zijn van deze apparatuur
- welke afspraken (protocollen) er vereist zijn bij communicatie
- hoe deze afspraken in computerprogramma's geïmplementeerd zijn
- welke de verschillende aspecten van communicatie via een computernetwerk zijn
- ...

Deze vrij complexe materie wordt op een eenvoudige en geordende wijze uiteengezet.
Van de onderwerpen van onderhavige cursus bestaan presentaties in Powerpoint en begeleidende labo's. Collega's kunnen hiervoor contact opnemen met de auteur via zijn emailadres. Op ditzelfde adres luistert de auteur naar alle opmerkingen of kritiek (zowel positieve als negatieve).
Aan de inhoud van deze cursus werd grote zorg besteed. Toch aanvaarden noch de auteur, noch de uitgever enige aansprakelijkheid voor schade die het gevolg zou zijn van eventuele fouten of onvolkomenheden in deze cursus.

Hoofdstuk 1

Inleiding

Sinds het ontstaan van computers bestaat de behoefte om bestanden van de ene machine te kopiëren naar de andere. Er is een tijd geweest dat bestanden gekopieerd werden op een externe drager (magneetband, verwisselbare harde schijf, diskette,...) en door een andere machine weer ingelezen werden.

Nu, alles kan beter: uiteindelijk is een bestand in een computer opgeslagen als een simpele bitrij. Langs een fysieke verbinding zowel bedraad of draadloos kunnen **signalen** verstuurd worden. Wanneer twee signalen respectievelijk een 0 (0V gedurende 0,2s) of een 1 (2V gedurende 0,2s) voorstellen, kunnen door middel hiervan diverse bitrijen overgebracht worden tussen de twee computers. Bestanden kopiëren, zoals een webpagina kan dan ook veel efficiënter als computers fysiek met elkaar verbonden zijn.

In de beginjaren van computers en computernetwerken was het ondoenbaar om elke computer rechtstreeks met elke andere computer te verbinden. Men kon wel alle computers aansluiten op een gemeenschappelijke bekabeling. Deze bekabeling werd **netwerk** genoemd. Dit netwerk vormt de fysieke verbinding die computers gebruiken om gegevens rechtstreeks met elkaar uit te wisselen. Deze gemeenschappelijke bekabeling kan tegelijkertijd door meerdere computers gebruikt worden.

Zo een netwerk bestaat uit:

- transmissiemedia voor signalen (kabels of draadloze verbindingen)
- schakelapparaten
- voor elke aangesloten computer een interface (hardware én software) tussen de computer en het transmissiemedium

Dit geheel laat toe een bitrij, zoals een webpagina te versturen van de ene computer naar de andere.

Om het netwerk correct te kunnen gebruiken is programmatuur of software nodig. De software neemt volgende taken voor zijn rekening:

- stelt bitrijen samen
- plaatst bitrijen op het netwerk
- ontvangt bitrijen van het netwerk
- interpreteert ingekomen bitrijen

De apparatuur, zowel computer- als schakelapparatuur (vanaf nu netwerkapparatuur genoemd), zorgt ervoor dat bitrijen verstuurd en ontvangen worden. Beide componenten van computernetwerken, **apparatuur** en **programmatuur** worden in deze tekst verder uitgediept.

Wat de netwerkapparatuur betreft, zijn er alvast 2 belangrijke verschilpunten:

- computers binnen één gebouw of binnen een groep van gebouwen (een campus) zijn verbonden door een LAN (local area network)
- computers die verder van mekaar verwijderd zijn (stad, land, wereld), zijn verbonden door een WAN (wide area network)

1.1 Protocollen (voor computernetwerken)

Als mensen communiceren (via een gesprek over de telefoon of per brief) houden ze zich aan bepaalde afspraken. Men begint een telefoongesprek bijvoorbeeld met een begroeting: "Hallo, u spreekt met Theo de Raadt. Wat kan ik voor u doen?". Het adres op een briefomslag schrijven wij allemaal op dezelfde wijze, omdat dit zo vastgelegd is. Ook computerprogramma's die communiceren moeten zich houden aan afspraken. Omdat mensen intelligente wezens zijn, kunnen wij communiceren zonder al te strikt vastgelegde afspraken. Voor computerprogramma's moeten de afspraken **wel** strikt vastgelegd worden.

Bij communicatie tussen mensen is het meestal zo dat één persoon het initiatief neemt en iets vraagt of iets meedeelt aan een ander persoon. Een vereiste hierbij is dat de andere persoon luistert (eventueel op een later tijdstip). Bij communicerende computerprogramma's wordt hetzelfde principe toegepast. Er is een programma dat luistert en wacht tot een ander programma het initiatief neemt. Men noemt dit programma het **serverprogramma**. Het programma dat het initiatief neemt, richt meestal een verzoek aan het serverprogramma. Het initiatiefnemend programma heet **cliëntprogramma**.

Om communicatie tussen computerprogramma's mogelijk te maken, moet er strikt vastgelegd of gespecifiëerd zijn:

- welke berichten er kunnen verstuurd worden
- wat elk bericht betekent
- wat de onderlinge verhoudingen zijn tussen de berichten, o.a. hoe op elk bericht kan gereageerd worden

Dit geheel van specificaties of afspraken wordt **protocol** genoemd. Omdat communicatie tussen computerprogramma's over een netwerk vrij complex is, wordt een principe van *verdeel en heers* toegepast. Met geheel van specificaties of afspraken wordt hierboven dus bedoeld: alles wat betrekking heeft op één welbepaald aspect i.v.m. computercommunicatie. Per aspect wordt een protocol opgesteld. Wat we met "aspect" bedoelen zal verder duidelijk worden, maar toch al dit:

- er zullen afspraken zijn over de wijze waarop een bit voorgesteld wordt
- er zullen afspraken zijn i.v.m. foutcontrole
- er zullen afspraken zijn om meerdere computers dezelfde fysieke verbinding te laten gebruiken zonder dat ze mekaar storen
- enz...

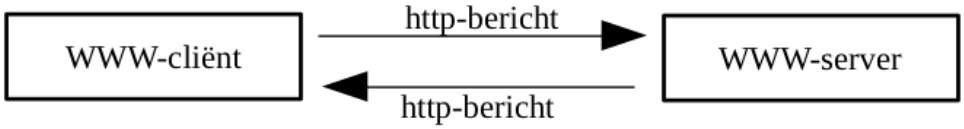
Dit zijn duidelijk verschillende aspecten. Per aspect zal er een set van afspraken zijn: een **protocol**. Verschillende aspecten worden vastgelegd is verschillende protocollen die samen een **protocolstack**[1] vormen.

Hoe de specificaties van een protocol of een reeks van afspraken in een programma geïmplementeerd worden, is de vrijheid van de ontwerper van het programma. Nemen we als voorbeeld de gekende WWW-toepassing. Verschillende softwareleveranciers maken verschillende cliëntprogramma's en/of verschillende serverprogramma's voor het World Wide Web. Er zijn verschillende browsers of WWW-clients, Firefox en Opera om er maar twee te noemen. Aan de server-kant bestaan er Apache webserver, NGINX, Lighttp, … . WWW-browsers en WWW-servers mogen verschillen wat betreft de gebruikersinterface, de programmeertaal waarin ze geschreven zijn, de bevelen die de programmeurs gebruikt hebben, enz. Wat strikt hetzelfde moet zijn, zijn de berichten die naar een serverprogramma gestuurd worden, alsook de berichten die een server(programma) terugstuurt. We kunnen dit vergelijken met een programmeeroefening. Alle studenten leveren een ander programma af, maar al deze programma's doen precies wat er gespecifiëerd werd.

[1] Een hoopje protocollen die voortbouwen op elkaar

Het protocol dat voor World Wide Web gebruikt wordt, heet HyperText Transfer Protocol of **HTTP**. Een WWW-cliënt en een WWW-server wisselen berichten met mekaar uit. Deze berichten voldoen aan de specificaties van het HTTP.

Schematisch:



Figuur 1.1: Uitwisseling van HTTP-berichten.

Oefening 1 Bedenk zelf een menselijk uitvoerbaar protocol om datum en tijd te vragen aan iemand.
Wat doet de "vrager"
Wat doet de "antwoorder"
Welke afspraken moeten worden nageleefd?

Oefening 2 Bedenk zelf een applicatieprotocol van de tijd-applicatie waarin de gebruiker volgende vragen kan stellen:

- Hoe laat is het?
- Welke datum is het?
- Welke dag is het?

Wat doet de cliënt?
Wat doe de server?
Wat zijn de afspraken?

Een netwerkprotocol is analoog aan een mensenprotocol waarbij de bericht-uitwisselende mensen hard- of software-entiteiten zijn.

1.2 Protocollagen

Een computernetwerk kan ook gezien worden als een infrastructuur die **gedistribueerde applicaties** voorziet van diensten. Denk zelf eens na hoe B-POST een infrastructuur aanlevert van diensten en hoe wij, als mens, deze diensten op een specifieke manier moeten gebruiken.

Als een programma, een gedistribueerde applicatie (bvb. een WWW-cliënt) een bericht stuurt naar een programma draaiend op een andere machine (bvb. een WWW-server), dan wordt hierbij het netwerk gebruikt. Nu is er een zeer grote verscheidenheid aan netwerken (niet alleen WAN vs LAN, maar binnen elk type LAN zijn er verschillende technologieën mogelijk: ethernet, token ring, ATM, …). Men kan natuurlijk niet verlangen dat een WWW-surfer telkens een andere browser zou moeten gebruiken al naargelang het type netwerk (LAN, ADSL, kabel, ...). Maar hoe komt het eigenlijk dat dezelfde browser bij verschillende type netwerken kan gebruikt worden?

Het antwoord luidt als volgt:

1. De browser moet een HTTP-bericht niet zelf omvormen tot een bitrij en hij moet deze bitrij niet zelf op het netwerk zetten. Anders zouden alle types van netwerken in de browser "moeten ingebouwd" zijn.

2. De browser zal zijn bericht afleveren aan programmatuur op een *lager niveau* die ervoor zorgt dat het HTTP-bericht omgevormd wordt in een bitrij en op het netwerk geplaatst wordt.

3. Hetzelfde geldt voor de berichten die de browser ontvangt: programmatuur op een *lager niveau* ontvangt een bitrij en levert het HTTP-bericht dat hierin vervat zit, af aan de browser.

Aldus, vooraleer een bericht van een applicatie of toepassing (WWW, E-mail, bestandsoverdracht, terminaal-emulatie, enz...) op het netwerk kan gezet worden, moet het door de netwerkprogrammatuur bewerkt worden. De netwerkprogrammatuur is verdeeld in verschillende **lagen**:

- Op het hoogste niveau, de **applicatielaag** of **toepassingslaag**. Op deze laag worden communicatieboodschappen uitgewisseld tussen de toepassingen die gebruik maken van het netwerk. Deze applicatieboodschappen kunnen een rechtstreeks gevolg zijn van de interactie tussen de applicatie en de gebruiker.

- Op het laagste niveau, de **fysieke laag**. Deze laag verstuurt de bitrij op het netwerk en zet ontvangen signalen van het fysieke communicatiemedium om in een bitrij.

- Tussen beide lagen bevinden zich een aantal **tussenlagen**. Verder in deze cursus zullen deze lagen grondig bestudeerd worden.[2]

Het is belangrijk even stil te staan bij het verschil tussen protocol en programma. Een protocol is een set van afspraken en een programma is één

[2]Toch al enkele voorbeeldjes: een besturingsprogramma voor een netwerkkaart bevindt zich in de onderste van deze tussenlagen; een bericht aannemen van een browser gebeurt door de bovenste tussenlaag. Voor elke laag zijn strikte afspraken vastgelegd in protocollen.

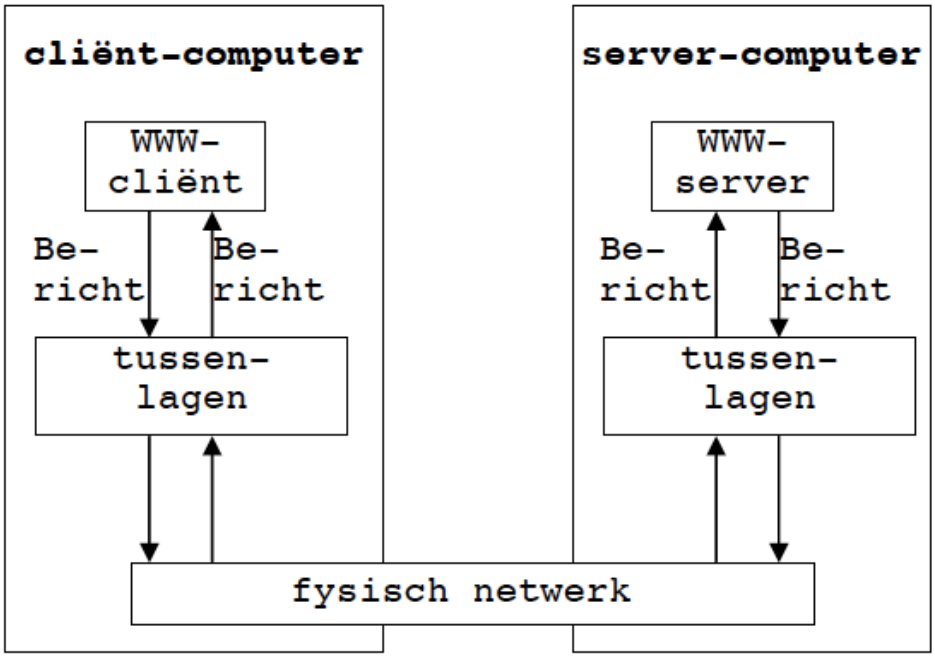
van de mogelijke implementaties van deze afspraken. Zo is HTTP een protocol en zijn Firefox en Apache-server programma's die deze protocollen implementeren. Het zijn deze programma's die berichten uitwisselen die beantwoorden aan het HTTP.
Eigenlijk heeft men de afspraken i.v.m. netwerken ingedeeld in (abstractie)lagen. Een laag op het hoogste niveau, nl. de toepassingslaag, hoeft geen kennis te hebben van het type netwerk of type communicatiemedium dat gebruikt wordt om bijvoorbeeld de HTTP-berichten te versturen of ontvangen. Het HTTP-bericht wordt via onderliggende tussenlagen vertaald in een bitrij. Via de laagste laag wordt de bitrij d.m.v. fysieke signalen over het communicatiemedium verstuurd of omgekeerd ontvangen en omgezet in een bitrij. De programmeur van de applicatie die boodschappen verstuurt in de toepassingslaag kan bijgevolg abstractie maken van het type netwerk of type communicatiemedium dat het netwerk gebruikt. Door de afspraken i.v.m netwerken op te delen in lagen heeft dit uiteraard tot gevolg dat ook de programmatuur die deze protocollen implementeert, in lagen verdeeld is.

De indeling in lagen heeft twee belangrijke voordelen:

- De complexiteit kan verdeeld worden d.m.v. **abstractie**
- Meerdere toepassingen kunnen gebruik maken van dezelfde programmatuur van de onderliggende laag

Dankzij de opdeling in lagen kan men bij het ontwerpen van cliënt- en serverprogramma's veronderstellen dat deze programma's berichten met mekaar uitwisselen (zie figuur 1.1) zonder rekening te houden met het fysiek netwerk (zie figuur 1.2). De communicatie in figuur 1.1 is bijgevolg **abstract** of **virtueel**. In werkelijkheid worden de berichten afgeleverd aan een tussenlaag of ontvangen van een tussenlaag. Een toepassing levert berichten af of ontvangt berichten van de bovenste van de tussenlagen. Elke laag bewerkt het bericht en levert een bericht (of meerdere berichten) af aan de laag er onder. Op het laagste niveau (fysieke laag) wordt het bericht echt overgebracht over het communicatiemedium. De functie van de tussenlagen kan enigszins vergeleken worden met die van een **transportbedrijf**: de gebruiker levert iets af en specifieert een bestemming. Het transportbedrijf doet de rest. Het transportbedrijf gebruikt het bestaande netwerk van auto-, spoor-, lucht- of waterwegen.

Het opdelen van de complexe netwerkaspecten in diverse tussenlagen heeft tal van voordelen, we illustreren er enkele aan de hand van een voorbeeld. In computernetwerken worden (lange) berichten van de toepassingslaag meestal niet als één geheel verstuurd maar opgedeeld in kleinere "stukken"die apart verstuurd worden. Als een toepassing een bestand file.txt moet versturen naar een andere machine dan is dit bestand file.txt het te versturen bericht op het niveau van de toepassingslaag. Door de tussenlagen zal het toepassingsbericht file.txt verdeeld worden in stukken. Zo



Figuur 1.2: Abstractielagen in een computernetwerk.

een stuk wordt meestal een **segment** genoemd. Er is een protocol dat alle afspraken voor de opdeling en wedersamenstelling van deze segmenten omvat. Hoe groot kan een segment zijn? Hoe worden segmenten genummerd? Wat als een segment niet correct toekomt? Dergelijke afspraken moeten toelaten dat (de programmatuur op) de ontvangende machine het oorspronkelijk bericht weer kan samenstellen uit de ontvangen segmenten.

Er zijn twee voor de hand liggende voordelen om grote berichten in segmenten op te delen:

- Omdat het ganse bericht groter is dan een segment, is de **kans** dat een ontvangen niet opgedeeld toepassingsbericht een **fout** bevat groter dan de kans dat een ontvangen segment van een toepassingsbericht een fout bevat. Als er een fout optreedt bij een segment moet alleen dat segment opnieuw verstuurd worden. Zou men een bericht niet opdelen, dan zou bij één foute bit, het ganse bestand opnieuw moeten verstuurd worden.
- Het netwerk kan segmenten van verschillende communicaties door mekaar versturen. Daardoor kunnen **meerdere applicaties of toepassingen** het netwerk tezelfdertijd gebruiken.

1.3 TCP/IP

Voor de lagen tussen toepassingslaag en fysieke laag wordt bijna altijd TCP/IP (transport control protocol/internet protocol) gebruikt. Eigenlijk is TCP/IP een geheel van diverse protocollen.

Enkele afspraken van deze protocollen zijn (later veel meer hierover):

- Elke computer wordt aangeduid met een uniek IP-adres. Een IP-adres (versie 4) bestaat uit vier getallen gescheiden door een punt. Elk van deze getallen ligt tussen 0 en 255, bijvoorbeeld 198.138.27.104. Dit is ook meteen de wijze waarop een IP-adres voorgesteld wordt
- Een proces, toepassing of applicatie op een computer wordt aangeduid d.m.v. een nummer. Dit nummer heet **poortnummer**. Voor ieder WWW-serverproces is het poortnummer 80. Dit nummer ligt vast. Ook het WWW-cliëntprogramma dat informatie opvraagt moet een poortnummer hebben. Op een computer kunnen immers meerdere cliëntprocessen actief zijn. Voor cliëntprocessen wordt geen vast nummer gebruikt. Er wordt willekeurig een (vrij) nummer gekozen, groter dan 1024.

1.3.1 TCP/IP toepassingen

Een cliëntprogramma kan niet zomaar om het even welk bericht sturen naar een serverprogramma. Eén en ander moet beantwoorden aan afspraken. Het server- en het cliëntprogramma zijn juist implementaties van deze afspraken. Voor elke toepassing is er een apart protocol. Enkele gekende toepassingen zijn: WWW, E-mail, bestandsoverdracht en terminaal-emulatie.

Al deze toepassingen hebben volgende eigenschappen gemeen: er zijn cliëntprogramma's en er zijn serverprogramma's, er wordt gecommuniceerd via TCP/IP en de berichten die uitgewisseld worden zijn vastgelegd in een protocol.

Per toepassing is er een ander protocol:

Toepassing	**Protocol**
WWW	HTTP, HTTPS
E-mail	POP3, IMAP, SMTP
Bestandsoverdracht	FTP, TFTP
Terminaal emulatie	SSH, Telnet

Een cliënt die een dienst wil, moet (het IP-adres van de computer waar het serverprogramma op draait en) het gepaste poortnummer van het serverprogramma opgeven. Er zijn zgn. alom gekende poortnummers. Hieronder enkele voorbeelden:

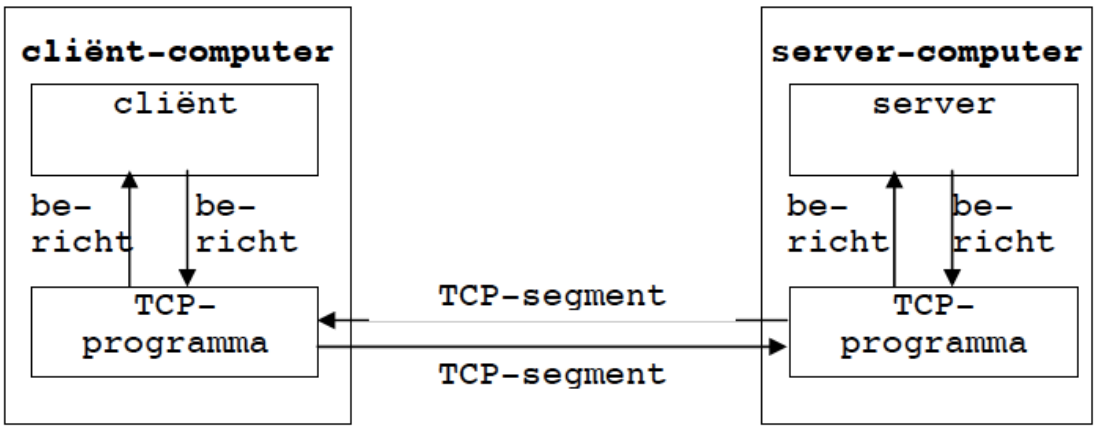
Protocol	Poortnummer
HTTP	80
SMTP	25
POP3	110
SSH	22
FTP	20 voor besturing en 21 voor data

Om serverberichten terug te sturen, moet de server ook (het IP-adres van de cliënt en) het poortnummer van het cliëntproces kennen. Als een cliënt een verbinding start met het serverprogramma, wordt een willekeurig nummer boven de 1024 als poortnummer gekozen. Bij elke nieuwe connectie of sessie kan dit een ander cliënt-poortnummer zijn. Het server-poortnummer blijft voor diezelfde applicatie hetzelfde.

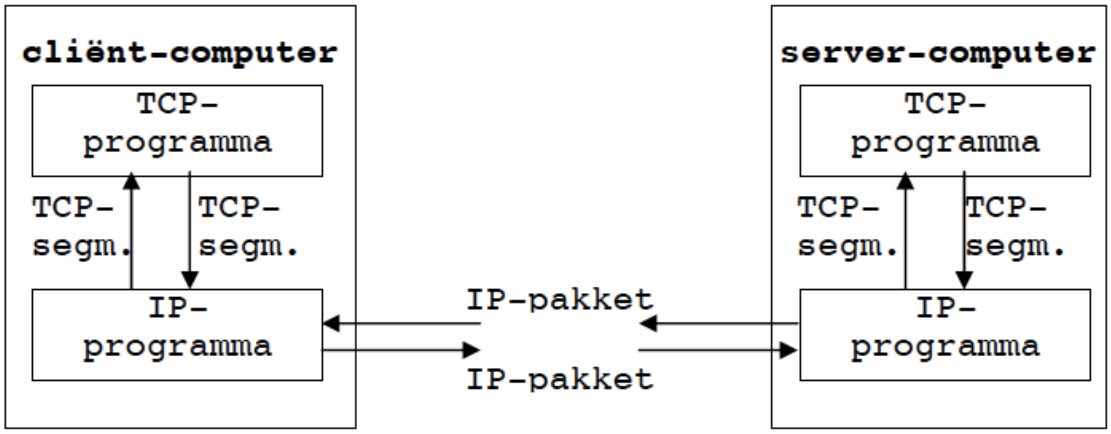
Internet RFC's Cliënt en server voeren protocollen uit die in een computernetwerk het volgende regelen:

- boodschappen en de volgorde ervan die worden verstuurd
- boodschappen en de volgorde ervan die worden ontvangen
- de structuur van de boodschappen

Om onafhankelijke en eventueel concurrerende bedrijven de mogelijkheid te geven om cliënts en/of servers te ontwikkelen om vervolgens deze met elkaar te laten communiceren over een computernetwerk, is het zeer belangrijk dat deze protocollen **publiekelijk** en **gestandaardiseerd** worden. Internetstandaarden worden ontwikkeld door de IETF. De beschrijving van deze standaarden wordt gedaan in een RFC. Wie een protocolspecificatie wil inkijken kan een RFC opvragen (request for comment), bvb. van FTP. Een mogelijke URL is ftp://ftp.isi.edu/in-notes/rfc1725.txt. RFC1725 gaat over POP. Wie de volledige specificatie van SMTP wil lezen, kan deze vinden in RFC821.
Protocollen zoals HTTP, SSH, FTP, SMTP, ... bevatten afspraken over de berichten die de toepassingsprogramma's kunnen uitwisselen. Met toepassingsprogramma's wordt bedoeld: een browser, een programma voor terminaal-emulatie, … of de verschillende serverprogramma's.



Figuur 1.3: Versturing van TCP-segmenten



Figuur 1.4: Versturing van IP-pakketten

1.4 TCP/IP-model van de tussenlagen

Zoals reeds vermeld, communiceren toepassingsprogramma's niet rechtstreeks met mekaar. Ze maken daarentegen gebruik van programmatuur op een lager niveau. Meestal is dit het **TCP-programma**. Dit is een implementatie van TCP, transmission control protocol. Het verstuurt en ontvangt berichten die beantwoorden aan het TCP. Deze berichten worden **TCP-segmenten** genoemd. Dit wordt voorgesteld in figuur 1.3.

De TCP-programmatuur wordt gebruikt door de toepassingsprogrammatuur en staat dus op een lager **niveau** of lagere **layer**. HTTP, FTP,... zijn protocollen voor de **toepassingslaag**. TCP is een protocol voor de **transportlaag** (een ander protocol voor de transportlaag is UDP). De TCP-programma's op verschillende machines wisselen niet rechtstreeks maar **virtueel** TCP- (of UDP-)segmenten uit. Er wordt weer gebruik gemaakt van programmatuur op een lagere laag: het **IP-programma**. Dit is een implementatie van het IP (internet protocol). De berichten die beantwoorden

aan het IP worden IP-pakketten genoemd. De laag waarop IP functioneert in figuur 1.4 wordt de **netwerklaag** genoemd.

Het IP-programma van de cliënt-computer communiceert niet rechtstreeks met het IP-programma van de server-computer maar virtueel. Wellicht zijn er op de fysieke verbinding tussen beide machines een aantal andere machines waarop de IP-programmatuur geïnstalleerd is. Deze tussenliggende machines zorgen er voor dat een verstuurd IP-pakket op zijn eindbestemming komt.

De IP-programma's maken gebruik van een lagere laag: de **datalinklaag**. Een voorbeeld van programmatuur in deze laag is het besturingsprogramma of driver voor de netwerkkaart. De berichten die in de datalinklaag uitgewisseld worden, zijn bijvoorbeeld Ethernet-**frames**.

Om frames van de ene machine naar de andere te sturen wordt gebruik gemaakt van de diensten van de **fysieke laag**. Deze laag bevat afspraken i.v.m. de voorstelling van 0 en 1 (bvb. welke spanningsniveaus). De apparatuur (netwerkkaart) genereert spanningen op de kabel en verstuurt aldus bits. De apparatuur in de andere machine herkent deze spanningen en zet deze terug om in bits.

1.5 OSI-referentiemodel van de tussenlagen

Afspraken voor datacommunicatie geven aldus eigenlijk aanleiding tot een **stapel (stack)** van afspraken. Vooraleer de TCP/IP-stapel gemeengoed geworden was, had de **ISO** (International Standards Organisation) de verschillende aspecten i.v.m. datacommunicatie ingedeeld in 7 groepen, **lagen** genoemd.
Het resultaat was het **OSI-referentiemodel** (Open Systems Interconnection Reference Model).

De zeven verschillende lagen zijn:

- toepassingslaag
- presentatielaag
- sessielaag
- transportlaag
- netwerklaag
- datalink-laag (ook dataverbindingslaag genoemd)
- fysieke laag

Het OSI-referentiemodel is een **referentiekader**:

- Het schrijft niet voor welke afspraken er moeten gemaakt worden, maar wel hoe men de afspraken moet indelen.
- Afspraken voor de ene laag mogen de afspraken voor een andere laag niet beïnvloeden.
- Met uitzondering van de fysieke laag worden deze lagen via programmatuur geïmplementeerd. Programmatuur in een bepaalde laag ”ziet”de programmatuur in de onderliggende laag als “subroutines”.
- Elke laag communiceert minimaal, op gestandaardiseerde wijze, met de laag er onder of er boven, en alleen met die twee lagen. Als men dus op de voorgeschreven wijze een component of programmatuur ontwikkelt voor één laag, dan volgt daaruit dat deze component compatibel is met alle lagen.

De **Fysieke Laag** heeft betrekking op alles wat nodig is om de data fysiek over een netwerk te transporteren, inclusief de bekabelingsmethodes, maar niet de bekabeling zelf. Hoeveel pinnen, hoe een 0-bit of 1-bit er uitziet, hoe lang een bit duurt, … .

De **Datalink-laag** heeft betrekking op de datatransmissie en afspraken om fouten te corrigeren, of te melden. Ook afspraken over de wijze waarop bits in pakketjes gebundeld en weer uitgepakt worden, ...

De **Netwerklaag** heeft betrekking op het transport, de adressering en de routering van pakketten doorheen het netwerk als ook op het opzetten van een route van bron naar eindbestemming.

De **Transportlaag** heeft betrekking op afspraken om een boodschap zonder fouten van bron naar eindbestemming te sturen, maar op een hoger niveau dan de datalink-laag (omdat er meerdere tussenstations kunnen zijn en omdat pakketten kunnen opgesplitst worden).

De **Sessielaag** heeft betrekking op de communicatie tussen twee toepassingsprocessen.

De **Presentatielaag** heeft betrekking op afspraken voor het coderen en decoderen van gegevens voor de applicatielaag. Bv. data-encryptie en omzetting van formaten gebeuren in deze laag.

De **Toepassingslaag** zou men het operating system kunnen noemen met de netwerktoepassingen.

Als een netwerktoepassing niet werkt, en als u niet direct de oorzaak ziet, begin dan bij de onderste laag van het OSI-model:
- is de bekabeling in orde (is de kabel aangesloten, is het de juiste kabel, ...)?
- is het goede kaartbesturingsprogramma geïnstalleerd?
- klopt de IP-configuratie?
- enz...

1.6 TCP/IP-model versus OSI-referentiemodel

Het OSI-model is een theoretisch model. Het TCP/IP-model daarentegen bestaat echt.
Bij TCP/IP wordt er maar met 5 lagen gewerkt:
- Toepassingslaag
- Transportlaag
- Internetlaag
- Datalinklaag
- Fysieke laag

Soms kom je een 4 lagen model van TCP/IP tegen, dan zijn de laatste twee lagen samengevoegd tot een netwerktoegangslaag.

Grosso modo kan men stellen:
- De afspraken van de bovenste 3 OSI-lagen (toepassing, presentatie, sessie) vormen de TCP/IP-toepassingslaag;
- De transportlagen van OSI en TCP/IP zijn dezelfde;
- De netwerklaag van OSI en de internetlaag van TCP/IP zijn dezelfde;
- De afspraken van de onderste 2 OSI-lagen (datalink, fysiek) vormen de samengevoegde TCP/IP-netwerktoegangslaag.

Hoofdstuk 2

Datalinklaag: Netwerken en LANs

In dit hoofdstuk bestuderen we de protocollen, algoritmen en afspraken voor een communicatie van complete informatie-eenheden ook wel frames genaamd, tussen twee rechtstreeks met elkaar communicerende computers. In de fysieke laag worden afzonderlijke bits (of een groep van bits) verstuurd, in de datalinklaag worden frames verstuurd.

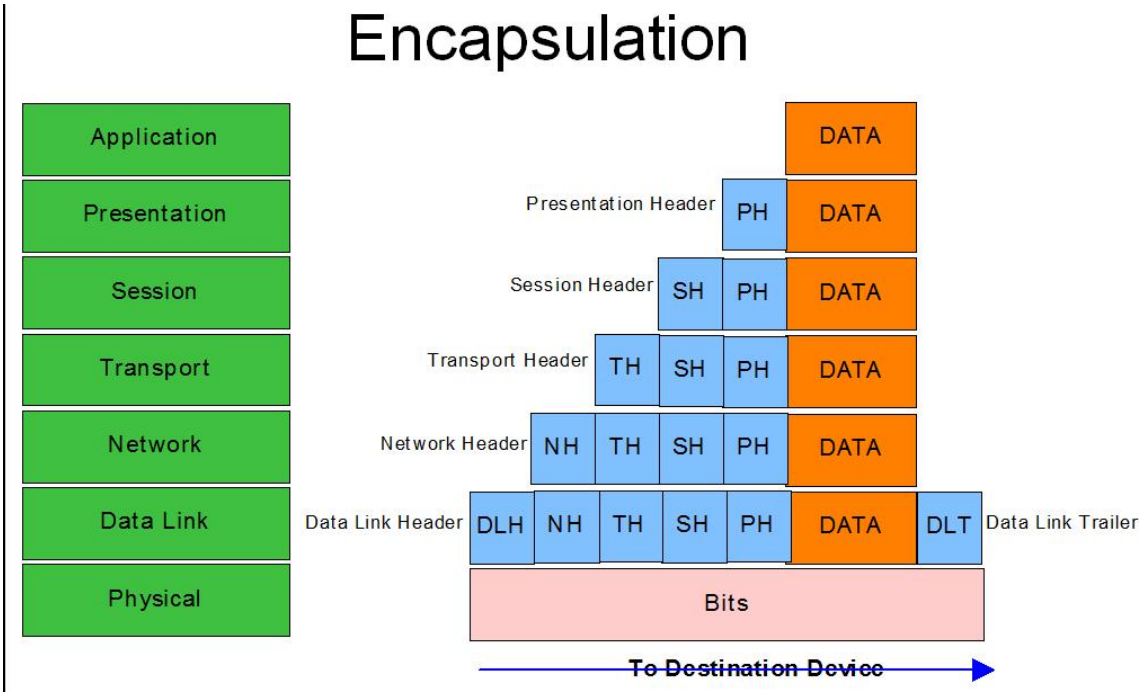
2.1 Terminologie: Link, node, netwerk en PDU

Een apparaat dat informatie uitwisselt over een computernetwerk wordt **node** genoemd. Het communicatiekanaal tussen aangrenzende nodes noemen we algemeen een **link**. Deze links kunnen op hun beurt onderverdeeld worden in broadcast- en point-to-point-links.

- Broadcastlinks: wanneer meerdere nodes verbonden zijn door middel van een gedeelde broadcastlink zal er een medium-access-protocol[1] nodig zijn dat de toegang tot de link en het verzenden van frames op deze link in goede banen zal leiden.
- Point-to-pointcommunicatielink: deze wordt vaak gebruikt tussen twee nodes die rechtstreeks met elkaar communiceren.

Twee of meerdere nodes die geconnecteerd worden door middel van een gedeeld medium/link en daardoor rechtstreeks met elkaar kunnen communiceren, wordt een **netwerk** genoemd. De data of het geheel van informatie dat verstuurd wordt tussen twee nodes en gespecifieerd wordt in een protocol wordt **Protocol Data Unit of PDU** genoemd.

[1] Vaak ook Multiple-access-protocol genoemd.



Figuur 2.1: De relatie tussen de PDU's op de verschillende OSI-lagen en encapsulatie. Bron: www.hackplayers.com

2.2 Diensten van de datalinklaag

De primaire functie van protocollen in de datalinklaag is ervoor zorgen dat data over een gemeenschappelijke medium of link uitgewisseld worden. De datalinklaag[2], die gebruik maakt van de services van de fysieke laag om bits over een communicatiekanaal te verzenden en te ontvangen, voorziet in drie basisdiensten[3]:

- Een duidelijk interface (API) met de netwerklaag voorzien en frames vormen.
- Frames versturen d.m.v. media access control: het coördineren van de verzending/ontvangst van frames door verschillende nodes over een gemeenschappelijke link.
- Frames ontvangen en foutdetectie: het is zinloos om frames of PDUs die fouten bevatten te verwerken of verder door te sturen.

Om deze diensten te realiseren ontvangt de datalinklaag een pakket van de netwerklaag en verpakt of encapsuleert het in een frame voor verzending, zie figuur 2.1.

[2]Wanneer een specifieke laag in het OSI- of TCP/IP-model wordt genoemd, wordt er vaak meer exact de protocollen in deze respectievelijke laag bedoeld.

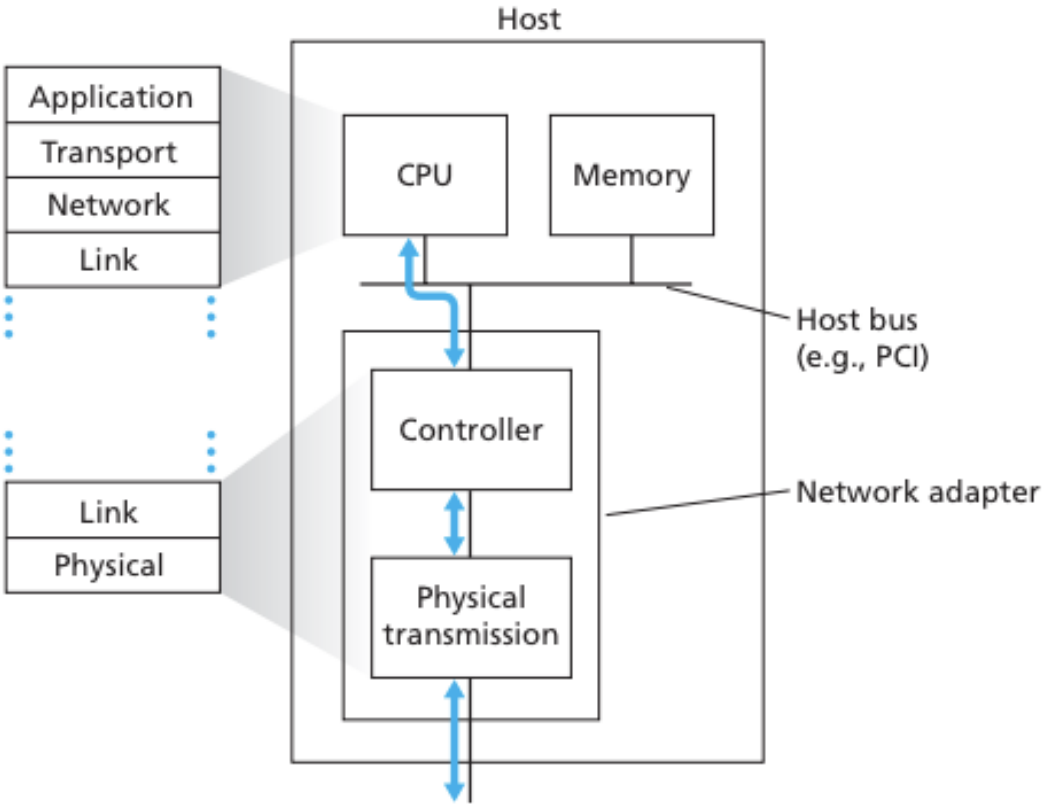
[3]Het reguleren van de stroom van frames tussen zender en ontvanger, nl. flowcontrol is ook een dienst die deze laag levert.

Een frame bevat steeds volgende elementen:

- frameheader
- framepayload, met hierin het pakket van de hoger gelegen netwerklaag.
- frametrailer ook wel framefooter genoemd.

2.3 Implementatie van de datalinklaag

Wordt de datalinklaag in een host of node geïmplementeerd in hardware of sotware? In figuur 2.2 wordt duidelijk dat veel van de datalinklaagfunctionaliteiten in de netwerkkaart of network interface card (NIC) draait: hardware dus. Tot begin deze eeuw waren de meeste NICs afzonderlijke kaarten. Tegenwoordig worden meer en meer NICs geïntegreerd op het moederbord.



Figuur 2.2: De relatie tussen de netwerkadapter (NIC) en de rest van het besturingssysteem. Bron: Computernetwerken, een top-downbenadering

In figuur 2.2 is ook te zien dat een deel van de datalinklaag geïmplementeerd wordt in software. De softwarecomponent van de datalinklaag is de driver

van je NIC. Deze handelt adresinformatie en interruptafhandeling van de NIC hardware af.
De datalinklaag is dus een combinatie van hard- en software. Dé plaats in de protocolstack die interfacet tussen hardware en software [1]. Het merendeel van de datalinklaag wordt in de NIC geïmplementeerd, zie figuur 2.2.

2.4 Multiple-accesslinks en -protocollen

Hoe de verschillende computers toegang krijgen tot dezelfde fysieke drager/link (bvb. Coaxkabel), is een belangrijke vraag die gesteld moet worden bij computernetwerken waaronder LANs. Hiervoor bestaan er verschillende multiple-access-protocollen die voor de datalinklaag ontwikkeld zijn. Ze worden onderverdeeld in drie types:

- kanaalpartitioneringsprotocollen: dit type zal niet worden besproken in deze inleidende cursus. Geïnteresseerden worden doorverwezen naar [2].
- willekeurige-toegangsprotocollen: de computer die wil zenden, wacht tot de link (of drager) vrij is en begint te zenden. Botsingen zullen gebeuren wanneer twee of meerdere nodes tegelijkertijd frames verzenden. Na een botsing wordt de toestand hersteld. Deze toegangsmethode wordt ook "**toegang door contentie**"genoemd (contentie = naijver). Ethernet is hier hét voorbeeld van.
- deterministische toegangsprotocollen: een computer wacht tot hij toegang krijgt. Als een computer toegang krijgt, dan is er zekerheid dat andere computers niet zullen storen. Hierbij wordt dikwijls gebruik gemaakt van een zgn. **token**. Token betekent hier: bewijs van toegangsrecht. Er is één token. Wie wil zenden moet eerst het token hebben. Wie het token heeft, mag zenden. De tokenhouder moet dit token uiteraard ook weer afstaan.

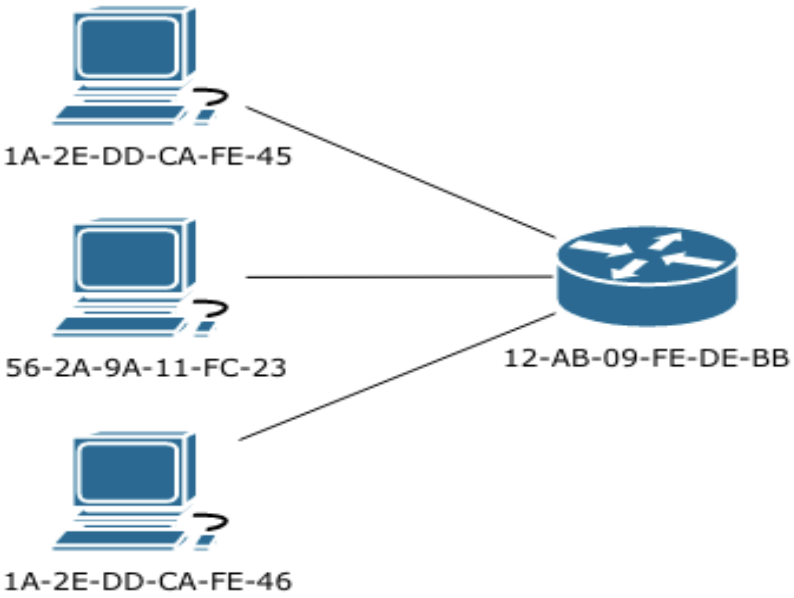
Door het IEEE (Institute of Electric and Electronic Engineers) zijn er verschillende types LAN's gestandaardiseerd.
- IEEE802.3: Ethernet (oorspronkelijk van Xerox),
- IEEE802.4: token bus,
- IEEE802.5: token ring (IBM).

In deze tekst beperken we ons tot Ethernet, omdat deze technologie tegenwoordig de meest gebruikte is.

2.5 Ethernet: fysiek

In een hedendaagse pc is de Ethernet-apparatuur meestal geïntegreerd op het moederbord. In dit geval heeft de pc een Ethernet-connector. Is dit niet zo, dan kan in de pc een Ethernetkaart geïnstalleerd worden. Verder in deze tekst gebruiken we de term Ethernetkaart voor beide gevallen:
- een echte ingeplugde kaart;
- Ethernet geïntegreerd op het moederbord.

Elke Ethernetkaart heeft een **uniek adres**. Het bestaat uit **48 bits** of **12 hexadecimale cijfers**, bvb. 00 0B DB 3F FA 77. Men noemt dit adres: het MAC-adres (medium access control), fysiek adres of een LAN-adres. Zoals je kan zien in figuur 2.3 worden MAC-adressen geschreven in hexadecimale notatie waarbij een byte wordt geschreven als een paar hexadecimale getallen.

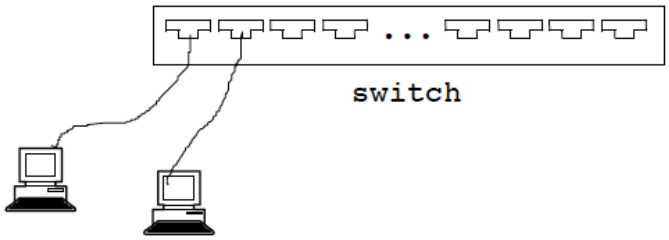


Figuur 2.3: Elke NIC heeft een uniek MAC-adres

Oefening Hoeveel Ethernetkaarten kunnen er wereldwijd zijn? Hoe zouden fabrikanten er voor kunnen zorgen dat er geen twee kaarten zijn met toevallig hetzelfde adres?

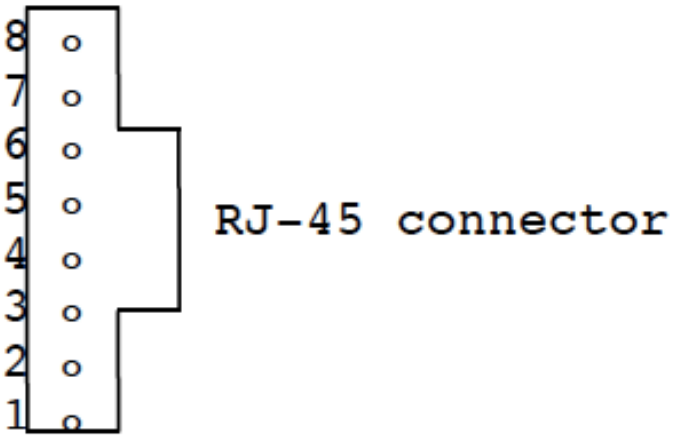
Op de Ethernetkaarten van de pc's wordt een kabel aangesloten. Bij de museum-versie van Ethernet werden trancievers en een dikke coaxiale kabel gebruikt. Later werd dunne coax als bekabeling gebruikt, maar tegenwoordig wordt gebruik gemaakt van UTP-bekabeling (unshielded twisted pair;

zie verder). Er bestaat ook STP (shielded TP) en FTP (foiled TP). Vaak heeft men het gewoon over TP.
Via een TP-kabel wordt elke pc verbonden met een centraal apparaat dat switch genoemd wordt. (I.p.v. een switch wordt soms een hub gebruikt; het verschil wordt later uiteengezet). Logisch hebben we dus een ster-structuur of ”star-topology”.



Een switch bevat bvb. 4, 8, 12, ... poorten. Poort betekent hier: een aansluiting voor TP. De connector heet RJ45-connector (recommended jack). Via de switch en de kabels is elke pc fysiek verbonden met elke andere pc.

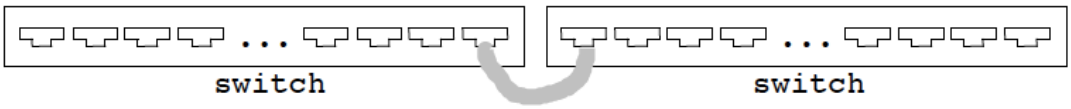
De TP-kabel bevat 8 draden. Voor Ethernet met een bitsnelheid tot 100 Mbit/s worden er hiervan 4 gebruikt. Hoe de draden worden gebruikt kan teruggevonden worden in tabel 2.1.



Draad nummer	Functie
8:	
7:	
6:	Ontvangen -
5:	
4:	
3:	Ontvangen +
2:	Verzenden -
1:	Verzenden +

Tabel 2.1: Draden in een RJ45

Het aantal poorten op een switch is beperkt maar switches kunnen als volgt in cascade geschakeld worden:



Figuur 2.4: Cascade switches

Zoals hoger vermeld, worden draden 1 en 2 gebruikt om te zenden en draden 3 en 6 om te ontvangen. Wat de ene zendt, ontvangt de andere. Bijgevolg moet er tussen 2 pc's van een netwerk een **omwisseling** gebeuren. Men kan 2 pc's al met mekaar laten communiceren door de Ethernetkaarten te verbinden met een gekruiste kabel (E. : crossover cable). In dit geval moet er dus geen extra apparaat nl. switch gebruikt worden. De omwisseling is als volgt: 1-3, 3-1, 6-2, 2-6.

Om meer dan 2 pc's te laten communiceren, wordt zoals reeds vermeld een switch gebruikt. In een switch gebeurt deze omwisseling. Wat door één pc op draden 1 en 2 gestuurd wordt, wordt door de switch verder gestuurd via draden 3 en 6. Pc's worden daarom met een rechte kabel (E. : straight-through cable) met een switch verbonden.

Als 2 switches in cascade geschakeld worden, moeten de signalen tussen de switches ook omgewisseld worden, vermits beide switches omwisselen. Er zijn diverse mogelijke oplossingen:

- De switches verbinden met een gekruiste kabel.
- Sommige switches hebben één dubbele poort, d.w.z. met 2 aansluitpunten. In één van de twee aansluitpunten gebeurt een omwisseling. Het kan gebruikt worden om een pc aan te sluiten via een rechte kabel. In het andere aansluitpunt gebeurt geen omwisseling. Het dient

om een andere switch aan te sluiten met een rechte kabel. Beide aansluitpunten kunnen NIET tezelfdertijd gebruikt worden. In plaats van een dubbele poort kan ook gebruik gemaakt worden van één poort die met een schakelaar kan ingesteld worden.

Als men ergens een rechte kabel gebruikt waar het een gekruiste had moeten zijn (of andersom), heeft men gegarandeerd een netwerkprobleem. De documentatie van de fabrikant doornemen is essentieel. Overigens ziet de toekomst er goed uit: voor sommige switches maakt het niets meer uit, de poorten passen zich automatisch aan.

2.6 Ethernet: frames

Via Ethernet kunnen pc's Ethernet-frames naar mekaar versturen en van mekaar ontvangen. Een Ethernet-frame is een bitrij en bevat o.m.:

- adres van de bestemmeling (het MAC-adres van 6 bytes)
- adres van de zender (het MAC-adres van 6 bytes)
- lengte of type (2 bytes)
- data (max. 1500 bytes, ook maximum transmission unit (MTU) genoemd)
- (eventueel) opvulbytes (de lengte van een frame moet minstens 64 bytes bedragen)
- controle-informatie (4 bytes)

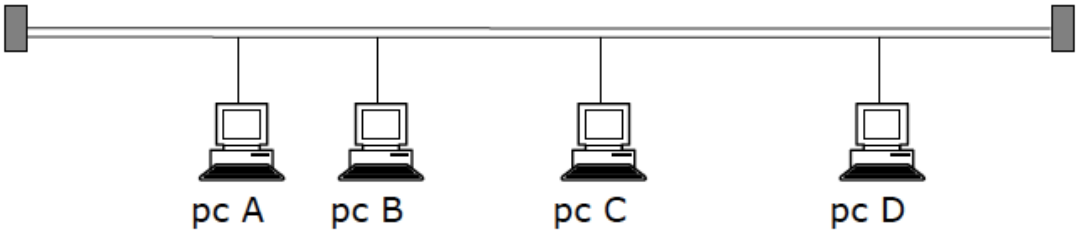
De totale lengte van een frame is dus ten hoogste 1518 bytes en ten minste 64 bytes. De data kan bvb. een deel van een bestand zijn (als de toepassing bvb. een bestandsoverdracht zoals FTP is) maar kan ook bestaan uit informatie die moet verstuurd worden om de werking van het netwerk te verzekeren, zoals OSPF-routing updates.

Om de Ethernetkaart te kunnen gebruiken, moet een kaartbesturingsprogramma (driver) geïnstalleerd zijn. Het is dit programma dat een frame verstuurt en ontvangt. We zien hier één van de verschilpunten tussen fysieke laag en de datalinklaag: op de fysieke laag bestaan bits. Bits groeperen tot frames, frames versturen en ontvangen gebeurt in de datalinklaag.

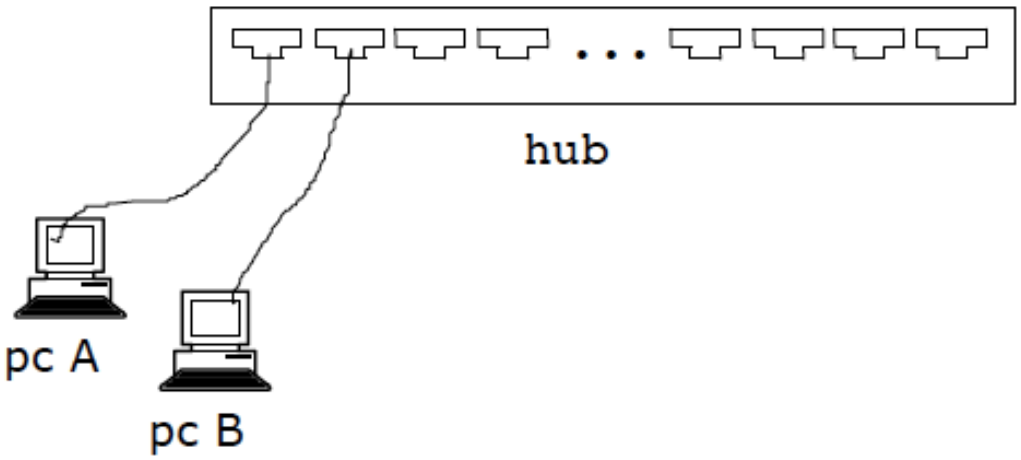
Preambule Aan een Ethernet frame gaat een zgn. preambule (7 bytes) en een SFD (start of frame delimiter, 1 byte) vooraf. Preambule en SFD vormen een speciaal bitpatroon en geven aan dat er een frame volgt.

2.7 Ethernet a.k.a. CSMA/CD

Bij het versturen van frames moeten de pc's (kaartbesturingsprogramma's) zich houden aan afspraken. Het protocol heet CSMA/CD. Voor een goed begrip van dit protocol moeten we even de Ethernet-geschiedenis bekijken (vaak is kennis van het verleden een vereiste om het heden te kunnen begrijpen), Ethernet m.b.v. coax-kabels.
Bij CSMA/CD wordt één frame dat door één pc verstuurd wordt, naar alle andere pc's in het netwerk gestuurd. Logisch is het netwerk dus een bus- (eigenlijk: broadcast-)netwerk. Vroeger was het netwerk fysiek ook een bus. Er was één coaxiale kabel die doorheen het gebouw liep en elke pc was op deze kabel aangesloten, zoals op onderstaande figuur weergegeven. Wie lang zoekt, treft misschien nog een dergelijk netwerk aan.



Het is duidelijk dat elk signaal dat (elke bit die) een pc op de kabel zet voorbij elke andere pc passeert. Op beide uiteinden van de kabel bevinden zich zgn. afsluitweerstanden (terminaties genoemd). Deze absorberen aan het uiteinde alle signalen en voorkómen aldus dat signalen weerkaatst worden.
In een later stadium werd de coaxiale kabel vervangen door één centraal apparaat, hub (NL. "naaf") genoemd. De pc's zijn via een TP-kabel verbonden met deze hub.



Deze figuur lijkt goed op die van een switch maar er is één belangrijk verschil: het centrale apparaat is een hub en geen switch. Een hub zal elk signaal dat ontvangen wordt, op alle poorten verder sturen, behalve op deze langs waar het is binnengekomen. Net zoals bij het gebruik van coaxiale kabel zal elk frame dat door een kaart verstuurd wordt, bij elke andere pc terecht komen. Om deze reden wordt Ethernet vaak als bus getekend zelfs al is het fysiek een sternetwerk.

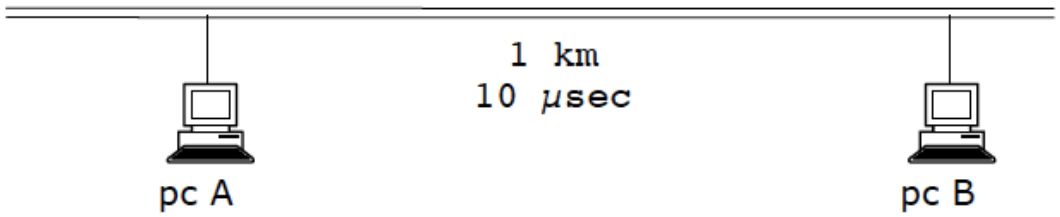
Ethernet-principe Een Ethernetkaart plaatst een frame op de kabel. Via de bekabeling passeert dit frame voorbij elke andere Ethernetkaart. Alleen de kaart waarvoor het frame bestemd is, mag het frame lezen. Hiertoe leest elke kaart het bestemmingsadres van het ontvangen frame. Als een kaart vaststelt:

- dit is mijn adres
- dit is een broadcast-adres
- dit is een multicast-adres voor mijn groep

dan wordt het frame gelezen. Een Ethernet broadcast-adres bestaat uit 12 F'en en wordt gebruikt om een frame naar alle netwerkkaarten in het netwerk te sturen. Een multicast-adres is een afgesproken adres voor een groep van Ethernetkaarten.
De afspraken (het protocol) die de Ethernetkaarten moeten naleven, worden aangeduid met de term CSMA/CD.

MA staat voor multiple access; meerdere knooppunten hebben tezelfdertijd toegang tot de gegevensdrager of link.
CS betekent carrier sensing, (carrier = (gegevens)-drager). Als een Ethernetkaart een frame op de kabel plaatst, terwijl er al een ander frame op de kabel is, dan is er een botsing (collision). Om botsingen te voorkomen moet een kaart eerst nagaan (to sense) of de kabel (carrier) vrij is. Indien niet, dan moet de kaart wachten en even later opnieuw proberen.
CD betekent collision detection. Er kunnen botsingen voorkomen. De kaarten moeten de kabel inspecteren om na te gaan (te detecteren) of er een botsing is. Als er een botsing is, dan zijn de frames verloren. Als een kaart vaststelt dat haar frame botst, stuurt ze gedurende een zekere tijd het zgn. jam-signaal. Dit garandeert dat elke kaart op het netwerk weet dat er een botsing was. Na een botsing wachten de zendende kaarten een lukraak gekozen tijdsinterval vooraleer ze opnieuw nagaan of de kabel vrij is.
Dat bij carrier sensing toch botsingen kunnen voorkomen, blijkt uit volgend belangrijk voorbeeld.

Onderstel dat pc's A en B, 1 km van mekaar verwijderd zijn. De tijd die de eerste bit van een frame nodig heeft om van A naar B te reizen is bvb.



10 μs (bits op een draad zijn eigenlijk elektrische signalen; deze reizen ongeveer aan 20 cm/ns of ongeveer 2/3 van de lichtsnelheid maar er moet ook rekening gehouden worden met vertragingen, bvb. veroorzaakt door hubs of switches).
Als A een frame op de bus plaatst, dan duurt het 10 μs vooraleer dit frame B kan bereiken. Als B 9,9 μs nadat A begonnen is, ook een frame wil versturen, zal B vaststellen dat de bus vrij is. B zal dan ook een frame op de bus plaatsen. Na 0,05 μs doet de botsing zich voor. Na nog eens 0,05 μs zal B hiervan op de hoogte zijn. Bemerk dat A pas 20 μs (19,9 μs om exact te zijn) na het versturen van de eerste bit van zijn frame, zal vaststellen dat er een botsing gebeurd is.
Als er zich een botsing voordoet dan weet de zender dit reeds tijdens het versturen van het frame, omdat een frame altijd uit minstens 64 bytes bestaat. De tijd nodig om 64 bytes te versturen is lang genoeg om te weten of het frame botst of niet. De lengte van de Ethernet-kabel en de snelheid waarmee bits verstuurd worden, spelen hierbij een rol. Als frames botsen, zijn deze frames verloren. Een restant van een gebotst frame wordt **runt** genoemd.
Nadat de 64° byte van een frame verstuurd is, kan het frame normaal (volgens het protocol) niet meer botsen. Soms gebeurt dit toch. Men noemt een dergelijke botsing een late collision. Een late collision is erger dan een gewone botsing omdat ze wellicht niet opgemerkt wordt door de zendende kaart. Het frame wordt dus niet onmiddellijk herverstuurd. Een programma op een hoger niveau (hogere OSI-laag) zal het probleem moeten oplossen.

2.8 Snelheid

In computernetwerken moeten we ons idee van snelheid enigszins herzien. Met snelheid bedoelt men de capaciteit of bitsnelheid, d.i. het aantal bits dat per seconde kan verstuurd worden. De bitsnelheid wordt ook wel eens ”(digitale) bandbreedte”genoemd. Deze bandbreedte heeft niets te maken met de snelheid waarmee de bits reizen, want dit is zowat 2/3 van de lichtsnelheid. De bandbreedte heeft alles te maken met de snelheid waarmee de bits mekaar mogen opvolgen om zonder fouten onderscheiden te kunnen worden bij de ontvanger. Ter vergelijking: als iemand lichtsignalen geeft, zien wij die ogenblikkelijk. Als een robot lichtsignalen te snel na mekaar

zou geven, dan zouden wij ze niet meer kunnen onderscheiden.
Vroeger (bij coaxiale bekabeling ook 10BASE2 of eerste TP-bekabeling ook 10BASE-T genoemd) was de bitsnelheid 10Mbit/s. De coaxiale kabel mocht tot 2500m lang zijn (hiervoor waren dan wel versterkers nodig).

Als:

- de kabel niet langer is dan 2500m (5 km in 50 μs)
- een frame uit minstens 64 bytes bestaat
- de bitsnelheid 10Mbit/sec bedraagt

dan kan:

- een frame van de ene kant van de kabel naar de andere kant reizen
- dit frame daar botsen
- het signaal dat door deze botsing ontstaat, de zender weer bereiken, vooraleer de zender het volledige frame van minstens 64 bytes verstuurd heeft

Ontwikkel-regel bij CSMA/CD: Als een kaart een frame op de bus plaatst, gaat ze na of dit frame botst. Als de eerste 64 bytes van het frame volledig op de bus kunnen geplaatst worden zonder dat de kaart vaststelt dat er een botsing is, dan is het zeker dat dit frame niet meer kan botsen.

2.9 Switch

Fundamenteel aan het CSMA/CD-protocol is dat er op elk ogenblik niet meer dan één pc gegevens verstuurt, anders is er een botsing. Het kan "veel beter" door een intelligente hub te gebruiken die de botsingen "vermindert". Een dergelijke hub wordt switch (of schakelaar) genoemd. De werking ervan is als volgt: als een Ethernet-frame in de switch aankomt, leest de switch eerst het adres van de bestemmeling. Het frame wordt dan verstuurd, alleen op de poort waarop de bestemmeling zich bevindt. Een hub daarentegen leest geen adressen en stuurt elk inkomend signaal (frame) verder op elke poort (behalve op die waarop het frame is binnengekomen). Eigenlijk is een hub alleen een elektrisch apparaat: het herkent een bit (elektrische spanning), regenereert deze en stuurt hem verder op al zijn poorten (behalve deze waarop het frame is binnengekomen). Een switch daarentegen kent Ethernet, kan een frame analyseren en het adres herkennen. (In OSI-taal : een hub zit in de fysieke laag, een switch zit in de datalinklaag.) Een hub wordt daarom ook wel eens multiport repeater genoemd. Als gebruik gemaakt wordt van een switch spreekt men over **geschakeld Ethernet** (switched Ethernet); bij een hub heeft men het over **gedeeld Ethernet** (shared Ethernet).

Hoe weet een switch welke Ethernet-adressen op elk van zijn poorten voorkomen? Een switch leert dat zelf en hoeft hiervoor niet geconfigureerd te worden. In het begin gedraagt de switch zich als een hub. Iedere keer dat er een frame via een poort binnenkomt, slaat de switch het poortnummer en het Ethernet-adres van de zender op. Aldus weet de switch na korte tijd welke Ethernet-adressen op elk van zijn poorten voorkomen. Een frame met een bestemmingsadres dat de switch nog niet kent, wordt noodgedwongen op alle poorten verder gestuurd.

Dankzij het gebruik van switches zijn botsingen uitzonderlijk. Ook wordt het zgn. sniffen op netwerken hiermee aan banden gelegd (een pc is aan het sniffen als een slechterik zijn netwerkkaart alle frames doet lezen).

Opdat een switch zijn taken zou kunnen vervullen, moet hij uitgerust zijn met een processor en werkgeheugen (RAM). De processor is een voorbeeld van een ASIC, (application specific ic), d.w.z. een ic ontworpen voor een specifiek doel (in casu : Ethernet-frames schakelen). Een niet-ASIC is bvb. een Pentium-processor. Deze is ontworpen voor meerdere meer algemene taken (rekenen, tekstverwerking, databanken bijhouden, …).
Het geheugen van een switch werkt anders dan dat van een pc. Het is niet zo dat de toegang tot het geheugen gebeurt in de vorm van ”lees (schrijf) het dubbelwoord met adres 00A2C350”zoals in de cursus Computersystemen uiteengezet. Het geheugen is daarentegen inhoud-adresseerbaar. De opgeslagen info bestaat uit items als:

AB 01 34 00 23 A6 : 1
12 34 45 A9 26 FD : 12
AB 01 34 A6 20 78 : 1
12 34 56 A3 B4 CC : 8
...

d.i. : ”mac-adres : switch-poort waarop dit adres voorkomt”. Vermits aan een switch-poort een hub of een andere switch kan hangen, kunnen er meerdere mac-adressen gelinkt zijn met hetzelfde poortnummer.
Als een frame met bestemmings-mac-adres bvb. 12 34 45 A9 26 FD in de switch aankomt, moet het geheugen snel het poortnummer kunnen leveren waarop dit mac-adres voorkomt (12 in het voorbeeld). Als het mac-adres nog niet voorkomt, moet het geheugen ook deze informatie snel kunnen leveren. Gewoon alle items overlopen tot het goede gevonden is, zou te traag gaan. De toegang tot het geheugen gebeurt dus niet door het adres van de geheugenplaats op te geven maar door de inhoud (het mac-adres) op te geven. Vandaar de naam: C.A.M.-geheugen, d.i. content addressable memory of inhoud-adresseerbaar geheugen. Dit is het principe van de (volledig) associatieve cache zoals uiteengezet in de cursus computersystemen

en besturingssystemen. De toegang tot een dergelijk geheugen gebeurt via een bevel als: ”geef mij de inhoud van een geheugen-item waar in een ander geheugen-item 12 34 45 A9 26 FD staat”.

Segment Sporadisch wordt nog de term Ethernet-segment gebruikt. Wat is een segment? In de tijd van coax was het antwoord op deze vraag duidelijk: een segment was een stuk coax dat aan elk uiteinde afgesloten was door een terminatie of door een repeater. Nu blijkt er geen eenduidige definitie meer te zijn. Toch een poging: een hub met een aantal aangesloten TP-kabels kan beschouwd worden als een samengeknepen stuk coax (E. collapsed backbone). Een TP-kabel die 2 hubs verbindt, heeft dezelfde functie als een stuk coax dat destijds 2 repeaters verbond. Dus een Ethernet-segment is:
- een stuk coax (met op elk uiteinde een terminatie of een repeater);
- of, een hub met TP-kabels waarop alleen computers aangesloten zijn;
- of, een TP-kabel die 2 hubs verbindt.

Microsegment Eén kabel die één pc verbindt met een **switch** (dus geen hub, maar een switch) wordt microsegment genoemd en moet dus ook als een segment beschouwd worden.

Frame-behandeling Er zijn meerdere manieren waarop een switch omspringt met de frames die hij moet versturen:

- **Cut-through switching**: van zodra de switch de eerste bytes van het frame met het bestemmingsadres gelezen heeft, zal de switch het frame verder[4] sturen op de poort waarop de bestemmeling zich bevindt
- **Store-and-forward switching**: de switch leest eerst het ganse frame, controleert op fouten (via de CRC, zie verder). Als er geen fouten zijn, wordt het frame verstuurd
- **Fragment-free switching**: pas nadat de switch de eerste 64 bytes gelezen heeft, stuurt hij het frame verder

Vraag: welke van bovenstaande switches zorgt ervoor dat het verbonden netwerk botsingsvrij is, waardoor full-duplex communicatie over één TP mogelijk wordt?

[4]Zoals hoger opgemerkt, kunnen Ethernet-segmenten van verschillende snelheid met mekaar verbonden worden. Cut-through switching is alleen mogelijk tussen 2 poorten die aan dezelfde snelheid werken omdat dergelijke switch geen buffergeheugen heeft. Is de snelheid niet dezelfde, dan moet het frame opgeslagen worden. Switchen tussen Ethernet-segmenten van verschillende snelheid, heet asymmetrisch switchen; is de snelheid dezelfde, dan is het symmetrisch switchen.

2.9.1 Virtuele Switch (VLAN)

Een goedkope switch met bvb. 8 poorten is een netwerk-apparaat met een meer dan gunstige prijs/kwaliteit-verhouding als alleen connectiviteit vereist is. Dit vereist dat frames op de goede poort verder gestuurd worden zonder botsingen.
Een switch brengt eigenlijk een punt-tot-punt verbinding tot stand tussen 2 computers en afluisteren wordt onmogelijk. Het lijkt alsof er tussen 2 communicerende computers een draad geschakeld is (vandaar de naam ”switch”).
Blijven evenwel de Ethernet broadcast-frames: via sniffing kan men deze afluisteren. En niet alleen afluisteren: via een broadcast wordt vaak een vraag gesteld aan alle verbonden computers. Iemand die afluistert, kan snel de vraag beantwoorden en aldus foute informatie verspreiden. Een 1° stap naar hacken.
Een goedkope switch heeft niet de mogelijkheid tot configuratie. Een duurdere wel. En er is nogal wat dat kan geconfigureerd worden. Zo kan een netwerkbeheerder een switchpoort zonder meer afsluiten of kunnen er zgn. vlan’s (virtuele LAN’s) geconfigureerd worden. Als volgt bijvoorbeeld:

Poorten	Status/VLAN
2, 3, 5 ,6	Management
4, 8, 12	Verkoop
1, 11, 16	Disabled

of
* Poorten 1, 2, 3 : vlan ”beheer”;
* Poorten 4, 5, 6, 7, 8, 9, 10 : vlan ädministratie”;
* Poorten 11 t/m 24 : vlan ”productie”.

Broadcasts worden beperkt tot het vlan. Dus een broadcast die binnenkomt op poort 3 wordt alleen op poorten 1 en 2 verder gestuurd. Men kan alleen nog broadcasts afluisteren op het eigen vlan.

2.10 Soorten Ethernet

Het oude Ethernet met 10 Mbit/s zal intussen wel volledig verdwenen zijn. De opvolger was ”fast Ethernet”.

2.10.1 Fast Ethernet

Enkele eigenschappen van fast Ethernet :

- Ook CSMA/CD als protocol
- 100 Mbit/s (tegen 10 Mbit/s voor het oude Ethernet)

- Niet via coax
- Kon zich perfect aanpassen (samenwerken met) aan het oude Ethernet (en heeft zich wellicht daarom razendsnel verspreid)
- 10 maal sneller; daardoor: lengte 10 maal korter, max. 250m. De maximale afstand wordt overigens nog meer beperkt wegens de eigenschappen van TP. Een TP-verbinding (tussen pc en hub/switch) mag niet langer zijn dan 100m.

  Intussen is ook fast Ethernet bijna uit circulatie. We zijn toe aan Gigabit Ethernet: 1Gbit/s

2.10.2 Gigabit Ethernet

Bij Gigabit-Ethernet worden de bits verstuurd aan 1000Mbit/s. Het CSMA/CD principe blijft gelden: een netwerkkaart luistert of er een botsing is terwijl ze bits op het netwerk zet. Eens een volledig frame verstuurd is, kan het niet meer botsen. Als gevolg hiervan zou de maximale lengte niet meer dan 25m kunnen bedragen (met een bitsnelheid van 1Gigabit/sec). Omdat dit onaanvaardbaar kort is, werden oplossingen bedacht. De enige goede oplossing is alleen full duplex Gigabit-switches gebruiken. Bij full duplex-communicatie geldt dat elke netwerkkaart tezelfdertijd frames kan versturen en frames ontvangen. Botsingen kunnen dan niet meer voorkomen. Om snelheden van 1Gigabit/sec te bereiken, werd vooral glasvezel gebruikt. Het kan nu ook over UTP-5e. De 8 draadjes moeten dan wel allemaal gebruikt worden. Er worden 2 bits ineens verstuurd. Hiertoe worden 5 spanningsniveaus gebruikt. (5? Ja, met 4 ervan kan 00, 01, 10, 11 gecodeerd worden, de 5° mogelijkheid dient voor besturingsinformatie). Gigabit over UTP lijkt echt wel de limiet te zijn. Men stuurt full duplex op elk paar. Als gevolg van het gebruik van glasvezel schijnt het hek helemaal van de dam te zijn: 10 Gigabit Ethernet kan al en over 100 Gigabit Ethernet zijn sommigen al aan het denken…. Enkele soorten Ethernet zijn aangegeven in onderstaande tabel.

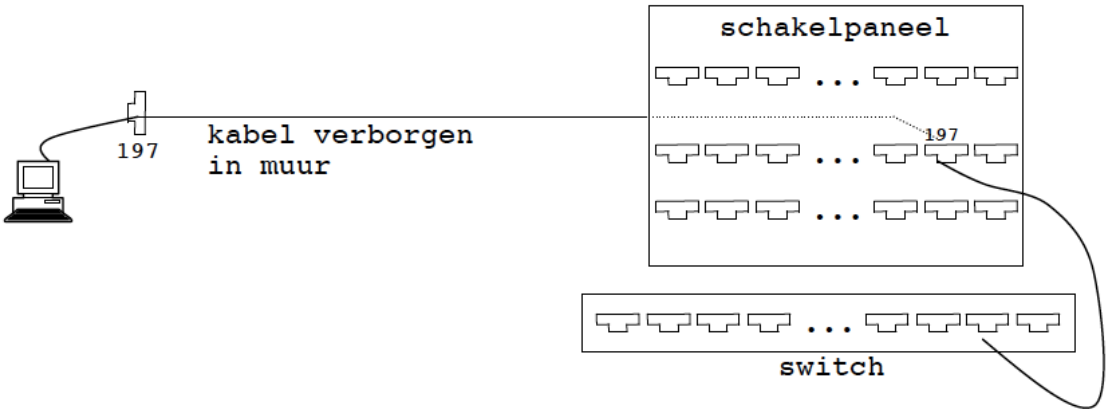
Naam	Kabel	Maximale lengte	Eigenschappen
100Base-TX	UTP cat. 5	100m	100Mbps, full duplex
100Base-FX	Glasvezel	2000m	100Mbps, full duplex
1000Base-T	UTP cat.5	100m	1 Gbps, full duplex
1000Base-SX	Glasvezel	550m	1 Gbps, full duplex
1000Base-LX	Glasvezel; 62 μm	5000m	1 Gbps, full duplex

2.11 Gestructureerde bekabeling

2.11.1 fysieke structuur

Mensen hebben de neiging om met van alles en nog wat te blijven haperen aan losliggende kabels. Voor het onderhoudspersoneel kunnen ze een nachtmerrie zijn. Er wordt daarom meer en meer gebruik gemaakt van gestructureerde bekabeling (figuur 2.5). Bij dit type van bekabeling:

- bevinden zich in de verschillende lokalen van het gebouw evenveel RJ-45 contactpunten als er pc's moeten aangesloten worden;
- loopt er van elk contactpunt een TP-kabel verborgen in de muur, in een kabelgoot, in een technische ruimte,... naar een centraal schakelpaneel (E. patch panel);
- wordt een pc fysiek in het netwerk opgenomen door:
  - de netwerkkaart via een TP-kabel te verbinden met het RJ-45-stopcontact in het lokaal waar de pc zich bevindt (stel dat dit stopcontact nr 197 is);
  - het overeenkomstig contactpunt van het schakelpaneel (in het vb. nr. 197) te verbinden met een poort op de switch.

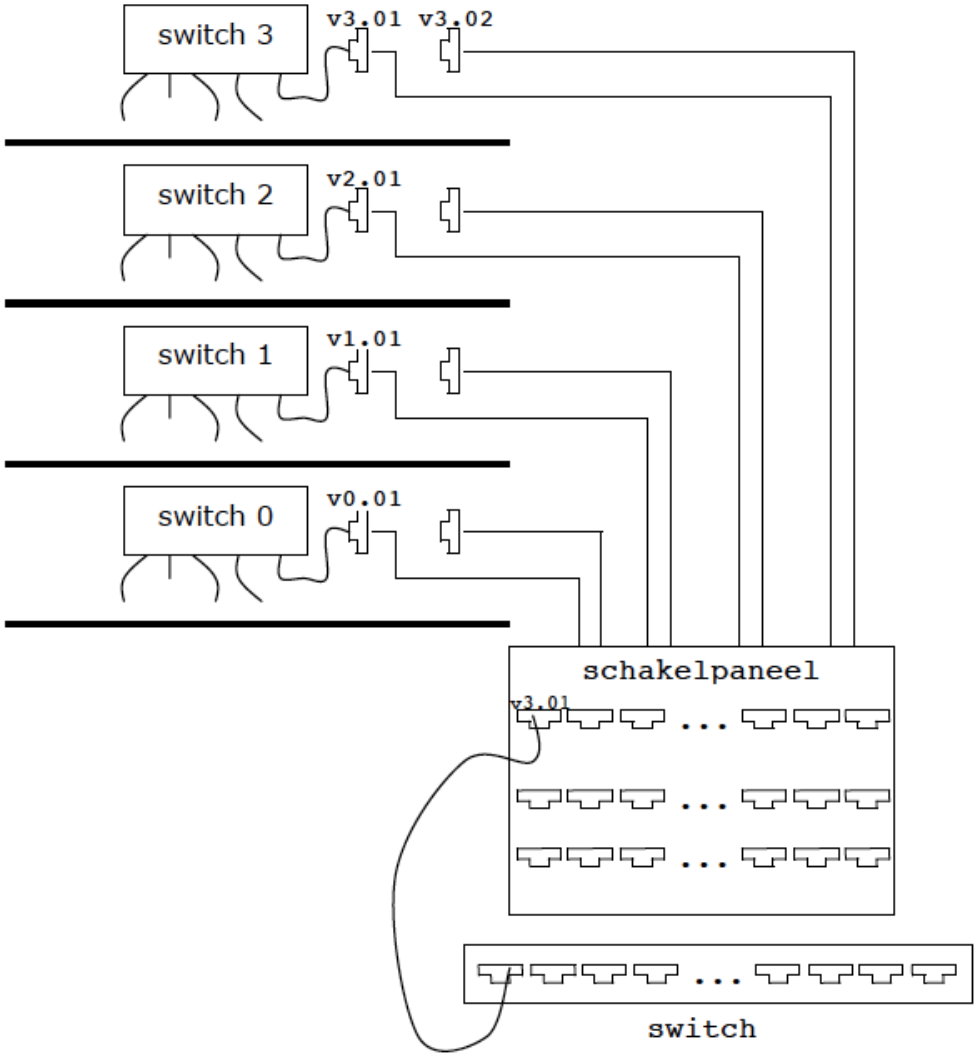


Figuur 2.5: Horizontale netwerkbekabeling

Er is bvb. één schakelpaneel per verdieping met één of meer switches. De bekabeling vanaf de pc's tot aan het schakelpaneel wordt **horizontale** bekabeling genoemd. Het schakelpaneel heet in het Engels horizontal cross connect (HCC). Voor de horizontale bekabeling wordt ook de term distributiebekabeling gebruikt. De EIA/TIA standaard specifieert volgende maximale lengtes voor de kabels
- de kabel van de pc tot aan het stopcontact : max. 10m
- de kabel tussen stopcontact en schakelpaneel : max. 90m

- de kabel tussen schakelpaneel en switch : max 5m

EIA/TIA = electronic industries association/telecommunication industries association



Figuur 2.6: Verticale bekabeling

Om de verschillende verdiepingen te verbinden, onderling en met de servers, is er de zgn. **verticale** bekabeling (figuur 2.6). Op elk verdiep zijn er in het lokaal waar de switches zich bevinden één of meer RJ-45-stopcontacten, vanwaar er weer UTP-kabels (= de verticale bekabeling) naar een schakelpaneel in een centraal lokaal lopen. Dit schakelpaneel wordt vertical cross connect genoemd (VCC). De verticale bekabeling wordt ook backbone (=ruggegraat) genoemd. De verticale bekabeling kan een groot debiet aan netwerkverkeer te verwerken krijgen. Een glasvezelverbinding is daarom geen overbodige luxe. Overigens, als de lengte meer dan 100 m bedraagt, kan men geen TP kabels meer gebruiken.

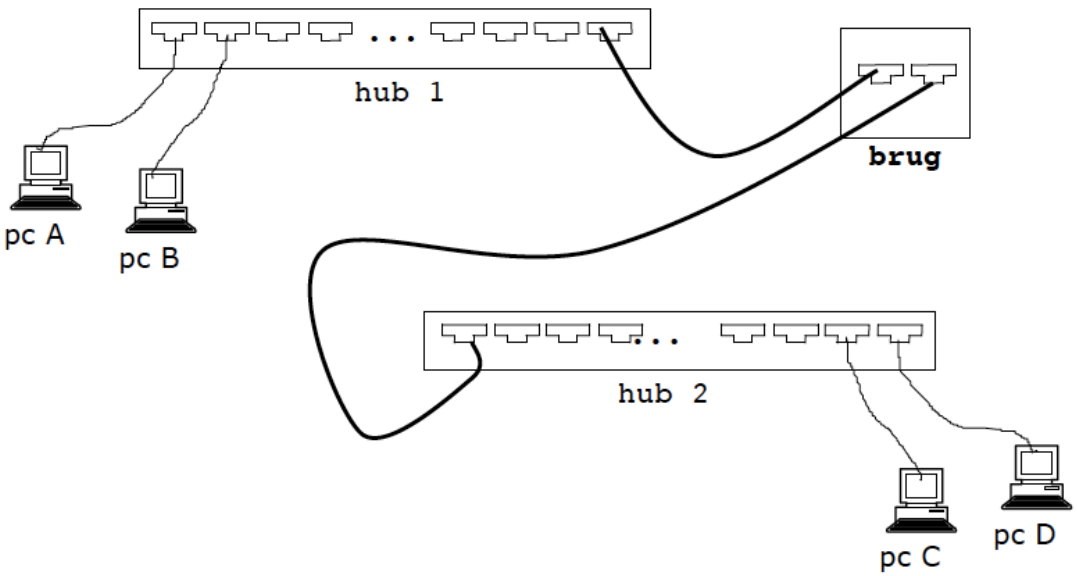
2.11.2 Hiërarchische structuur van het (bedrijfs)netwerk

Als het bedrijfsnetwerk enige omvang heeft, wordt er onderscheid gemaakt tussen:
- toegangsnetwerk (access network),
- distributienetwerk (distribution network),
- kernnetwerk (core network).

Het onderscheid en waar precies de grens ligt tussen deze 3 is niet altijd even duidelijk. De term access network is wel duidelijk: het netwerk dat bestaat uit de kabels die de pc's verbinden tot aan de eerste switch (of hub) inclusief die switch. Het netwerk dat die switches verbindt en waarop geen gebruikers voorkomen is het distributienetwerk. Sommigen rekenen de kabels die de eerste switches onderling verbinden evenwel ook tot het toegangsnetwerk. Meerdere distributienetwerken worden verbonden door een kernnetwerk. Dit netwerk bestaat uit o.m. routers en snelle verbindingen.

2.12 Brug

Om 2 Ethernet-LAN's rechtstreeks te verbinden, kan men de hubs/switches verbinden. Het kan ook door een brug te gebruiken, zoals in onderstaande figuur 2.7.



Figuur 2.7: Brug

Hub1 en de computers die ermee verbonden zijn, vormen segment 1 (idem segment 2). Beide segmenten zijn verbonden met een brug. Een brug is intelligent zoals een switch. In een brug worden de adressen van zender en

bestemmeling nagegaan. Indien ze op dezelfde poort aangesloten zijn, dan wordt het frame NIET doorgestuurd via de andere poort. Als in figuur 2.7, A een frame naar B zendt, dan zorgt de brug ervoor dat dit frame NIET op segment 2 terecht komt. Als C tezelfdertijd, een frame naar D zendt, dan is er geen botsing. Zou men, i.p.v. een brug, de hubs in cascade schakelen, dan komt het frame van A naar B, zowel op segment 1 als op segment 2. Zoals een switch, leert een brug welke adressen bij welke poort horen. Het begrip brug (E. bridge) is ruimer dan hierboven aangegeven. Zo zijn er bruggen die verschillende types 802-LAN's kunnen verbinden. Zo kan men een 802.3-LAN (Ethernet) koppelen aan een 802.5-LAN (Token ring) met een brug. In figuur 2.7 zouden:
- hub1 en de aangesloten pc's onderling via Ethernet communiceren
- hub2 en de aangesloten pc's onderling via Token ring communiceren
- de brug frames die voor het andere segment bestemd zijn omzetten (802.3-frames naar 802.5 frames en omgekeerd).

Bij het koppelen van verschillende 802-LAN's zijn sommige aspecten eenvoudig, zo zijn de MAC-adressen bij de verschillende types LAN gelijkaardig, d.i. getallen van 12 hexadecimale cijfers. De meest complexe taak bij het omzetten, is de toegang tot de drager:
- bij 802.3: wachten tot de drager vrij is en botsingen corrigeren
- bij 802.5: wachten tot men het recht krijgt om te zenden
Ook kunnen de maximaal toegelaten frame-lengtes verschillend zijn.

Algemeen wordt een brug gedefinieerd als een apparaat dat netwerken kan koppelen in de datalink-laag. Een switch wordt wel eens een **multiport bridge** of zelfs gewoon bridge genoemd.

2.13 Frame-types

In LAN's wordt vrijwel uitsluitend nog gebruik gemaakt van Ethernet (en als transport- en netwerkprotocol vrijwel nog uitsluitend TCP/IP). Als gevolg van standaarden uit het verleden zijn er 4 types van Ethernet frames:

- Ethernet II
- IEEE 802.3
- IEEE 802.2
- SNAP (SNAP = subnet access point)

Het gebruik van 802.x-frames maakt het mogelijk om LAN's van verschillend type (bvb. Ethernet en Token Ring) te koppelen in de datalinklaag d.m.v. een brug.
De TCP/IP-adepten blijven het evenwel houden bij Ethernet II frames. Koppelen van verschillende netwerken gebeurt in de netwerklaag door routers

(zoals uiteengezet zal worden in een volgend hoofdstuk). Ethernet II bestaat al van in de jaren 80 van de vorige eeuw. Het is een verbetering van het oorspronkelijke Ethernet van Xerox en werd op punt gesteld door Digital, Intel en Xerox. Daarom wordt het ook DIX genoemd. Een stuurprogramma van een netwerkkaart kan de verschillende types aan. Als men TCP/IP als communicatieprotocol installeert, wordt automatisch Ethernet II gekozen. Ter illustratie, enkele verschillen (geen examenstof).

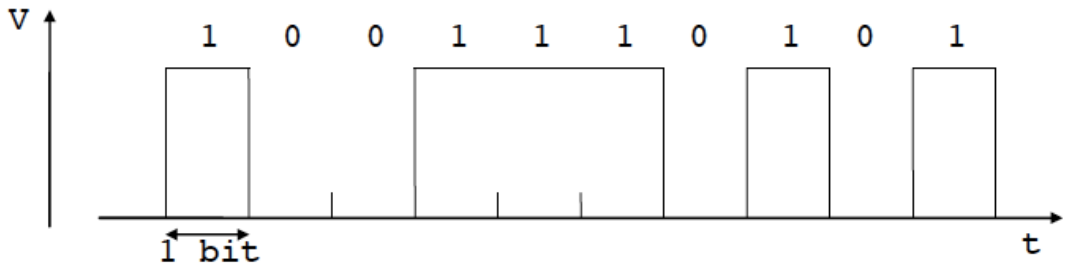
- Een Ethernet II-frame bevat geen lengte-veld maar wel een code die aangeeft voor welke toepassing het frame verstuurd wordt (bvb. ARP, IP, …). Ethernet II frames bevatten geen lengte-veld, hoe kan dan de lengte achterhaald worden? De lengte kan afgeleid worden uit de toepassingscode : sommige berichten hebben een vaste lengte; in andere (bvb. een IP-pakket) staat de lengte op een vaste plaats. Het is duidelijk dat men zich hier (dus bij de TCP/IP-aanhangers) niets aantrekt van de scheiding in OSI-lagen.
- In frames van de andere types wordt de lengte aangegeven (2 bytes). De lengte is nooit groter dan 1500 bytes. Dit veld wordt ook gebruikt om de toepassingscode in te zetten. Deze code is altijd groter dan de maximale frame-lengte.

2.14 Fysieke voorstelling van bits

Om bitrijen te versturen moet één of ander fysiek signaal en medium gebruikt worden. Dit kan bvb. elektrische spanning op een draad zijn. Het aantal bits dat per seconde verstuurd kan worden, hangt af van de bandbreedte.

2.14.1 Binaire codering

I.p.v. ”binaire codering” zegt men ook: non return to zero of NRZ. Bits worden voorgesteld door het gebruik van 2 spanningsniveaus. Zo bvb.:
(0 Volt) = 0
(5 Volt) = 1 De bitrij 1001110101 wordt bij binaire codering verstuurd als (V = spanning):

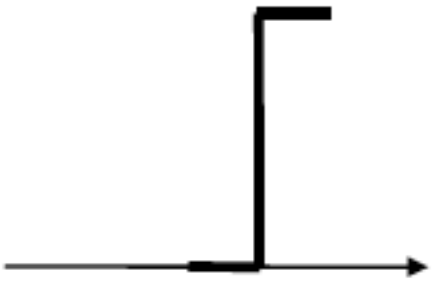


Voor correcte transmissie is het bij deze codering nodig dat er ook synchronisatie-informatie meegestuurd wordt, zodat de klok van zender en ontvanger perfect gelijk lopen. I.p.v. de klok-informatie apart te versturen kan men ook een andere codering gebruiken.

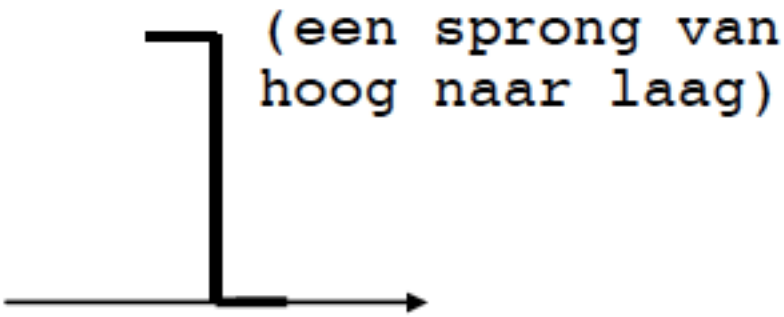
2.14.2 Manchester-codering

Bij Manchester-codering wordt:

- een 0 voorgesteld door:

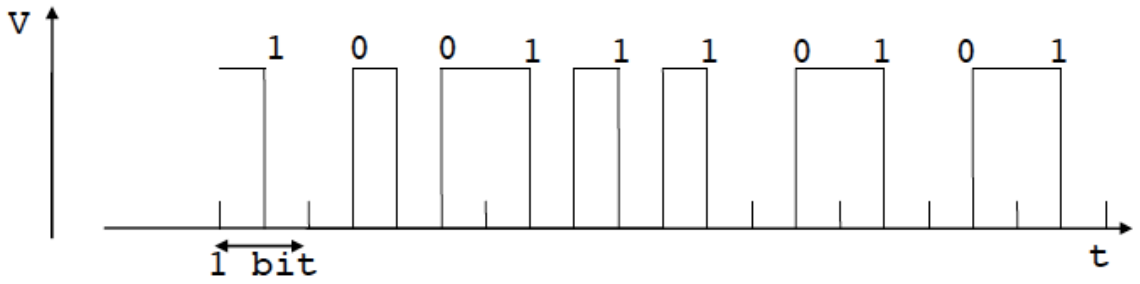


- een 1 voorgesteld door:



Halfweg elke bit is er altijd een sprong. Dit laat toe dat de klok van de ontvanger perfect gelijk loopt met die van de zender.
De bitrij 1001110101 wordt verstuurd als in figuur 2.8.

Het voordeel van Manchester-codering t.o.v. NRZ is dat bij Manchester-codering de voorstelling van een bit ook klok-informatie bevat. Als een lange rij gelijke bits moet verstuurd worden dan zou de synchronisatie van de klokken van zender en ontvanger kunnen verloren gaan bij NRZ. Een ander nadeel van NRZ dat in de literatuur aangehaald wordt, is : een lange rij van gelijke bits heeft als gevolg dat er gedurende langere tijd elektrische



Figuur 2.8

spanning op de lijn is. Dit zou oververhitting en transmissiefouten als gevolg kunnen hebben. Het nadeel van Manchester-codering is dat de pulsen maar half zo breed zijn als bij binaire codering. Hiertoe is er meer (analoge) bandbreedte nodig. Manchester-codering was de techniek voor de oude versie van Ethernet (10 Mbit/s).

Hoofdstuk 3

Het Internet Protocol (IP)

Als in een computernetwerk Ethernet geïnstalleerd is, dan kunnen de computers Ethernet-frames naar mekaar sturen/van mekaar ontvangen. Werd er gekozen voor een ander type netwerk (Token Ring, Token Bus, ...) dan zijn het andere frames. LAN-frames kunnen niet ”naar buiten” of buiten het netwerk verstuurd worden.

Een gebruiker is maar weinig geïnteresseerd in het uitwisselen van (Ethernet)frames maar wil daarentegen toepassingen: surfen, e-mails versturen, een centrale printer, ... gebruiken. Er is dus een kloof tussen wat de gebruiker wil; netwerktoepassingen gebruiken en wat het netwerk kan; frames versturen. Het dichten van deze ”kloof” is erg complex. Om deze complexiteit te beheersen, wordt in stappen[1] gewerkt.

De eerste stap naar een meer gebruiksvriendelijk netwerk is de **IP programmatuur**[2]. Als IP op de PC’s geïnstalleerd en geconfigureerd is, dan:

- heeft elke computer een uniek adres, een zgn. IP-adres
- kan elke computer naar (van) elke computer een zgn. IP-pakket sturen (ontvangen)

De netwerktechnologie (Ethernet, Token Ring, ATM ,...) die gebruikt wordt om het netwerk op te zetten, **speelt geen rol meer** voor wat IP betreft. Het is dan ook normaal dat afstand evenmin een rol speelt. Een IP-pakket wordt verstuurd naar een computer waarvan het IP-adres gekend is. Deze computer mag zich in hetzelfde gebouw of aan de andere kant van de wereld bevinden.
IP zorgt voor een versturing van IP-paketten zonder garanties. IP is een **niet verbindingsgeoriënteerde, best-effort service**.

[1] en bijgevolg in abstractielagen

[2] De ”netwerklaag” van het Internet bevat drie functies: routering, controle en het IP protocol. In dit hoofdstuk bespreken we enkel het IP protocol.

Het zou rampzalig zijn als verschillende fabrikanten eigen versies van de IP-programmatuur zouden verspreiden. De voorschriften waaraan de IP-programmatuur moet beantwoorden, zijn vastgelegd in het **internet protocol**. Het is niet onbelangrijk nog even te wijzen op de verschillende betekenissen waarin de term IP gebruikt wordt:

- IP betekent strikt genomen: het geheel van afspraken
- IP wordt ook gebruikt in de betekenis van: de implementatie of het computerprogramma van deze IP afspraken

Momenteel worden er twee verschillende versies van het IP protocol gebruikt op het Internet nl. IPv4 en IPv6.

Om verder dan het eigen LAN te communiceren, zijn er default gateways en routers vereist. Uiteraard moeten deze correct geconfigureerd zijn, later meer daarover. Computers wisselen dus de IP-pakketten niet rechtstreeks met mekaar uit, maar de IP-pakketten worden d.m.v. routers doorheen netwerken[3] gerouteerd tot aan de bestemming. De IP-programmatuur gebruikt hiervoor de diensten van de datalinklaag. Hiervoor wordt een IP-pakket ingepakt in een frame waarna dit frame verstuurd wordt. Het type frame hangt af van het gebruikte netwerk: Ethernet-fame, Token-Ring-frame, HDLC-frame (op een WAN-verbinding), ...

3.1 IPv4-adressen

Computers wisselen IP-pakketten uit. In deze pakketten moet o.m. staan, waar ze vandaan komen en waar ze naartoe moeten. Daarom moeten de netwerken zelf en bijgevolg de computers in een netwerk een adres hebben, logischerwijs moeten verschillende netwerken en verschillende computers binnen een netwerk verschillende adressen hebben.

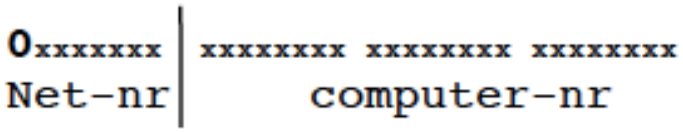
IPv4 Adressen van 32 bits Elk computernetwerk krijgt een adres van 32 bits. Daarnaast krijgt elke computer binnen een netwerk ook een adres van 32 bits. Deze adressen moeten opgevat worden als bestaande uit **2 delen**: links het netwerknummer en rechts het computernummer. Er wordt onderscheid gemaakt tussen 3 klassen of grote van netwerken: A, B, C.

Netwerken van klasse A Netwerken van klasse A zijn zeer grote netwerken. Er zijn niet veel van dergelijke netwerken:

- slechts 7 bits worden gebruikt om het netwerknummer aan te duiden; er kunnen dus hoogstens 128 netwerken van klasse A zijn

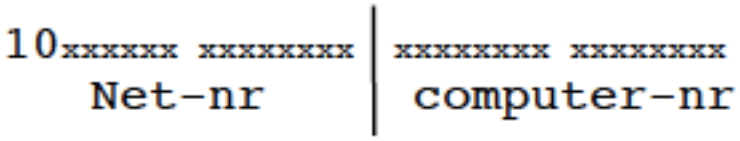
[3] Let op de meervoudsvorm.

- 24 bits worden gebruikt om het computernummer voor te stellen; er kunnen dus meer dan 16 miljoen computers deel uitmaken van een netwerk van klasse A
- de eerste bit van het adres is 0. De structuur van het IP-adres van een computer in een netwerk van klasse A is dus:



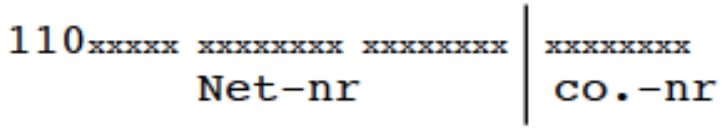
Netwerken van klasse B Netwerken van klasse B zijn middelgrote netwerken.

- 14 bits worden gebruikt om het netwerknummer aan te duiden; er kunnen dus hoogstens 16.384 netwerken van klasse B zijn
- 16 bits worden gebruikt om het computernummer voor te stellen; er kunnen dus niet meer dan 65.536 computers deel uitmaken van een netwerk van klasse B
- de eerste 2 bits van het adres zijn 10. De structuur van het IP-adres van een computer in een netwerk van klasse B is dus:



Netwerken van klasse C Netwerken van klasse C zijn kleine netwerken.

- 21 bits worden gebruikt om het netwerknummer aan te duiden; er kunnen dus meer dan 2 miljoen netwerken van klasse C zijn
- 8 bits worden gebruikt om het computernummer voor te stellen; er kunnen hoogstens 256 computers deel uitmaken van een netwerk van klasse C
- de eerste 3 bits van het adres zijn 110. De structuur van het IP-adres van een computer in een netwerk van klasse C is dus:



Andere adressen Adressen die met 1110 beginnen, zijn multicast-adressen of *adressen van klasse D*. Adressen (van klasse E) die met 11110 beginnen, zijn niet in gebruik. Er zijn nog andere uitzonderingen. Zo is er bvb. geen computer met nummer 0. Als het computernummer in een IP-adres gelijk is aan 0, dan stelt dit IP-adres het ganse netwerk voor. Er is evenmin een computer waarvan het computernummer uit allemaal enen bestaat. Als een dergelijk adres voorkomt in een IP-pakket, dan is dit een zgn. broadcast-pakket: het pakket is bedoeld voor alle computers van het netwerk. **Daarom kan een computer, in het computer-nummer gedeelte van zijn IP-adres, nooit een nummer hebben dat uitsluitend uit nullen of enen bestaat.**

Gepunte notatie Ondanks de grote inspanningen van de lectoren Computersystemen, doet de binaire notatie voor de meeste mensen nogal onwennig aan. IP-adressen worden dan meestal ook anders voorgesteld. Elke groep van 8 bits wordt omgerekend naar de decimale voorstelling. De groepen worden gescheiden door een punt. Het adres:

11011010 01001101 00110111 01111111

wordt dan:

218.77.55.127

Met deze notatie gaan:

- IP adressen voor netwerken van klasse A van 0.0.0.0 tot 127.255.255.255
- IP adressen voor netwerken van klasse B van 128.0.0.0 tot 191.255.255.255
- IP adressen voor netwerken van klasse C van 192.0.0.0 tot 223.255.255.255

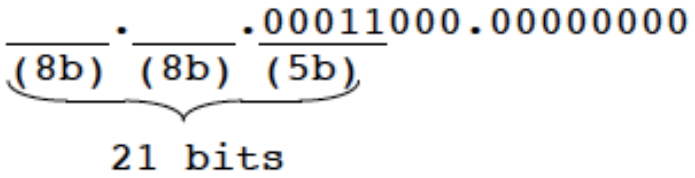
Alhoewel men met deze adressering beschikt over meer dan 4 miljard adressen, blijkt dit te weinig te zijn. Intussen is er een nieuwe standaard: IPv6. Hierin zullen de adressen uit 128 bits bestaan. De hierboven gebruikte versie heet IPv4. Een bedrijf dat zijn eigen internet (zoiets wordt wel eens intranet genoemd) wenst uit te bouwen, d.i. meerdere eigen netwerken koppelen, kan IP-adressen naar eigen goeddunken toekennen. Wenst men echter een netwerk te koppelen aan het Internet, dan moet men het netwerkdeel van de IP-adressen aanvragen. De ICANN (Internet Corporation for Assigned Names and Numbers; vroeger de IANA) is bevoegd om netwerk-adressen toe te kennen. Voor sommige delen van de

IP-adresruimte is deze bevoegdheid gedelegeerd aan andere organisaties. Een bedrijf, een school, gebruikt het toegekende netwerk-adres of netnummer en kan haar computers naar eigen goeddunken (liefst op een ordentelijke wijze) van een computernummer voorzien. Aldus wordt verzekerd dat wereldwijd, verschillende computers verschillende IP-adressen hebben.

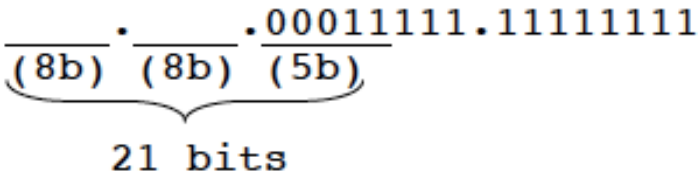
Vernieuwing Zoals hoger vermeld, bestaat een IP-adres uit 2 delen, het netwerknummer en het computernummer binnen het netwerk. De opdeling in klassen A, B, C is intussen voorbijgestreefd. **Het zijn niet langer de eerste bits van het IP-adres die bepalen hoeveel bits het netwerknummer voorstellen.** Het aantal bits die het netwerknummer voorstellen, wordt nu **expliciet** aangegeven. Zo is de betekenis van

185.200.24.0/21

het netwerk dat begint bij IP-adres 185.200.24.0 en waarvan de eerste 21 bits dezelfde zijn. D.i. 24 decimaal is binair 00011000 vanaf:



tot:



of, in decimale notatie: van 185.200.24.0 tot 185.200.31.255.

Men noemt deze notatie: **basisadres/aantal bits**. De combinatie basisadres/aantal bits moet wel zinvol zijn. Met 185.200.24.0/21 is het o.k. omdat in 185.200.24.0 de bits die na de eerste 21 bits komen, allemaal 0 zijn. Maar hetzelfde basisadres gevolgd door /20 is geen zinvolle notatie. Waar vroeger een bedrijf met een middelgroot netwerk IP adressen van type B moest aanvragen en dus zowat 65000 IP-adressen moest afnemen, kan een bedrijf nu IP-adressen voor bvb. 8 netwerken van klasse C krijgen en deze met een notatie *basisadres/aantal bits* identificeren. Men noemt deze vernieuwing: **CIDR** (classless interdomain routing).

3.2 Het IP-pakket

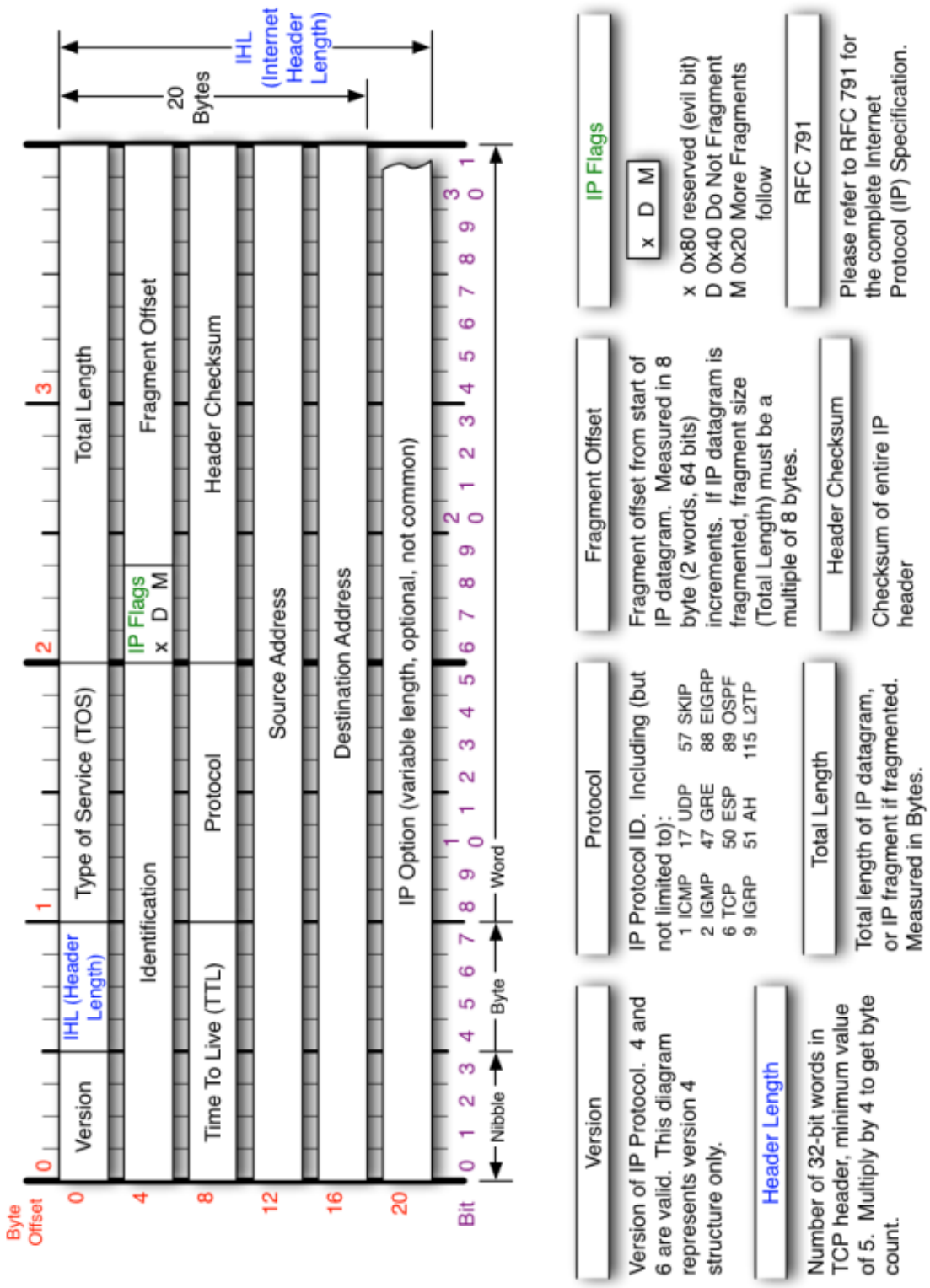
De gebruikers van de IP-programmatuur zijn programma's uit de transportlaag of programma's uit hogere lagen. Op vraag van deze programma's, leveren de IP-modules van de twee machines onder andere de dienst om IP pakketten uit te wisselen. Als de afzender en de bestemmeling zich niet op hetzelfde fysiek netwerk bevinden, dan is de tussenkomst van een router (zie verder) vereist. Het versturen van IP-pakketten is een **verbindingsloze dienst** en IP-pakketten worden ook datagrammen genoemd. Een IP-pakket bestaat uit: een **hoofding** en een **datadeel**. De lengte van hoofding en datadeel liggen niet vast. Het Internet Protocol specifieert:

- welke velden moeten en mogen voorkomen
- het aantal bits dat voor elk van deze velden voorzien is
- waar (d.i. bij welke bit) elk van deze velden begint; eventueel moet deze beginpositie afgeleid worden uit een opgegeven lengte
- wat de inhoud van deze velden kan zijn
- wat deze inhouden betekenen

Vermits het protocol niet specifieert hoe lang hoofding en datadeel zijn, moeten er velden zijn die de lengte ervan aangeven. Het eerste veld geeft het protocolversienummer **4** of **6** aan.

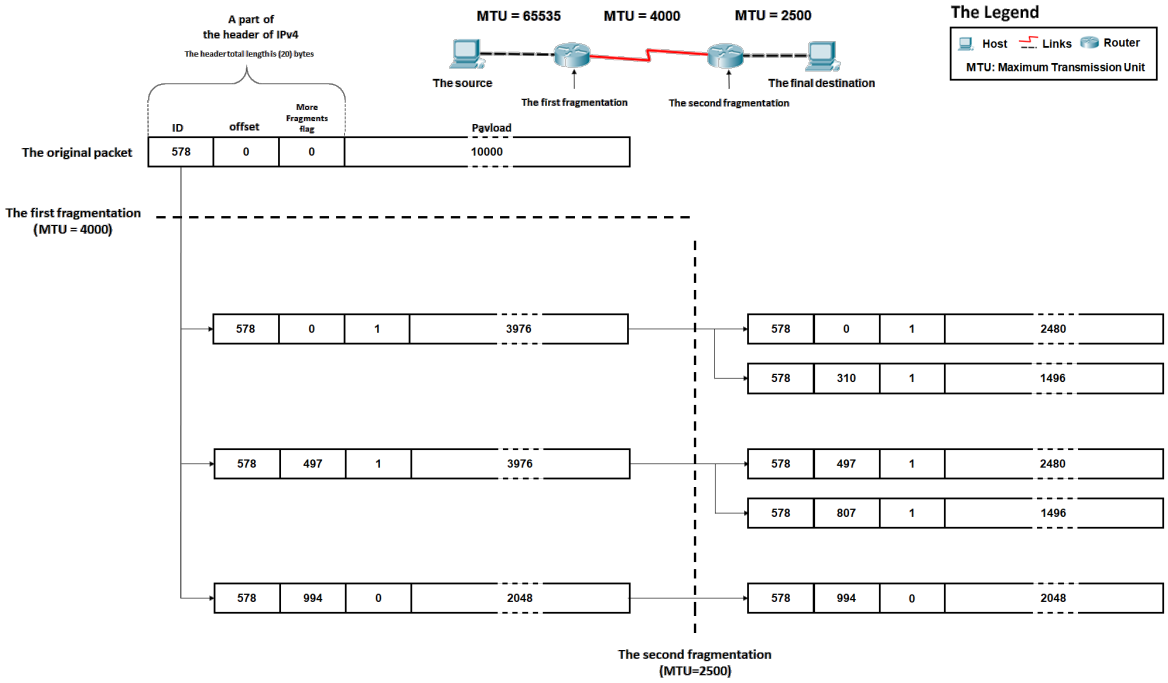
De hoofding van een IPv4-pakket De hoofding bevat volgende velden, zie figuur 3.1:

- Versie (4 bits). De versie is nu 4. Als hier (in de toekomst) 6 zal staan, klopt onderstaand verhaal niet meer.
- Lengte van de hoofding (4 bits: aantal woorden van 32 bits). Deze lengte zal afhangen van het aantal opties dat gespecifieerd wordt.
- Totale lengte van het pakket (16 bits). De totale lengte zou dus 64 Kbyte kunnen zijn!
- DS-veld (differentiated services, heette vroeger TOS-veld, d.i. type of service).
  - De eerste 6 bits ervan zijn het DSCP-veld (differentiated services code point) : hiermee kan het type van toepassing weergegeven worden, bvb. een real-time toepassing zoals IP-telefonie
  - de laatste 2 bits zijn het ECN-veld (explicit congestion notification; hiermee kan een router melden dat er congestie zit aan te komen of een feit is ...).



Figuur 3.1: IPv4 header

- Identificatie (16 bits). Dit is een nummer dat door de zender toegekend wordt (0, 1, ..., 65535,0, ...) en wordt gebruikt bij fragmentatie (later meer daarover).
- Resterende bestaanstijd (E. time to live of TTL). De zender vult hier een tijd in. Langer dan deze tijd zal het pakket niet op het internet vertoeven. Als een pakket door een router gestuurd wordt, wordt de resterende bestaanstijd met 1 verminderd. Wordt de resterende bestaanstijd gelijk aan 0, dan wordt het pakket weggegooid. De afzender krijgt in dit geval een foutmelding. Het gebruik van een resterende bestaanstijd voorkomt dat (bij een fout) pakketten zouden blijven ronddolen op het netwerk.
- Protocolnummer. Het protocolnummer is een code die aangeeft voor welke bovenliggende module (TCP, UDP, ICMP, ...) in het TCP/IP-model het pakket verstuurd wordt. Aan de ontvangstzijde, beslist de IP-module op basis van dit nummer, aan welke module de data zullen afgeleverd worden. Enkele voorbeelden:
  - 1 : ICMP
  - 2 : IGMP
  - 6 : TCP
  - 50 : ESP
  - 51 : AHP
- CRC. Alleen voor de hoofding wordt de CRC berekend. Hierdoor kunnen eventuele fouten in de hoofding door de routers vastgesteld worden. Als er een fout vastgesteld wordt, wordt het pakket vernietigd. Bemerk: vermits elke router de resterende bestaanstijd vermindert, moet elke router ook de CRC herberekenen!
- IP-adres v.d. afzender (32 bits).
- IP-adres v.d. bestemmeling (32 bits).
- Fragmentatie offset. In principe komt één IP-pakket terecht in één frame (bvb. Ethernet-frame). In sommige netwerken is de maximale frame-lengte echter kleiner dan de framegrootte van het netwerk waar het frame vertrok. Als een IP-pakket op zijn weg naar de eindbestemming moet verstuurd worden over een netwerk waarvan de maximale frame-lengte kleiner is, zal het IP-pakket gefragmenteerd worden. De fragmenten zien er uit als IP-pakketten met dezelfde hoofding als het oorspronkelijk pakket, behalve:
  - de totale lengte: in de hoofding van een fragment staat de totale lengte van het fragment
  - het fragmentatieveld: dit veld geeft aan:



Figuur 3.2: IP fragmentatie. Bron: wikipedia

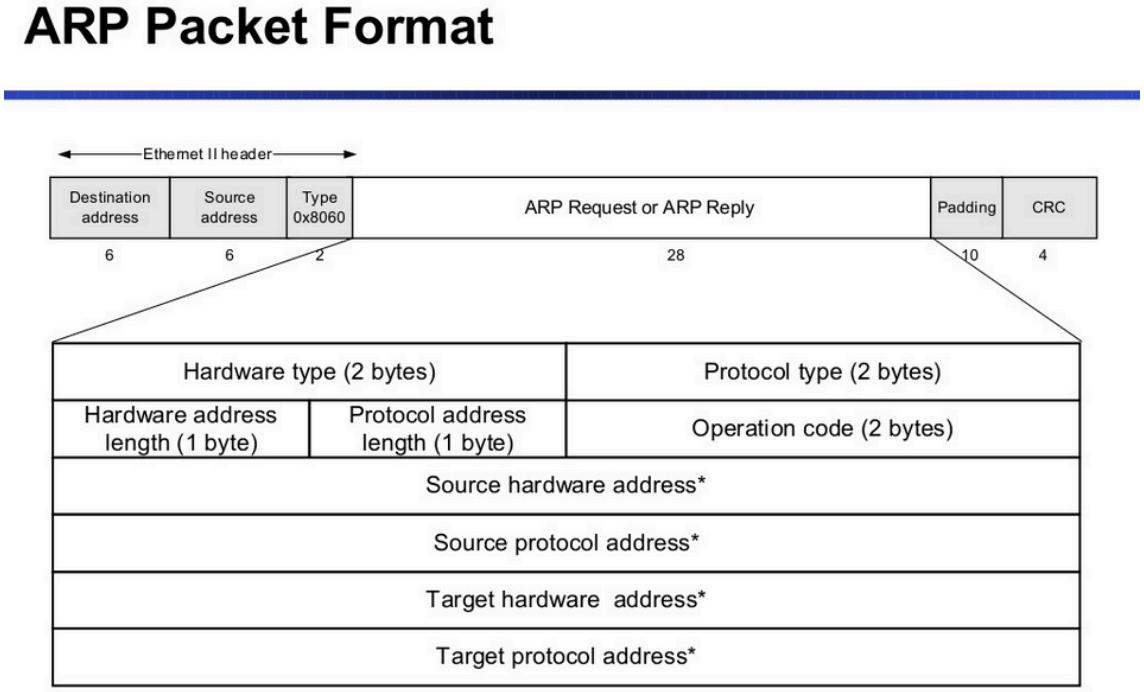
  * of een fragment het laatste fragment is (of niet)
  * waar het fragment thuishoort in het oorspronkelijk IP-pakket
- de CRC

De fragmenten worden als onafhankelijke IP-pakketten verstuurd. Op de eindbestemming (niet onderweg) wordt het oorspronkelijk pakket weer samengesteld. Zie figuur 3.2.

3.3 ARP

Programma's die de diensten van de IP-laag gebruiken (dit zijn dus programma's in een laag boven de IP-laag), werken met IP-adressen. De taak van de IP-laag is IP-pakketten op de plaats van bestemming af te leveren. Het IP-adres van de bestemmeling staat in het IP-pakket. Om haar doel te bereiken, maakt de IP-laag gebruik van de diensten van de datalink- en de fysieke laag. (Het TCP/IP model maakt geen onderscheid tussen deze 2)

Wat de adressen betreft: we noemen de adressen op het niveau onder de IP-laag, fysieke adressen of MAC-adressen (van medium access, zoals de MAC-deellaag van de datalinklaag). Dit zijn bvb. de 48-bits-adressen van Ethernet-kaarten. Het datablok (PDU of de bitrij) die op dit niveau overgebracht wordt, noemen we **frame**. Dit kan een Ethernet-, Token Bus-, Token Ring-, HDLC-frame (bvb. op een telefoonlijn) of nog iets anders zijn.



Figuur 3.3: ARP pakketformaat

Het IP-pakket wordt in een dergelijk frame ingepakt. Eén van de vragen die hierbij moet beantwoord worden, is: **gegeven het IP-adres van de bestemmeling[4], wat is dan het bijhorende MAC-adres**? Het IP-adres (van de bestemmeling) omzetten naar het MAC-adres, is het voorwerp van de adres resolutie (A.R.)

Voor adres-resolutie (A.R) wordt het ARP (address resolution protocol) gebruikt. Het werkt als volgt. Elke computer houdt een zgn. cache bij. Dit is een geheugen waarin zich IP-adressen en de bijhorende fysieke adressen bevinden. Als een IP-pakket moet verstuurd worden, wordt het IP-adres van de bestemmeling opgezocht in die cache. Staat het IP-adres erin, dan is het bijhorende fysiek adres gevonden. Als één (of meerdere) pakketten moet(en) verstuurd worden naar een IP-adres dat NIET in het cachegeheugen staat, dan komen deze pakketten in een wachtrij, en verstuurt de computer een speciaal broadcast-frame, als volgt:

- Het bestemmingsadres is een broadcast-adres: dit is geen adres van een Ethernetkaart; alle kaarten lezen een frame met dit adres. Het broadcast-adres bestaat uit allemaal enen.
- Het type is: 0806. (TCP/IP gebruikt Ethernet II-frames); het type-veld bepaalt de module waaraan de data van het frame moeten afgeleverd worden; 0x0806 betekent afleveren aan de ARP-module.

[4]Voor de eenvoud mag je nu even veronderstellen dat de bestemming zich binnen hetzelfde netwerk bevindt.

- De ARP-boodschap bevat o.m.: fysieke adres van de afzender (6 bytes), IP-adres van de afzender (4 bytes), 6 niet gebruikte bytes, IP-adres waarvan men het fysiek adres wenst te kennen, een code die aangeeft dat het een ARP-verzoek is.
- Alle computers op het fysiek netwerk lezen een dergelijk frame EN geven de boodschap door aan hun ARP-module.

Enkel de ARP-module van de computer die haar IP-adres herkent, stuurt aan de zender het gevraagde antwoord[5]. Dit antwoord heeft dezelfde vorm als de ontvangen ARP-boodschap, met dit verschil: nu geeft de code aan dat het om een ARP-antwoord gaat, en **de 6 niet gebruikte bytes bevatten het gevraagde fysisch adres**. De informatie in de cache wordt vernietigd als een vooraf vastgelegde tijd **verstreken** is. Wie zich zou afvragen waarom: een pc kan een andere Ethernetkaart krijgen, bvb. omdat de vorige defect is.

3.4 Subnetten en VLSM

De term *internet protocol* suggereert dat meerdere computernetwerken of netten (daarom dus 'internet') met mekaar verbonden geworden zijn. Men kan netwerken die in gebruik zijn, na enige tijd met mekaar verbinden of men kan al bij de installatie van een nieuw netwerk, dit netwerk opdelen. Een fundamenteel begrip bij IP is de term **subnet**. **Een subnet bestaat uit computers die deel uitmaken van hetzelfde fysiek netwerk.** Dit betekent dat ze rechtstreeks frames naar mekaar kunnen sturen. De indeling van netwerken in subnetten, is te herkennen aan de IP-adressen van de computers in elk subnet. Computer A en computer B maken deel uit van hetzelfde subnet als "de eerste bits" van het IP-adres van A gelijk zijn aan de eerste bits van het IP-adres van B. Belangrijke vraag hierbij is natuurlijk, hoeveel bits moeten als eerste bits beschouwd worden?

Dit wordt aangegeven met 32 bits, als volgt:

11111111...100...0

(enen gevolgd door nullen). Dit speciale bitpatroon wordt subnetmasker genoemd. Het wordt genoteerd als een IP-adres. In een IP-adres moet een bit tot "de eerste bits"gerekend worden als er in het subnetmasker een 1 staat op de plaats van die bit.

Voorbeeld:
Subnetmasker = 255.255.255.248
Geef enkele subnets in volgende vorm subnet nr. : x.y.z.…

[5]Let op: het antwoord wordt dus NIET in broadcast verstuurd.

subnet 1 : 10.2.4.1 t/m 10.2.4.6;
subnet 2 : 10.2.4.9 t/m 10.2.4.14;
…

Omdat 255 = $(1111\ 1111)_b$ en 248 = $(1111\ 1000)_b$ bestaat het masker hier dus uit 29 enen en 3 nullen. Om tot hetzelfde subnet te behoren mogen alleen de laatste 3 bits v.h. IP-adres van een computer verschillen. Onderstaande tabel geeft mogelijke subnetmaskers en het aantal computers dat kan deel uitmaken van de overeenkomstige subnetten.

Subnetmasker	**Aantal computers**	
255.255.255.255	-	255=$(1111\ 1111)_b$
255.255.255.254	-	254=$(1111\ 1110)_b$
255.255.255.252	2	252=$(1111\ 1100)_b$
255.255.255.248	6	248=$(1111\ 1000)_b$
255.255.255.240	14	240=$(1111\ 0000)_b$
255.255.255.224	30	224=$(1110\ 0000)_b$
255.255.255.192	62	192=$(1100\ 0000)_b$
255.255.255.128	126	128=$(1000\ 0000)_b$
255.255.255.0	254	
255.255.254.0	510	

(Opm. : als u overal 2 computers mist, er is geen computer 0 en er is geen computer waarvan het computernummer uit alleen enen bestaat.)

Subnetmasker en CIDR

Ook voor subnetten wordt de notatie basisadres/aantal gebruikt, zo bvb.: het netwerk 185.200.24.0/21. In dit netwerk zijn de (host) IP-adressen die je aan een PC kan geven: (185).(200).00011xxx.xxxxxxxx of adressen van 185.200.24.1 tot en met 185.200.31.254. Men zou dit netwerk evengoed kunnen aanduiden als : 185.200.24.0 met subnetmasker 255.255.248.0.

Hier worden dus de laatste 11 bits gebruikt voor het computernummer. Hiermee kunnen 2048 computernummers gevormd worden. Als netwerkbeheerder heb je vaak geen nood aan een netwerk waarin je 2046 hosts kan adresseren. Stel dat de beheerder 4 ”kleinere” netwerken of subnets maak. Als er dus bvb. 4 subnetten zijn, dan kunnen de IP-adressen als volgt verdeeld worden:

subnet 1 : -.-.00011**00**x.xxxxxxxx
subnet 2 : -.-.00011**01**x.xxxxxxxx
subnet 3 : -.-.00011**10**x.xxxxxxxx

subnet 4 : -.-.00011**11**x.xxxxxxxx

De IP-adressen van deze subnetten kunnen als volgt voorgesteld worden:
- 185.200.24.0/23 (d.i. : host IP-adressen van 185.200.24.1 tot 185.200.25.254)
- 185.200.26.0/23 (d.i. : host IP-adressen van 185.200.26.1 tot 185.200.27.254)
- 185.200.28.0/23 (d.i. : host IP-adressen van 185.200.28.1 tot 185.200.29.254)
- 185.200.30.0/23 (d.i. : host IP-adressen van 185.200.30.1 tot 185.200.31.254)

Het subnetmasker waardoor deze opdeling gekarakteriseerd is, is 255.255.254.0 d.i. 23 enen gevolgd door 9 nullen.

In een netwerk wordt het **netwerk-adres van het subnet**, waartoe een computer behoort, gevonden door het IP-adres van de computer te **ennen** (E. to and) met het subnetmasker. Zo bvb. de computer met IP-adres 185.200.29.234 met subnetmasker 255.255.254.0 behoort tot het subnet 185.200.24.0/21. De en-bewerking levert : 185.200.28.0, dus het subnet

```
  (185).(200).00011101.(234)
&(255).(255).11111110.00...
  (185).(200).00011100.(0)
```

3 van hierboven. Vandaar de naam masker: door de nullen van het subnetmasker op het IP-adres van een host of computer te leggen, houden we enkel het netwerkadres over. Alle PC's in een netwerk of subnet hebben datzelfde netwerkadres. Het verbergt (= maskeert) in een IP-adres alles wat in een subnet kan veranderen.

Vaak wordt het werkwoord **subnetten** ook gebruikt voor het opdelen van een groter netwerk in verschillende kleinere netwerken met allemaal hetzelfde subnetmasker.

3.4.1 VLSM

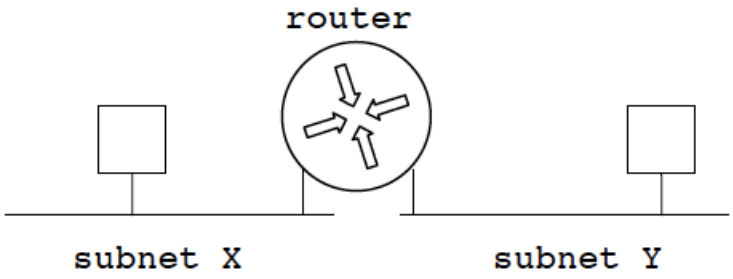
Bij traditioneel subnetten wordt hetzelfde subnetmasker toegepast bij alle subnets. Hierdoor hebben alle subnets evenveel beschikbare host IP-adressen. Traditioneel subnetten creëert subnets van dezelfde grootte.
VLSM of variable length subnet masking laat toe om de IP ruimte op te delen in ongelijke delen. Bij VLSM kan het subnetmask variëren afhankelijk van het aantal host IP's dat er beschikbaar moeten zijn in het subnet.
VLSM subnetting is analoog aan traditioneel subnetten waarbij bits "geleend[6]" worden van het host-gedeelte om het subnet te creëren. Alleen worden de geleende bits optimaal afgestemd op het aantal host IP's dat er

[6]Afgepakt is een beter woord.

nodig zijn. Het komt er op neer dat bij VLSM het netwerk wordt gesubnet om daarna de bekomen subnets verder te subnetten afhankelijk van de specifieke behoeften. Zodoende worden er subnets van verschillende grootte gecreëerd.
Belangrijk is dat behoefte aan IP adressen voor de verschillende subnetwerken, wordt gerangschikt van groot naar klein en dat er dan voor deze afnemende rankschikking consequent wordt gesubnet.

3.5 Router

Binnen een subnet kunnen computers direct met mekaar communiceren, d.w.z. rechtstreeks frames naar mekaar sturen. Om IP-pakketten van het ene subnet naar het andere te sturen is de tussenkomst van een router vereist. Een router bevat programmatuur tot en met de 3^e OSI-laag (dus tot

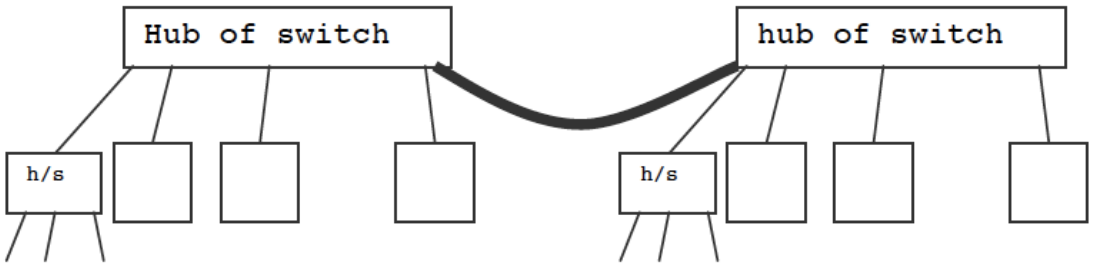


en met de IP-laag). In het samengestelde netwerk zullen 3 soorten verkeer plaats vinden:
- IP-pakketten van een computer van subnet X naar een computer van subnet X
- IP-pakketten van een computer van subnet Y naar een computer van subnet Y
- IP-pakketten van een computer van subnet X naar een computer van subnet Y (en omgekeerd)

Alleen voor de laatste bewerking wordt de router gebruikt. Een IP-pakket wordt door een lagere laag verpakt in een frame (bvb. een Ethernet-frame). Als een IP-pakket naar een ander subnet moet, zal het ingepakt worden in een frame met als bestemmingsadres, het MAC-adres van de router. De IP-laag van de router ontvangt dus het IP-pakket, stelt vast dat dit pakket naar een (ander) subnet moet. Het IP-pakket wordt dan opnieuw ingepakt in een nieuw frame om dit op het andere subnet te versturen. De router maakt deel uit van beide subnetten, de router scheidt bijgevolg netwerken. De router heeft 2 IP-adressen, bvb. : 185.200.25.254 en 185.200.31.254 (respectievelijk : in subnet X; in subnet Y).

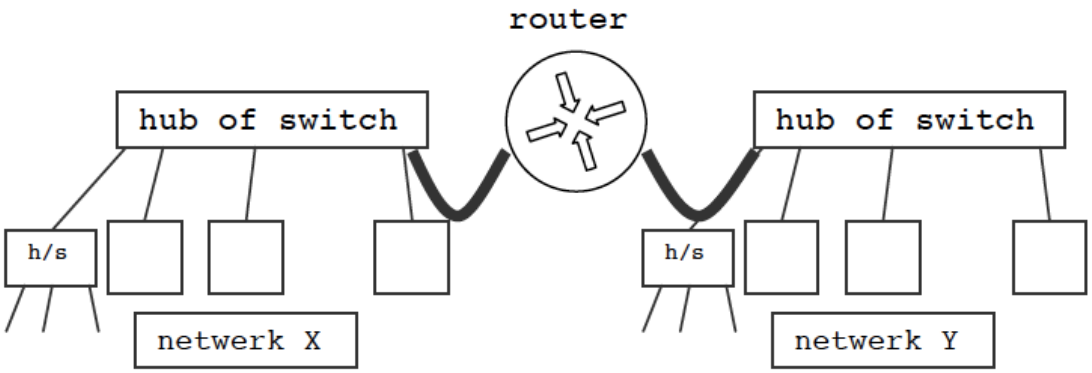
Verbinden van netwerken

1. Ethernet-LAN's in mekaars buurt kunnen verbonden worden door de hubs of switches te verbinden (eventueel met een extra hub of switch).



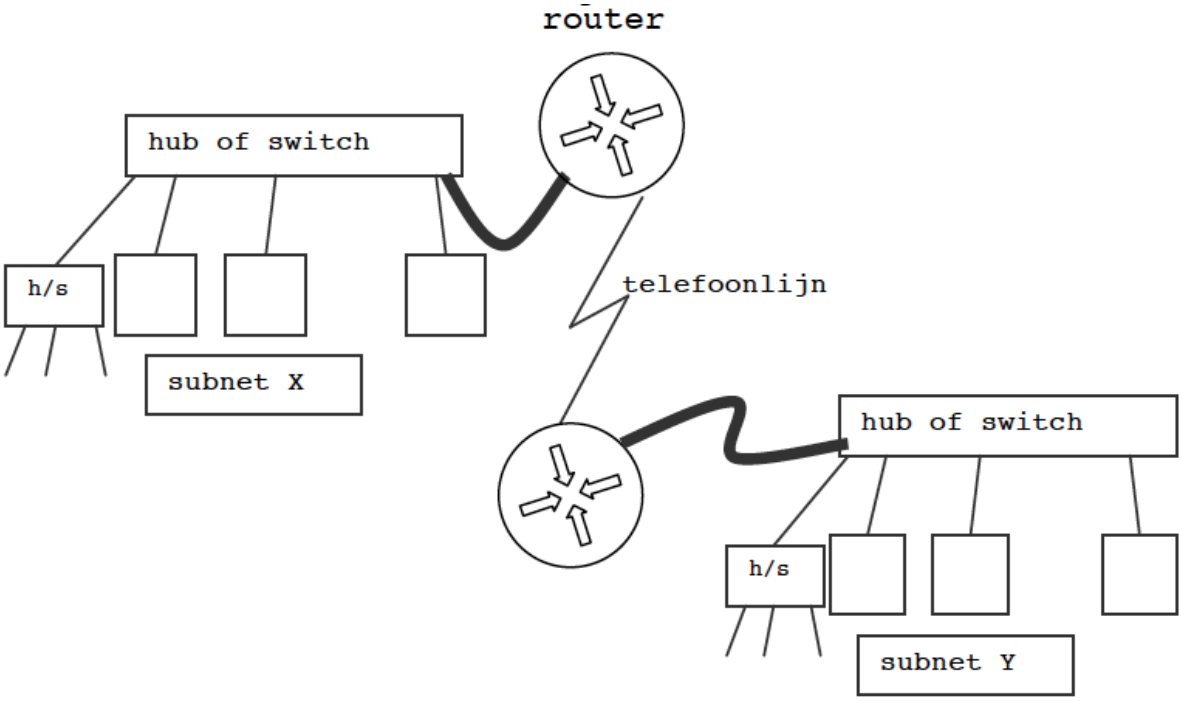
   Deze oplossing kan leiden tot een grote belasting van de netwerken: hubs sturen alle frames verder; gebruikt men switches dan worden nog altijd de broadcast-frames verder gestuurd.

2. Om netwerken van verschillende soort te verbinden wordt gebruik gemaakt van een router; bvb.:



   Een router stuurt de broadcast-frames NIET verder. Om deze reden worden routers ook gebruikt als de netwerken van dezelfde soort zijn (bvb. 2x Ethernet).

3. Een voorbeeld met een WAN-verbinding:



3.6 IP routering

Een internet is opgebouwd uit subnetten die met mekaar verbonden zijn door routers. De taak van een router is voor elk IP-pakket:

- nagaan wat het IP-adres van de bestemmeling is
- op basis van dit IP-bestemmingsadres, beslissen naar welke machine het IP-pakket moet gestuurd worden
- het IP-pakket naar deze machine sturen

Een IP-pakket zal op weg naar zijn eindbestemming langs meerdere routers passeren. Een pakket bereikt zijn bestemming in **etappes** (E. : hop).

3.6.1 Routetabel

Om te beslissen naar waar een IP-pakket moet, wordt door de router een routetabel gebruikt. Een routetabel geeft, voor elk IP-bestemmingsadres, de machine die **next hop** is naar deze bestemming. Een routetabel is natuurlijk geen tabel met 4 miljard items (één voor elk mogelijk IP-adres). Het aantal mogelijke next hops is overigens niet zo groot vermits het aantal interfaces dat een router heeft (meestal 2 tot 4) niet zo groot is. Voor de meeste eindbestemmingen is de next hop dezelfde. Een routetabel moet toelaten om snel te weten te komen welke de next hop is. Een routetabel bestaat uit items van de vorm:

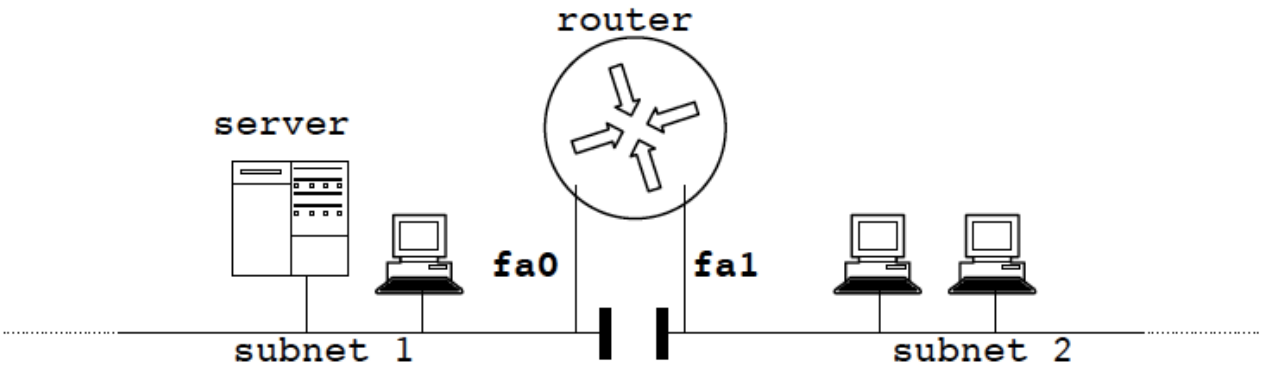
(netwerk, next hop)

Het netwerk kan op 2 manieren aangegeven worden, bvb.:

- (vroeger) : 170.170.138.0 255.255.254.0
- (nu) : 170.170.138.0/23

De next hop is :
- het IP-adres van een router,
- een aanduiding dat directe aflevering mogelijk is (d.i. : de router heeft een netwerk-aansluiting met het netwerk van de bestemmeling).
In plaats van het IP-adres van een router kan ook de interface aangegeven worden langs waar deze router bereikt wordt.

Vb 1 In een LAN met één router verstuurt een pc een IP-pakket (fa0 = fastethernet 0; fa1 = fastethernet 1) Dit pakket kan:



- een verzoek bevatten aan de server op hetzelfde LAN (bvb. een vraag om een dir van de harde schijf van de server)
- een HTTP-verzoek bevatten, met bestemming : ergens in de wereld (niet op hetzelfde LAN)

De routeringsbeslissing die de (IP-programmatuur van de) **PC** moet nemen is : zelfde net of niet?
- In het eerste geval wordt het IP-pakket verpakt in een frame, met als MAC-adres: het adres van de eindbestemming (bvb. de server);
- in het tweede geval wordt het IP-pakket verpakt in een frame, met als MAC-adres: het adres van de router.
Er zijn dus maar 2 mogelijkheden.

Nagaan of een IP-adres zich op hetzelfde net bevindt, kan op basis van het subnetmasker. Een pc heeft een IP-adres en een subnetmasker. Dit subnetmasker geeft aan, hoeveel van de eerste bits het netnummer vormen. Een IP-adres w.x.y.z behoort tot het net van IP-adres a.b.c.d. met subnetmasker k.l.m.n. als en alleen als:

$$w.x.y.z \ \& \ k.l.m.n. = a.b.c.d. \ \& \ k.l.m.n.$$

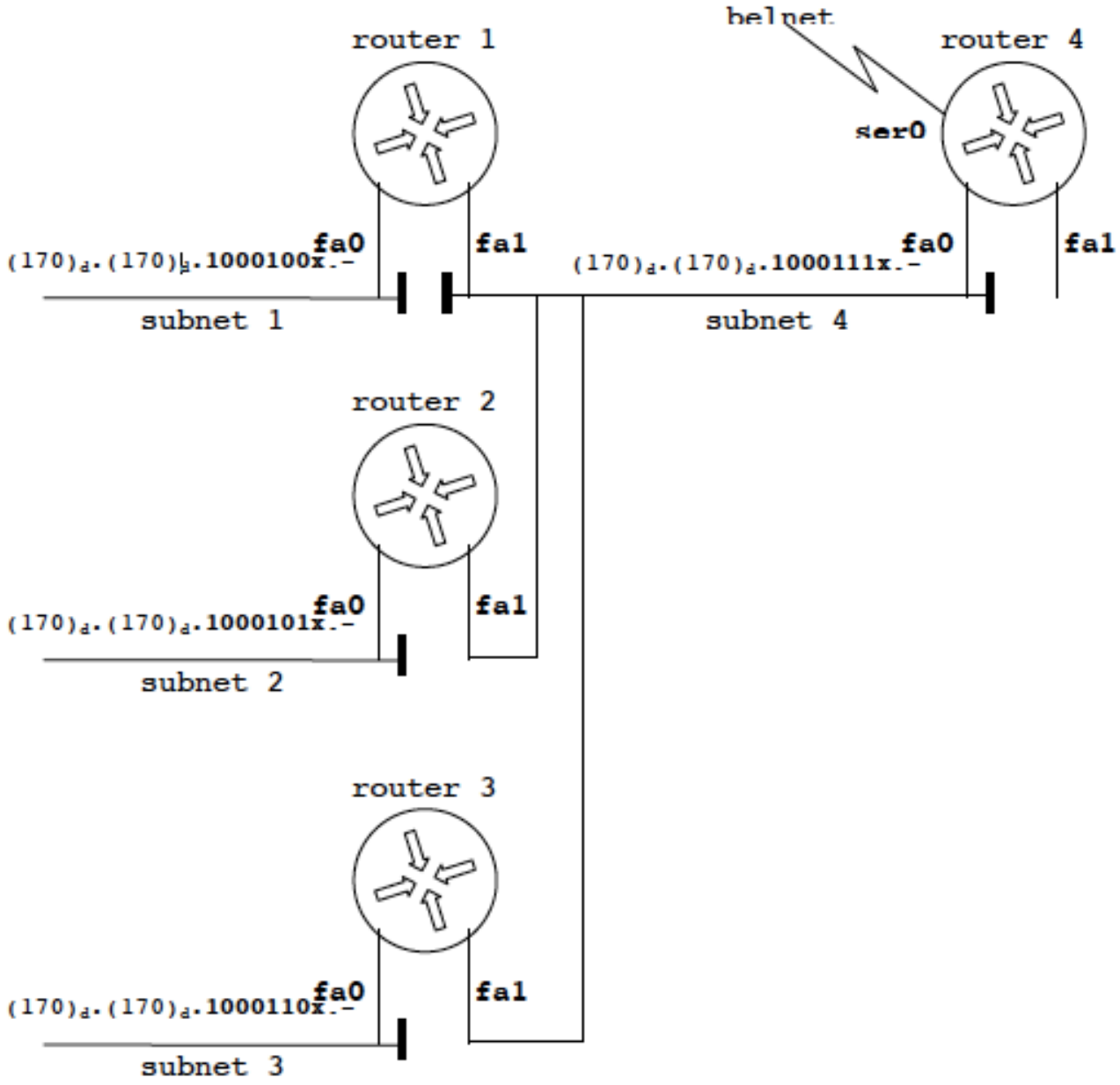
Vb 2 Idem als vb 1, maar nu de routeringsbeslissing voor de router. Van elk IP-pakket zal de router:

- eerst nagaan of het voor hem zelf bestemd is (zoals we verder zullen zien, wisselen routers ook onderling pakketten uit)
- dan nagaan welke het net is waarvoor het pakket bestemd is; hiertoe wordt het netnummer bepaald:
  - is dit het netnummer van het UCLL-netwerk, dan wordt het IP-pakket naar zijn eindbestemming gestuurd, door het in te pakken in een frame met als MAC-adres, het MAC-adres van deze eindbestemming
  - is dit niet het netnummer van het UCLL-netwerk dan gaat het IP-pakket in een frame dat naar de router van Belnet gestuurd wordt.

Vb 3 (10 jaar later) het groep M&T heeft nu 4 subnetten, verbonden met routers (zie volgende blz.). De 4 subnetten met subnetmask 255.255.254.0 zijn:
170.170.136.0/23;
170.170.138.0/23;
170.170.140.0/23;
170.170.142.0/23

Van bvb. router 4 zal de routetabel aangeven: indien bestemming op :
- 170.170.136.0/23 (net 1) : naar router 1,
- 170.170.138.0/23 (net 2) : naar router 2,
- 170.170.140.0/23 (net 3) : naar router 3,
- 170.170.142.0/23 (net 4) : directe aflevering, via fa0
- anders naar Belnet via ser0 (= serieel 0)

Voor bvb. router 2 zal de routetabel aangeven : indien bestemming op :
- 170.170.136.0/23 (net 1) : naar router 1;
- 170.170.138.0/23 (net 2) : directe aflevering, via fa0;
- 170.170.140.0/23 (net 3) : naar router 3;
- 170.170.142.0/23 (net 4) : directe aflevering (!), via fa1;
- anders naar router 4.



Routeringsalgoritme Routetabellen met items van de vorm (netwerk, next_hop) bieden alle mogelijkheden.
Een aparte next hop voor een specifiek IP-adres, kan bijvoorbeeld. Het item (199.138.17.17, 255.255.255.255, R8) betekent : voor alle (! zo is er maar één) IP-adressen waarvan de eerste 32 bits dezelfde zijn als die van 199.138.17.17 is R8 de next hop.
Opdeling in subnetten, kan bvb. met het item (185.200.24.0, 255.255.254.0, R1). Dit betekent dat voor alle IP-adressen waarvan de eerste 23 bits dezelfde zijn als die van 185.200.24.0, R1 de next hop is. De test op de IP-adressen gebeurt door het IP-adres te ennen met het masker en te zien of het resultaat gelijk is aan 185.200.24.0.

Een default router aangeven, kan bvb. met het item (0.0.0.0, 0.0.0.0, R9). Als dit item gebruikt wordt, dan wordt een gegeven IP-adres ”ge-eend”met 0.0.0.0. Dit levert gegarandeerd 0.0.0.0 op. Dus er wordt besloten dat R9 de next hop is.

Voorbeeld Een routetabel zou kunnen zijn:
(199.138.17.17, 255.255.255.255, R8)
(185.200.24.0, 255.255.254.0, R1)
(185.200.26.0, 255.255.254.0, R2)
(185.200.28.0, 255.255.254.0, R3)
(185.200.30.0, 255.255.254.0, R4)
(0.0.0.0, 0.0.0.0, R9)

De routetabel wordt als volgt gebruikt. Voor een gegeven IP-adres, zoek vanaf het begin in de tabel tot er een net gevonden wordt, waartoe dit IP-adres behoort. Gebruik de aangegeven next hop. Indien geen net gevonden: stuur een foutboodschap naar de afzender. Het zoeken van het net kan gedaan worden door het masker in het (elk) item van de tabel te ennen met het gegeven IP-adres. Als het resultaat gelijk is aan het adres in het item dan is de next hop gevonden.
Vb. 1: bestemming 199.138.17.17
199.138.17.17 & 255.255.255.255 = 199.138.17.17, dus R8 is de next hop.
Vb. 2: bestemming 185.200.29.115
185.200.29.115 & 255.255.254.0 $\neq$ 185.200.24.0
185.200.29.115 & 255.255.254.0 $\neq$ 185.200.26.0
185.200.29.115 & 255.255.254.0 = 185.200.28.0, dus R3 is de next hop.
Vb. 3: bestemming 200.18.5.2
Alleen bij het laatste item is er gelijkheid: 200.18.5.2 & 0.0.0.0 = 0.0.0.0, dus R9 is de next hop.

Zou het item (0.0.0.0, 0.0.0.0, ...) niet voorkomen in de routetabel dan is er geen default router en dan zou er een foutboodschap verstuurd worden.

Routerconfiguratie

Om een router te configureren moet men voor zijn interfaces een IP-adres en een subnetmasker opgeven.

3.7 ICMP (Internet control message protocol)

Dit protocol is vereist voor elke TCP/IP-implementatie. ICMP specifieert:

- de structuur van de boodschappen

- hoe deze boodschappen moeten geïnterpreteerd worden

Een ICMP-boodschap heeft een eenvoudige structuur: een ICMP-hoofding en ICMP-data. Deze boodschap of ICMP pakket wordt verstuurd als datadeel van een IP-pakket.
Het ICMP wordt vooral gebruikt om fouten te signaleren : als een router een IP-pakket niet kan verder sturen, dan wordt een ICMP-boodschap naar de afzender gestuurd (tenzij de netwerkbeheerder deze functie zou afgezet hebben).
Om een stroom van ICMP-boodschappen te voorkomen, is afgesproken dat er geen foutmeldingen over foutmeldingen gestuurd worden. Dus als een router een IP-pakket niet kan verder sturen, dan wordt eerst nagegaan of het datadeel een ICMP-boodschap is. Indien niet, dan wordt een ICMP-boodschap gestuurd, indien wel, dan wordt geen ICMP-boodschap gestuurd.
Ter herinnering : de hoofding van een IP-pakket bevat een protocolnummer. Precies dit nummer geeft aan voor welke module (bvb. ICMP) een IP-pakket verstuurd wordt. De ICMP-hoofding bevat :
- type (1 byte) : zie de vb. hierna;
- code (1 byte) : d.i. extra informatie over het type;
- CRC (2 bytes).

De ICMP-data bevatten (in principe): de hoofding en de eerste 64 bytes van het datadeel van het IP-pakket waarvoor een ICMP-boodschap gestuurd wordt. Het kan ook iets anders zijn (één en ander hangt van het type af).
Het type kan bvb. zijn :
0 : antwoord op een echo-verzoek
3 : bestemming onbereikbaar
8 : echo-verzoek
enz...

Vb 1 Een machine kan, als ICMP-boodschap, een echo-verzoek (type nr 8) versturen. De afspraak is, dat een machine die een dergelijke boodschap ontvangt, antwoordt met een ICMP-boodschap antwoord op het verzoek (type nr 0). Een machine verstuurt een echo-verzoek, na een ping commando van een gebruiker.
Vb 2 Als een router een IP-pakket niet kan verder sturen, dan stuurt hij een ICMP-boodschap bestemming onbereikbaar (type nr 3) aan de afzender. Het codeveld vermeldt dan de oorzaak:
- 0 : netwerk onbereikbaar
- 1 : computer onbereikbaar
- 6 : netwerk van bestemmeling onbekend
- 7 : bestemmeling (computer) onbekend
- enz...

3.8 DHCP

Communiceren via TCP/IP vereist dat :

- deze protocol-suite geïnstalleerd is;
- de computers een IP-adres hebben (en een subnetmasker en een default router[7] kennen).

Dit IP-adres is informatie die in de computer moet aanwezig zijn. Er zijn een aantal redenen om het IP-adres NIET vast (bvb. op de harde schijf) in de computer op te slaan :

- de computer heeft geen harde schijf;
- de computer is mobiel en wordt in verschillende netwerken opgenomen;
- er zijn meer computers in het netwerk dan IP-adressen (bvb. meer dan 254 in een C-netwerk) en deze computers willen allen communiceren (hoogstens 254 tezelfdertijd);
- IP-adressen kunnen best centraal beheerd worden, bvb. door een server.

Als een computer geen IP-adres heeft, moet hij er één krijgen, bvb. tijdens het opstarten. Dit gebeurt door het aan te vragen aan een server. Vroeger werd hiertoe BOOTP (bootstrap protocol) gebruikt. Dit protocol liet NIET toe dat IP-adressen dynamisch toegekend worden; d.w.z. : een computer kan bij een volgende start geen ander IP-adres krijgen. BOOTP is opgevolgd door DHCP. Bij dit protocol worden een aantal IP-adressen ter beschikking gesteld door een server die door de netwerkbeheerder daartoe geconfigureerd werd. Computers (cliënten) kunnen dan van de server een adres ”huren”voor één sessie. De systeembeheerder heeft hier alle faciliteiten i.v.m. instellingen :

- aan sommige computers altijd hetzelfde adres geven,
- aan andere computers altijd het eerste adres dat vrij is, geven;
- een verhuurperiode specifiëren (bvb. in functie van de drukte).

DHCP specifieert de boodschappen die kunnen uitgewisseld worden. DHCP-boodschappen worden via UDP verstuurd (UDP is de niet-verbindingsgerichte tegenhanger van TCP). Goede vraag : hoe kan een computer UDP-berichten versturen als hij zijn IP-adres niet kent? Antwoord : er wordt een broadcastadres[8] gebruikt. D.w.z. : een computer verstuurt (na het opstarten) een aanvraag voor een IP-adres naar alle computers van het netwerk via een UDP-bericht (IP-pakket) naar poort 67. Alle computers lezen dit IP-pakket. Een DHCP server (er kunnen er meerdere zijn) behandelt UDP-berichten voor poort 67, en stuurt een antwoord. De bestemmingspoort van dit antwoord is 68. Het DHCP bevat afspraken zodat in alles voorzien wordt:

[7] Een breakout point voor het netwerk.
[8] Op welke laag?

- Wat als er 2 DHCP-servers een IP-adres willen aanbieden?

- Wat als 2 cliënten ongeveer tezelfdertijd een IP-adres aanvragen?

- Wat als de geldigheidsduur verstrijkt terwijl de computer nog aan het werken is?

- Enz...

3.9 IPv6

Netwerkbeheerders zetten nu de stap naar IP versie 6. In besturingssystemen is de mogelijkheid om versie 6 te gebruiken al aanwezig (bij de oudere versie was IPv6 een extra te installeren protocol). Zoals iedereen weet, bestaan IPv6-adressen uit 128 bits, d.i. 16 bytes of 8 keer 4 hexadecimale cijfers. Maar ook de hoofding van een IPv6-pakket is anders. Er wordt geen subnetmasker meer gebruikt, het aantal netwerkbits wordt expliciet aangegeven.

3.9.1 Adresnotatie

Een IPv6-adres wordt genoteerd als:

1A22:34CF:000C:6E2B:0123:CD01:7825:FA23

d.i. 8 keer 4 hexadecimale cijfers, gescheiden door een dubbel punt (E. : a colon). Als een groepje van 4 hexadecimale cijfers leidende nullen bevat, mogen deze weggelaten worden. Dus

1A22:34CF:C:6E2B:123:CD01:7825:FA23

is hetzelfde IPv6-adres als hoger. Eén opeenvolging van nullen mag vervangen worden door 2 dubbele punten

22CD:34CF:0:0:0:0:7825:0A23

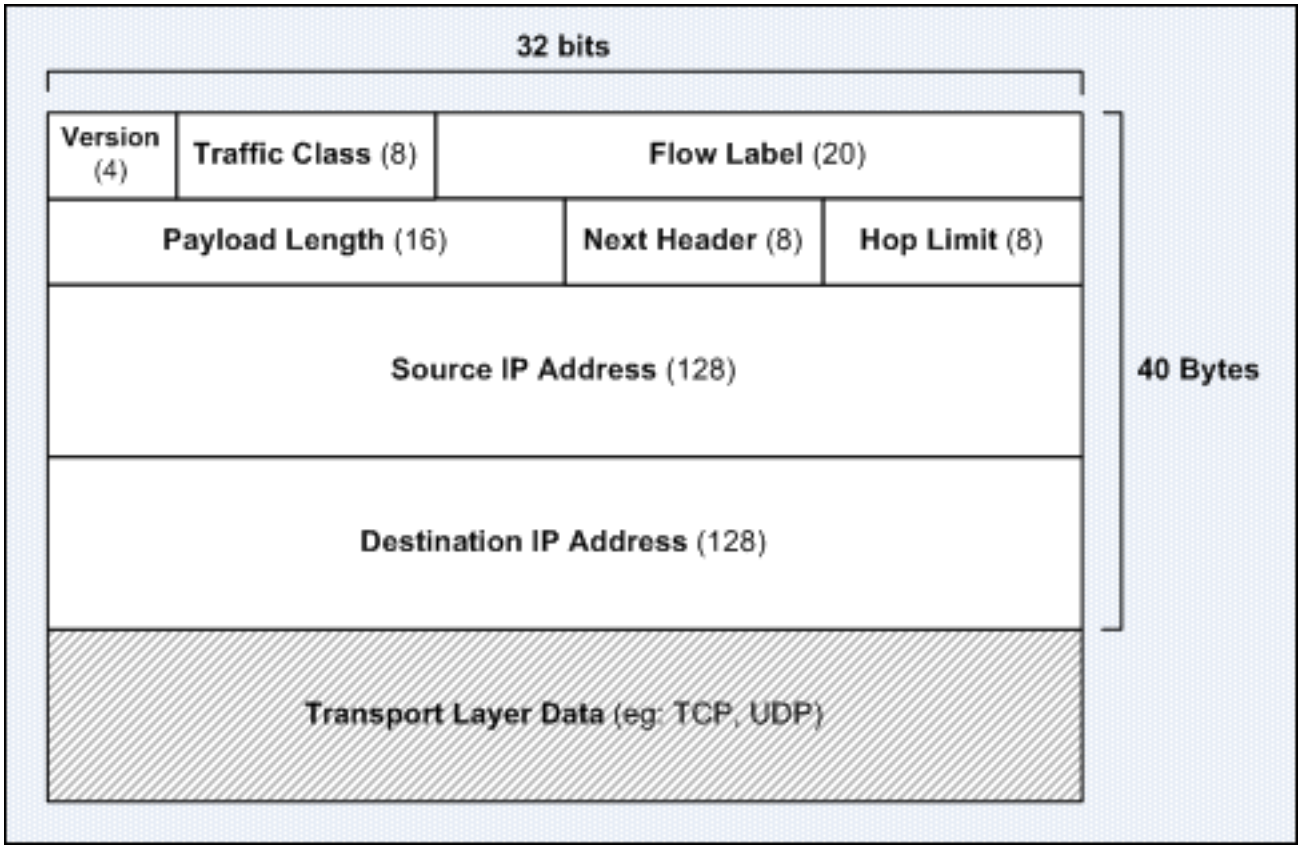
schrijft men korter als

22CD:34CF::7825:A23

3.9.2 IPv6-hoofding

De IPv6-hoofding bestaat altijd uit 40 bytes. Om –zoals bij versie 4- opties te gebruiken, worden extra hoofdingen toegevoegd. Deze komen na de IPv6-hoofding en vóór de data. De IPv6-hoofding bevat volgende velden:

Versie ½ byte. De waarde in dit veld is natuurlijk 6.



Trafiek klasse (traffic class): 1 byte. Dit veld komt overeen met het DS-veld van versie 4.

Stroomlabel (flow label): 2½ byte. Een flow is een stroom van IP-pakketten en wordt bvb. gegenereerd door één toepassing. Het zouden bvb. al de IP-pakketten voor één bestandsoverdracht kunnen zijn (of voor spraak, voor video). De afzender kan al deze pakketten hetzelfde stroom-label geven. Hierdoor weten de routers dat al deze pakketten op dezelfde wijze moeten behandeld worden.

Lengte (payload length), 2 bytes. De lengte in aantal bytes van de rest van het pakket (d.i. van hetgeen na de Ipv6-hoofding komt). Vermits hier-voor 16 bits voorzien zijn, kan de pakketlengte dus maximaal 65535 bytes zijn ($2^{16} - 1$). IPv6 is wel zo flexibel dat het met grotere pakketten kan werken (zie verder).

Volgende hoofding 1 byte. Dit veld geeft aan wat er na de IPv6-hoofding komt. Dit kan één van volgende zijn:
- de TCP-hoofding;
- de UDP-hoofding;
- de ESP-hoofding (Encapsulating Security Payload : wordt gebruikt bij

VPN's.);
- een extra IP-hoofding en die kan zijn:
* HbHO (hop-by-hop opties) hoofding,
* Fragment-hoofding,
* Routering-hoofding,
* Bestemmingsopties-hoofding.
Het veld "volgende hoofding"vervult dus dezelfde functie als het protocolveld bij IP4 maar het laat ook het gebruik van extra-opties toe.

Hop limiet 1 byte. Dit is het TTL-veld (time to live) van IPv4. Uit de praktijk bleek dat er nooit met seconden gewerkt werd maar altijd met het aantal routers. Elke router vermindert het veld met 1 en zendt een ICMP-bericht als de waarde 0 geworden is. Nu heeft het veld dus de juiste naam.

Adres afzender 16 bytes

Adres ontvanger 16 bytes

3.9.3 Fragmentatie

Het is niet langer mogelijk dat routers pakketten fragmenteren als op een netwerksegment de maximale frame-lengte te klein is (MTU, maximal transfer unit). Als een pakket te groot is, stuurt de router een ICMP-bericht naar de afzender. De afzender moet dus nagaan hoe groot de pakketten naar een bepaalde bestemming mogen zijn en kan geen pakketten sturen die te groot zijn. Hij moet zelf zijn pakketten fragmenteren. Als hij hiervan gebruik maakt, moet de extra hoofding fragment-hoofding gebruikt worden.

IPv6 adressen toekennen

Voorlopig verdeelt de IANA 1/8 van de adresruimte vastgelegd onder gebruikers. Anderzijds zijn er ook delen van de adresruimte gereserveerd voor specifieke toepassingen. De adressen die nu uitgedeeld worden zijn deze die beginnen met 2x en 3x dat is dus –bemerk de CIDR-notatie:

2000::/3

d.i.: alle IP-adressen waarvan de eerste 3 bits dezelfde zijn als van $2_x=0010_b$.

RIR (Regional Internet Registry) Van deze adresruimte zijn blokken toegekend aan de 5 RIR's (Regional Internet Registries). Deze organisaties kennen dan blokken van IP-adressen toe aan ISP's. De 5 RIR's zijn (via Google kan je een wereldkaart opvragen en zien voor welk gebied elke RIR bevoegd is):

RIP (Réseaux IP Européens, jaja, oui oui, en français; maar de volledige naam is RIPE NCC, van Network Coordination Centre) : Europa, het Midden-Oosten, Noord-Azië (van de Oeral tot Vladivostok en omstreken);

ARIN (American Registry for Internet Numbers) : Noord-Amerika;

LACNIC (Latin American and Caribbean Internet Addresses Registry): Latijns-Amerika (van Mexico tot Ushuaia en omstreken);

APNIC (Asia-Pacific Network Information Centre) : China, India, tot Australië;

AfriNIC (African Network Information Centre) : Afrika.

Als we via Google "IPv6 Global Unicast Address Assignmentsöpvragen, zien we bvb. (op : http://www.iana.org/assignments/ ipv6-unicast-addressassignments/ ipv6-unicast-address-assignments.xml):
2001:0000::/23 IANA
2001:0200::/23 APNIC
2001:0400::/23 ARIN
2001:0600::/23 RIPE NCC
2001:1200::/23 LACNIC
2001:4200::/23 AfriNIC
...

De grootte van deze blokken is /23 (aha, hoe kan /23"de grootte van een blok IP-adressen voorstellen?). Wat betekent bvb. "2001:0600::/23"? Alle IP-adressen waarvan de eerste 23 bits gelijk zijn aan die van 2001:0600, d.i. alle IP-adressen die beginnen met : 2001:06 of 2001:07. Soms worden opeenvolgende blokken van grootte /23 toegekend, zo bvb. :
2001:3800::/22 RIPE NCC
d.w.z. de blokken 2001:3800::/23 én 2001:3A00::/23 of alles wat begint met 2001:38, 2001:39, 2001:3A, 2001:3B.

ISP (Internet Service Provider) Elke RIR gaat zijn /23-blokken van IP-adressen verder opdelen en toekennen aan de ISP's. De grootte van de blokken is nu /32. Ook deze toewijzing kunnen we opvragen: Google en "IPv6 ISP's"levert bvb. :
BIT BV 2001:7B8::/32
LeaseWeb 2001:1AF8::/32
Nxs Internet B.V. 2001:14A0::/32
...
Siemens IT-Dienstleistung und Beratung GmbH 2A02:8A0::/32 ...

De juiste interpretatie van bvb. de eerste lijn is : alle ip-adressen die beginnen met 2001:07B8 (dit zijn 32 bits) zijn toegekend aan BIT BV. **Vraag**: hoeveel /32-blokken gaan er in één /23-blok?

Bedrijven Elke ISP kent aan zijn klanten /48-blokken van IP-adressen toe. Als we surfen naar de site van Belnet (http://ipv6.belnet.be) zien we bvb.:
IMEC 2001:06a8:29a0::/48
IMELDA 2001:06a8:0b30::/48
IPC 2001:06a8:0b90::/48
IWT 2001:06a8:06c0::/48
JESSA 2001:06a8:20e0::/48
KHLEUVEN 2001:06a8:2880::/48

Dus, alle IP-adressen die beginnen met 2001:06a8:2880 (dit zijn 48 bits) zijn toegekend aan de UCLL. **Vraag**: hoeveel /48-blokken gaan er in één /32-blok?
Samenvatting: van een IPv6-adres:

- identificeren de eerste 48 bits de ISP (die zijn IP-adressen heeft van een RIR)
- worden de laatste 80 bits toegekend door het bedrijf

Opmerking: een thuisgebruiker zal van zijn ISP wel geen /80-netwerk-adres krijgen (om dan te subnetten) maar slechts 1 /64-netwerk.

3.9.4 IPv6 subnetten

Aan een bedrijf wordt een /48-blok van IP-adressen toegekend. Er zijn dus 80 bits vrij te kiezen. Het is de bedoeling dat een bedrijf de eerste 16 bits gebruikt om het subnet aan te duiden. De laatste 64 bits identificeren dan het apparaat in het subnet. De eerste 64 duiden het subnet aan. De eerste 64? Ja, de 48 die toegekend zijn door de ISP samen met de 16 subnetbits die het bedrijf zelf gekozen heeft.

3.9.5 Soorten IPv6 adressen en hun bereik

Eén apparaat kan meerdere IP-adressen krijgen. Het bereik is niet hetzelfde (zie verder). Het type van een adres kan een van volgende drie zijn : unicast, multicast of anycast[9]. Let op: IPv6 heeft geen broadcast adressen meer.

[9]Wordt niet besproken in de cursus.

Unicast IPv6 adressen

Er bestaan verschillende unicast IPv6 adressen: globaal unicast, lokale link, uniek lokale, e.a. We bespreken hier enkele.

Globaal unicast Dit type adres is het adres zoals beschreven hierboven: als de bestemmeling van een pakket zich buiten het subnet bevindt en als er een correct geconfigureerde router is, zal deze het pakket op een ander subnet verder sturen.
De eerste 3 bits zijn 001; dan volgt :

- de zgn. global routing prefix (45 bits)
- subnet-id (16 bits)
- host-id

Lokale link

- Iedere IPv6-enabled netwerk interface vereist een lokale link IPv6 adres
- deze adressen beginnen met FE8, FE9, FEA of FEB (d.i. FE80::/10)
- de andere bits worden meestal afgeleid van het MAC-adres
- IP-pakketten met een dergelijk adres als bestemming worden door een router niet op een ander subnet verder gestuurd
- deze adressen dienen om te communiceren met apparaten op het subnet
- deze adressen worden gebruikt voor automatische adresconfiguratie en om vast te stellen welke andere apparaten er zijn zoals bvb een router.

Uniek lokale

- deze adressen zitten in de range FC00::/7;
- IP-pakketten met een dergelijk adres als bestemming worden door een router op een ander subnet verder gestuurd maar alleen als dit een subnet is binnen dezelfde site;
- een aparte toepassing van deze adressen : als er een Microsoft IPv6- besturingssysteem geïnstalleerd is, wordt er automatisch gezocht naar een DNS-server op de adressen : FEC0:0:0:FFFF::1, FEC0:0:0:FFFF::2, FEC0:0:0:FFFF::3.
- deze adressen kunnen beschouwd worden als het IPv6 equivalent van RFC1918 private IPv4 adressen.

3.9.6 Configuratie van een global unicast adres

De configuratie van een global unicast adres kan op verschillende manieren gebeuren.

1. Statisch: je geeft manueel als netwerkadministrator het IP-adres in.
2. Dynamisch: het IP-adres wordt dynamisch toegewezen of gecreëerd. Dit kan op twee verschillende manieren:
    - SLAAC[10]: door middel van ICMPv6 Router Advertisement (RA) berichten komt het device zijn prefix, prefix length en default gateway te weten van de IPv6 router op zijn subnet. Er wordt geen gebruik gemaakt van een DHCPv6 server. Alleen prefix? Waar komt de rest van het ipv6-adres vandaan? eui-64 (zie verder) of at random gegenereerd.
    - SLAAC + DHCPv6: het adres wordt verkregen via SLAAC. Additionele informatie wordt verkregen via een DHCPv6 server, zoals de DNS server bijvoorbeeld.
    - DHCPv6 all the way, door middel van ICMPv6 Router Advertisement (RA) berichten komt het device te weten dat het op zoek moet gaan naar een DHCPv6 server. Deze zal alle nodige informatie verschaffen.

EUI-64 Process Bij het opgeven van een IPv6-adres volstaat het om alleen de eerste 64 bits op te geven. De laatste 64 worden afgeleid van het MAC-adres. Als het MAC-adres bvb **00:26:B9:8C:9D:AE** is dan wordt in het midden FF FE tussengevoegd. Bvb. na (bij configuratie van een router-interface en met bovenstaand Mac-adres) via *ipv6 address 2001:07a9:2880::/64 eui-64*
wordt het IP-adres: 2001:07a9:2880:0:0026:B9**FF:FE**8C:9DAE

3.9.7 Configuratie van een link lokale adres

Het link lokale adres wordt dynamisch gecreëerd gebruik makende van de prefix FE80::/10 en de interface ID of MAC-adres d.m.v. eui-64 of het wordt at random gegenereerd.

Multicast IPv6 adressen

IPv6 multicast adressen hebben de prefix FF00::/8. Er bestaan verschillende type multicast adressen. We beperken ons tot twee types:

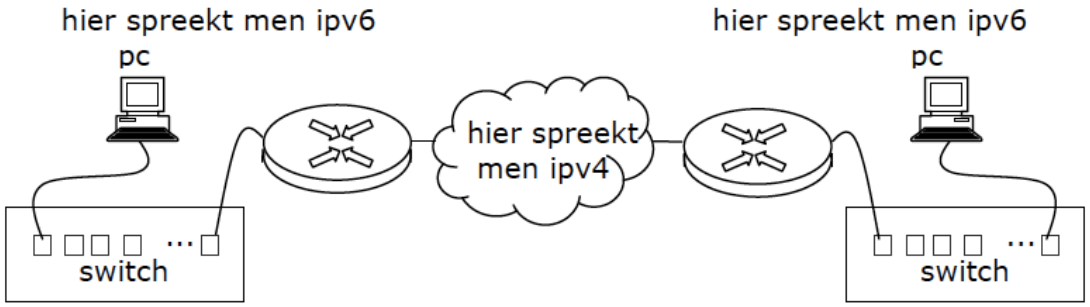
[10]Stateless Address AutoConfiguration

- Assigned multicast: dit zijn adressen die toegewezen zijn aan specifieke functies zoals DHCPv6 server, ed. FF02::1 is een multicast adres equivalent aan een IPv4 broadcast ook all-nodes multicast adres genoemd.
- Solicited node multicast: dit is analoog aan de all-nodes multicast addres en komt alleen met de laatste 24 bits van het IPv6 unicast address van het device overeen. Het wordt automatisch gegenereerd wanneer een unicast adres is geconfigureerd. Dergelijk adres wordt bekomen door FF02:0:0:0:0:1:FF00::/104 met de 24 bits rechts van het unicast adres. Dit adres wordt bij IPv6 gebruikt voor het opvragen van mac-adres met behulp van het ICMPv6 Neighbor Discovery Protocol.

3.9.8 IPv4 en IPv6 samen

Een pc waarop alleen IPv4 geïnstalleerd is en een pc waarop alleen IPv6 geïnstalleerd is, kunnen niet communiceren met mekaar. Er is dus een probleem omdat het ondoenbaar is om alle apparatuur (pc's maar ook routers) in een oogwenk om te schakelen. Er zijn 2 oplossingen: dual stack (=dubbele stapel) en het gebruik van een tunnel.

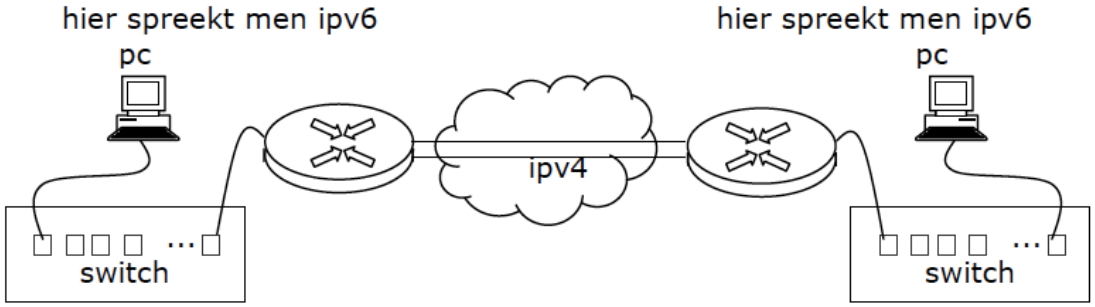
Dual stack Zowel het IPv4 als het IPv6 software worden geïnstalleerd en geconfigureerd. Met een IPv4-apparaat worden IPv4-pakketten uitgewisseld en met een IPv6-apparaat worden IPv6-pakketten uitgewisseld. Indien nodig wordt het data-deel uit het ene pakket gelicht en opnieuw ingepakt in een pakket van een andere versie.



Pc A verstuurt IPv6-pakketten naar pc B. De eerste router met zijn dual stack neemt de data uit dat pakket en plaatst die data in een IPv4-pakket. De 2° router doet de omgekeerde bewerking en pc B ontvangt de data die pc A verstuurd heeft.

Tunnel De vorige oplossing is niet altijd goed, omdat het oorspronkelijk IPv6-pakket niet bij pc B terecht komt. Ten minste toch de hoofding niet. De hoofding van een IPv6-pakket bevat een aantal nieuwe mogelijkheden,

bvb. het gebruik van het flow-veld en door de omzetting naar IPv4 verliezen we juist dat soort van IPv6- voordelen. Bij de tunnel-oplossing zal de eerste router het ganse IPv6-pakket inpakken als data in een IPv4-pakket en dit pakket over het IPv4-netwerk versturen. De 2° router ontvangt dit IPv4-pakket, haalt er het oorspronkelijk IPv6-pakket uit en verstuurt dit pakket.



Een dergelijke tunnel wordt ISATAP-tunnel genoemd (intrasite automatic tunnel addressing protocol). Nog een advies vanwege Cisco aan netwerk-beheerders : dual stack waar je kunt, tunnel waar je moet.

Stapel? Men heeft het over de IPv4- en de IPv6-stapel. Er toch even op wijzen dat het niet de ganse protocolstapel is die anders is, maar alleen het netwerkprotocol. Aan toepassingen (www, e-mail, enz...) hoeft er niets te veranderen en bvb. ook niet aan de Ethernet-kaart of het Ethernet-protocol. Wat we nu meemaken is een goede illustratie van de voordelen van de lagenstructuur (OSI of TCP) : in één laag wordt iets veranderd (en wel zeer grondig) maar op de andere lagen heeft dit geen invloed.

Hoofdstuk 4

Transportlaag

De netwerklaag zorgt voor een end-to-end pakketlevering tussen een broncomputer en bestemmingscomputer. De transportlaag maakt gebruik van de services of diensten van de netwerklaag om datatransport te leveren tussen een proces of programma op een broncomputer en een proces of programma op een bestemmingscomputer.

Deze laag biedt de abstracties die de toepassingen of processen nodig hebben om het netwerk te gebruiken. Applicatieontwikkelaars kunnen dus software schrijven op basis van standaard transportlaag API's en er zich van vergewissen dat de applicaties op verschillende type netwerken zullen functioneren.

4.1 Diensten die de transportlaag levert

In deze paragraaf zullen we beschrijven welke "bewerkingenöf ächties"de transportlaag uitvoert voor zijn gebruikers, nl. de applicaties. Iedere transportlaag of transportdienst heeft zijn eigen interface.

Machines waarop de IP-programmatuur geïnstalleerd is, kunnen IP-pakketten naar mekaar sturen. Maar IP levert geen garantie op correcte aflevering van de pakketten. Om bruikbaar te zijn voor bepaalde toepassingen (www, e-mail, enz...) is het nodig dat er een betrouwbare verbinding is. Er is dus nog extra programmatuur nodig. Deze programmatuur bevindt zich in de laag boven de netwerklaag nl. **de transportlaag**.
Voor deze laag zijn er 2 protocollen: **UDP (user datagram protocol)** en **TCP (transmission control protocol)**. Men zou natuurlijk een doe-het-zelf-toepassing kunnen ontwikkelen die geen gebruik maakt van TCP of UDP en zelf de functies van de transportlaag op zich neemt.

Protocolveld in de hoofding van een IP-pakket De IP-module vervult volgende taken:
- een gegeven IP-pakket naar een gegeven IP-adres sturen
- uit een inkomend IP-pakket het datadeel nemen en doorgeven.

Dit laatste roept meteen de vraag op: aan wie of wat wordt dit datadeel afgeleverd? Er zijn meerdere kandidaten:
- de ICMP-module (ICMP maakt deel uit van de IP-laag);
- de transportlaag: UDP of TCP;
- eventueel een toepassingsprogramma.

De hoofding van een IP-pakket bevat een protocol-veld, dit veld duidt het protocol aan waarvoor het IP-pakket verstuurd wordt; bijvoorbeeld 17 voor UDP of 6 voor TCP. In de lagen boven de IP-laag kunnen er meerdere processen zijn die tezelfdertijd de IP-module gebruiken. Sommige processen gebruiken IP via UDP andere via TCP. Nog andere kunnen IP rechtstreeks gebruiken. Via het protocol-veld weet de IP-module of inkomende data moeten afgeleverd worden aan de UDP-module, de TCP-module of een ander programma.

Poortnummers Meerdere processen kunnen **tezelfdertijd** de UDP-module gebruiken. De UDP-module moet dan ook deze processen van mekaar kunnen onderscheiden en inkomende data afleveren aan het juiste proces waarvoor ze bestemd zijn. Hetzelfde geldt voor de TCP-module. Processen worden van mekaar onderscheiden door **poortnummers**. **De TCP- (of UDP-)module houdt per applicatie of proces een wachtrij voor inkomende (en uitgaande) data bij.** Een proces kan data uit deze wachtrij lezen. Als de data sneller toekomen dan ze gelezen worden, dan komen ze in de wachtrij terecht. Als een proces wil lezen vooraleer er data aangekomen zijn, wordt het in de wachttoestand geplaatst tot de data er zijn.

4.2 UDP (user datagram protocol)

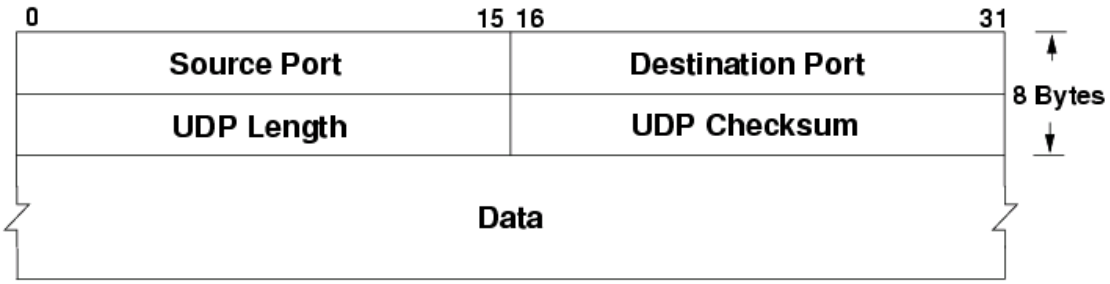
Zoals IP levert ook UDP een onbetrouwbare dienst: berichten kunnen verloren gaan of ze kunnen fouten bevatten. Als een zender meer dan 2 berichten naar dezelfde bestemmeling stuurt, kunnen ze in de verkeerde volgorde aankomen.
Aan IP voegt UDP volgende functies toe :
- de mogelijkheid te werken met controlebytes;
- het gebruik van poortnummers, wat multiplexing toelaat, in die zin dat meerdere processen de IP-module tezelfdertijd kunnen gebruiken.

Een programma kan communiceren met een programma op een andere machine via UDP. Vermits UDP geen betrouwbaar transport biedt, zullen de communicerende programma's zelf foutcontrole en -behandeling moeten verzorgen. Het voordeel van UDP is dat er weinig extra belasting of overhead is. Daarentegen kunnen programma's ook TCP als transportmiddel gebruiken. TCP is wel betrouwbaar (maar bevat dan ook de complexiteit die daartoe vereist is). Een andere reden om UDP te gebruiken i.p.v. TCP is broadcasting. Via UDP kan dit wel en via TCP niet.

Enkele UDP-poorten	
7	Echo
18	Message Send Protocol
53	Domain Name Server
67	DHCP-cliënt
68	DHCP-server
69	Trivial File Transfer
92	Network Printing Protocol
115	Simple File Transfert Protocol
123	Network Time Protocol
137	NETBOIS Name Service
144	News
161	SNMP
194	Internet Relay Chat Protocol
520	RIP

Structuur van een UDP-bericht Bemerk de gebruikte termen:
UDP : bericht of datagram
netwerklaag (IP): pakket
onder de netwerklaag : frame
Een UDP-bericht bestaat uit : hoofding, datadeel
De hoofding bevat telkens 2 bytes voor:
- bron : poortnummer van het verzendende proces;
- bestemming : poortnummer van het ontvangende proces;
- lengte : aantal bytes in het bericht;

- controle-veld (checksum): controle-bytes.[1]



4.3 TCP (transmission control protocol)

Een TCP-verbinding tussen 2 processen is een virtueel circuit. D.w.z. dat de verbinding eerst tot stand moet gebracht worden en er daarna een soort pijp is tussen de 2 processen. De programma's die TCP gebruiken, kunnen er zeker van zijn dat er een bytestroom van de ene machine naar de andere loopt en omgekeerd: **TCP werkt full duplex**. Vermits IP de correcte volgorde van de ontvangen pakketten niet garandeert, moet TCP de volgorde herstellen. Een programma levert bytes af aan de TCP-module, zoals het met write bytes naar een bestand stuurt. (Ook voor open en close is er een gelijkaardige bewerking bij TCP). De transportentiteit regelt de TCP-segmentstromen. Een TCP-transportentiteit of TCP-module neemt datastromen van een gebruikersproces in ontvangst, verdeelt ze in blokken van maximum 64 kbytes[2] en verstuurt elk blok apart in een IP-datagram of IP-pakket.

[1]De berekening van de checksum valt buiten de scope van deze inleidende cursus. For the hardliners: dit controle-veld is een beetje speciaal: 0x0000 wil zeggen dat er geen controle gebeurd is. Als er wel controle gebeurt, en als de controle-bytes toevallig 0000 zijn, wat doet men dan? Wie denkt dat dit een ernstig probleem is, rekent buiten de waard: er wordt met 1-complement voorstelling gewerkt. In deze voorstelling zijn er 2 coderingen voor 0. Als er wel controle gebeurt, en de controlebytes zijn toevallig 0, dan gebruikt men voor de controle-bytes de andere code voor 0(!) Het IP-adres bepaalt samen met het poortnummer het proces. Om de inhoud van het controle-veld te berekenen:
- wordt aan het UDP-bericht een pseudo-hoofding toegevoegd; deze bestaat uit : IP-adres van bron en bestemming, het protocolnummer (17);
- wordt het controle-veld voorlopig op 0x0000 gezet.

Om bij de ontvangende UDP-module toe te laten het controle-veld te berekenen, moet de IP-module naast het UDP-bericht ook het IP-adres van de afzender doorgeven. Indien het een machine is die meer dan één IP-adres heeft, dan moet de IP-module doorgeven via welk IP-adres het bericht is binnengekomen.

[2]Vaak wordt de gebruikersgegevensstroom opgedeeld in blokken van 1460 bytes, zodoende kan de data samen met de TCP- en IP-header net passen in een Ethernet frame

Bytestroom Een TCP-verbinding is een bytestroom en geen berichtenstroom. De TCP-module heeft geen idee wat de bytes betekenen en wil dat ook niet weten. De TCP-module plaatst de ontvangen bytes in een buffer. Als er voldoende bytes zijn, wordt een TCP-segment naar de andere machine gestuurd (d.w.z.: afgeleverd aan de eigen IP-module). De andere machine bewaart inkomende bytes in een buffer (als deze bytes correct zijn, en nadat de volgorde eventueel hersteld is). Een programma kan bytes lezen uit deze buffer. Waar er bij UDP geen verbinding is (een proces stuurt een datagram naar een ander proces), werkt TCP wel met verbindingen, het virtueel circuit zoals we het hierboven hebben genoemd. Deze verbinding wordt bepaald door twee eindpunten. Eén eindpunt wordt **socket** genoemd en heeft een socketnummer of beter een socketadres. Dit socketadres bestaat uit twee delen:

- Het IP-adres van de host
- Een 16 bit getal dat lokaal is aan de host, een **poort** genaamd

Een verbinding is bvb.: [(192.18.16.7,317);(180.17.126.5,25)]. Als de verbinding tot stand gebracht is, kunnen segmenten in beide richtingen gestuurd worden. Is deze verbinding een point-to-point[3] verbinding?
Een socket kan voor verschillende verbindingen tegelijkertijd gebruikt worden. Denk maar aan een web server die diverse verbindingen actief heeft op poort 80, allemaal met verschillende clients. Voor al deze clients eindigen hun verbinding met dezelfde socket.

4.3.1 Het TCP protocol

- Elke byte in een TCP verbinding heeft zijn eigen 32-bit volgnummer.
- De zendende en de ontvangende TCP-modules wisselen data uit in de vorm van segmenten.
- Een TCP-segment bestaat uit een header van 20 bytes gevolgd door nul of meerdere databytes.
- De databytes in één segment kunnen worden gecombineerd uit meerdere schrijfacties naar de socket of omgekeerd. De TCP-module beslist doorgaans hoe groot de segmenten zijn.
- Opdracht: Zoek uit wat de link is tussen **MSS** en **MTU**.
- Opdracht: Bewijs dat een TCP segment maximaal 65495 bytes mag bevatten.

[3]Point-to-point wil zeggen dat de verbinding precies twee eindpunten heeft.

Betrouwbare dienst Een eindpunt van een TCP-verbinding moet na ontvangst van een segment de foutcontrole uitvoeren en een ACK (bevestiging) sturen naar het andere eindpunt. Is er een fout dan wordt er geen ACK gestuurd. De bevestiging vermeldt het volgnummer van de byte die verwacht wordt. **Bijvoorbeeld x, dit wil dan zeggen dat tot en met x-1 alles correct ontvangen is.** Dit betekent meteen: als een segment correct, maar vóór zijn beurt aankomt, dan kan de correcte ontvangst ervan nog NIET bevestigd worden omdat de bevestigingen gerelateerd zijn aan de verzonden bytes en niet aan segmenten.
Er wordt NIET gewerkt met foutmeldingen. Als bij een zender de bevestiging voor een verstuurd segment uitblijft, dan wordt het segment opnieuw verstuurd. Hiertoe wordt een vooraf vastgelegde wachttijd (E. : timeout) gehanteerd. De grootte van deze wachttijd kiezen, is een probleem op zich. De wachttijd moet zowat gelijk zijn aan de tijd die er normaal nodig is om een segment te versturen en een bevestiging terug te sturen. Men noemt dit de HTT (heen- en terugtijd, E. : round trip time(**RTT**)). Uiteraard is deze tijd korter op een LAN dan op een WAN. Bovendien speelt de belasting op het netwerk ook een rol (IP-pakketten moeten langer wachten in routers bij zwaardere belasting). Het blijkt dat de HTT een exponentiële verdeling heeft. TCP-implementaties schatten voortdurend de gemiddelde HTT. Als wachttijd wordt dan bvb. 2 keer de gemiddelde HTT genomen.

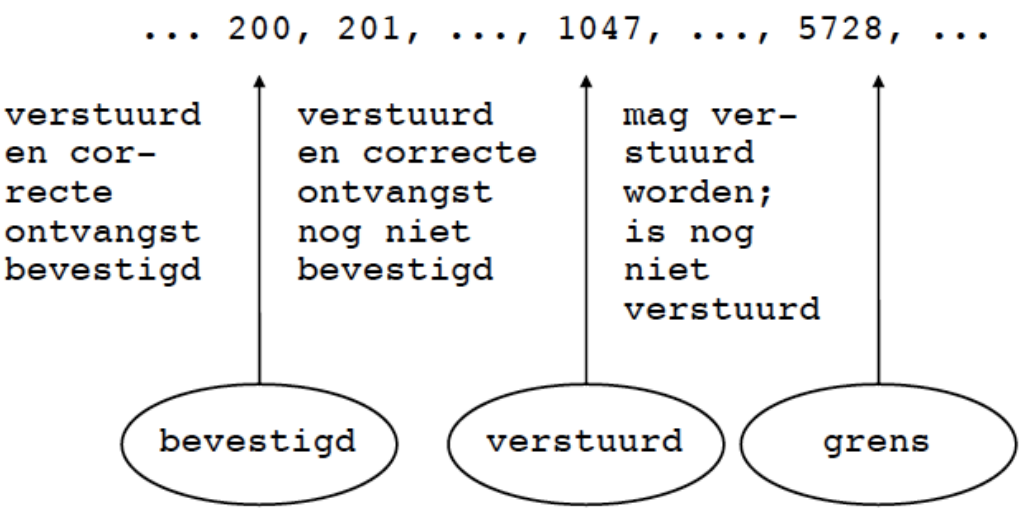
Venster Het verzendende proces levert bytes af aan de TCP-module. Het ontvangende proces leest deze bytes in de volgorde waarin ze afgeleverd zijn. Wanneer een zender een segment stuurt, start hij ook een timer. Wanneer het segment bij de ontvanger aankomt zendt deze een segment, met data als die er zijn ander zonder data, terug met de bevestigingsnummer van de volgende byte die hij van de zender verwacht en de resterende venstergrootte (zie verder). Als de timer verloopt voordat de bevestiging is ontvangen, wordt het segment opnieuw verzonden. Aan de zendende zijde houdt TCP een rij bytes bij. Bvb.:

...,200, 201, 202, ...

Bytes die de TCP-module heeft verzonden, worden aan de rij toegevoegd. Verder houdt TCP 3 wijzers (E. : pointers) bij. De rij bestaat uit 3 delen, zie figuur 4.1.
De bytes die:

- verstuurd zijn en waarvan de (correcte) aankomst nog niet bevestigd is
- de bytes die mogen verstuurd worden maar nog niet verstuurd zijn (om welke reden dan ook)

vormen samen het **venster**. De wijzers **bevestigd** en **grens** begrenzen het venster. Er wordt begonnen met een venster met een initiële grootte. De



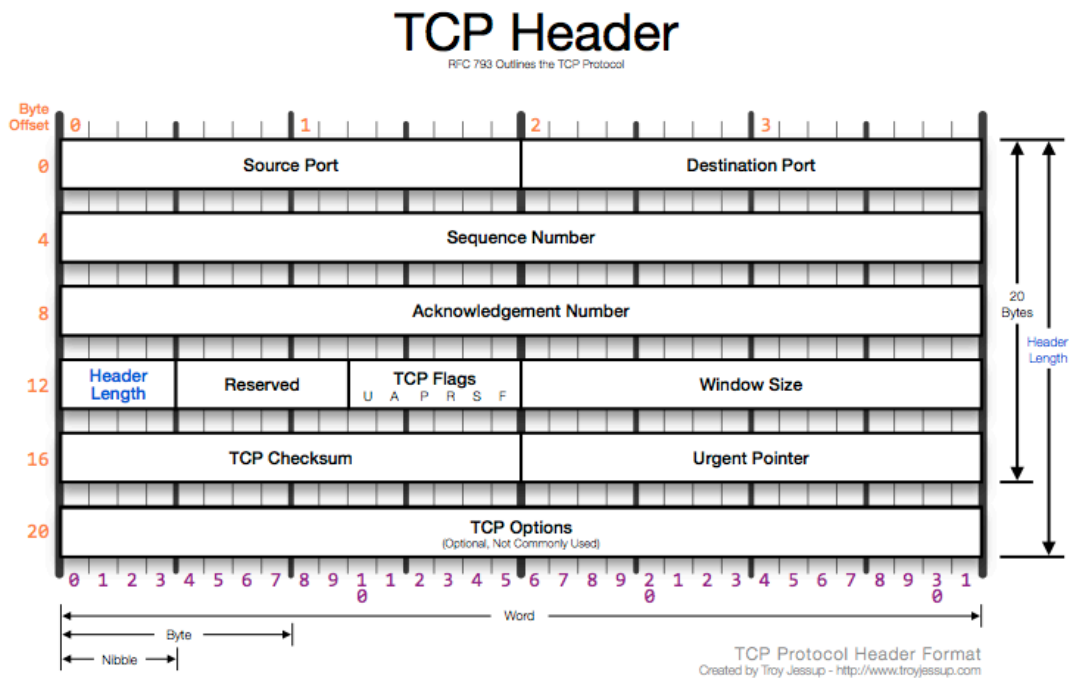
Figuur 4.1: TCP pointers

bytes tot vóór bevestigd, worden niet langer bewaard. De TCP-module verstuurt bytes zolang verstuurd < grens. Als er bvb. 400 bytes verstuurd worden, krijgt **verstuurd** de waarde 1447. Als er een bevestiging komt, wordt **bevestigd** aangepast. In het voorbeeld zou de ontvangende TCP-module bvb. kunnen melden dat de bytes tot en met byte nr. 500 correct ontvangen zijn; in dit geval zal bevestigd de waarde 500 krijgen en ook grens zal met 300 opgehoogd worden. Een bevestiging bevat ook een mededeling over de grootte van het venster: verkleinen, vergroten. Het venster dient voor stroomcontrole (E. flow control). De zender kan zoveel bytes sturen als het venster groot is. De ontvangende zijde moet die kunnen opslaan. Door het venster kleiner te maken kan men de zender afremmen.

Poorten Zoals UDP, gebruikt TCP poorten voor de communicatie met de toepassingen.

TCP-segment Het bericht of PDU dat door een TCP-module verstuurd wordt, heet segment.
Elk TCP segment begint met een hoofding van 20 bytes. Na deze verplichte hoofding van 20 bytes kunnen extra TCP opties volgen. We bekijken nu de verschillende velden van de hoofding.

- poortnummer van het zendend proces
- poortnummer van het proces dat de bestemmeling is
- bytevolgnummer (4 bytes) : dit is het volgnummer in de totale bytestroom van de 1° byte in het datadeel van dit segment (in bovenstaand voorbeeld zou in het volgende segment dat verstuurd wordt, dit nummer gelijk zijn aan 1048)



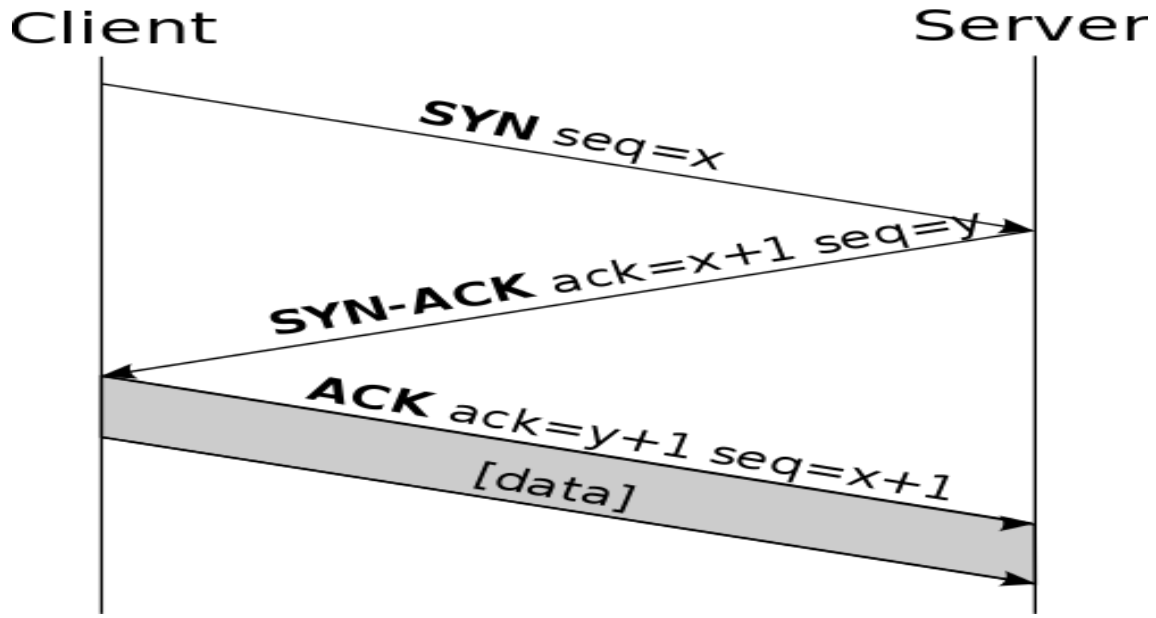
Figuur 4.2: De TCP-header

- lengte van de hoofding: dit getal geeft aan uit hoeveel woorden van 32 bit de header bestaat, deze lengte is uiteraard afhankelijk van de eventuele opties die vermeld worden
- bevestigingsnummer: dit is het volgnummer van de byte die verwacht wordt van de datastroom in de andere richting. De inhoud van dit veld heeft alleen betekenis als ACK=1 (zie verder)
- zes één bit TCP-vlaggen:
  - URG = 1 : er zijn dringende data en de urgent pointer is in gebruik. De pointer geeft aan tot waar t.o.v. het volgnummer in het segment de urgente data staat.
  - ACK = 1 : het bevestigingsnummer heeft betekenis. In de hoofding staat er altijd iets op de plaats van het bevestigingsnummer. Alleen als deze vlag actief is, is het een ëcht"bevestigingsnummer.
  - PSH = 1 : de data wordt meteen verstuurd als gevolg van een push-bevel aan de TCP-module. De transmissie van de data wordt dus niet uitgesteld. De ontvangende module dient de data meteen aan de applicatie af te leveren en niet te bufferen.

  - RST = 1 : dit is een reset-segment en zorgt voor een reset van een verbinding die in de war is geraakt.
  - SYN = 1 : dit is een SYN-segment en wordt gebruikt om een verbinding op te zetten.
  - FIN = 1 : dit is een FIN-segment, gebruikt om een verbinding op te heffen omdat de zender van dit segment geen data meer te versturen heeft.

- venstergrootte : hiermee wordt de venstergrootte aangegeven die de correspondent mag gebruiken voor de datastroom in de andere richting.
- controlebytes : deze worden berekend zoals bij UDP, ook met een pseudohoofding die de IP-adressen bevat.
- lengte van het segment
- (eventueel) opties: dit veld schept de mogelijkheid extra functionaliteiten toe te voegen aan het TCP-protocol die er initieel niet in voorzien waren. Enkele voorbeelden zijn MSS, windows-scale-optie, PAWS, SACK.

Een verbinding tot stand brengen Een proces kan beginnen te communiceren met een proces op een andere machine, als dit laatste proces bereid is om te communiceren. Een proces verklaart zich bereid om op verzoeken tot communicatie in te gaan door het uitvoeren van een passieve open-functie, zie puntje 4.3.2. Een passieve open-functie uitvoeren betekent: signaleren aan de eigen TCP-module dat men bereid is te communiceren. Het besturingssysteem kent dan een poortnummer toe voor dit proces. Het eindpunt (IP-adres, poortnummer) bevindt zich in luister-toestand. Een proces op een andere machine kan dan een **actieve open-functie** uitvoeren. D.w.z. een verzoek richten aan de TCP-module om een verbinding tot stand te brengen met een luisterend eindpunt. Het besturingssysteem zal een poortnummer toekennen. De TCP-module zal een SYN-segment naar het andere eindpunt sturen. Een SYN-segment heeft de SYN-bit = 1 en heeft als bytevolgnummer een willekeurig getal n. De TCP-module die een dergelijk segment ontvangt, stuurt een SYN/ACK-segment.
Een dergelijk segment heeft de SYN-bit op 1 gezet, heeft als bytevolgnummer een willekeurig getal m, heeft de ACK-bit op 1 gezet en heeft als bevestigingsvolgnummer n+1. Een SYN-segment neemt dus 1 byte in beslag in de reeks van verstuurde bytes. Hierdoor kan het ondubbelzinnig worden bevestigd. B geeft te kennen dat hij data verwacht genummerd vanaf byte n+1.
Hierop antwoordt A met een ACK-segment. Een dergelijk segment heeft de ACK-bit =1 en heeft als bevestigingsvolgnummer: m+1, d.w.z. A geeft te kennen dat het data verwacht vanaf byte m+1, zie figuur 4.3.



Figuur 4.3: Het opzetten van een TCP-verbinding m.b.v. een TCP three-way-handshake

Aldus zijn aan weerszijden de zend- en ontvangstteller geïnitialiseerd.
Bij A:
- de bytes die ik ga versturen hebben volgnummers : n+1, n+2, n+3, ...
- de bytes die ik moet ontvangen van u, hebben volgnummers : m+1, m+2, ...
Bij B :
- de bytes die ik ga versturen hebben volgnummers : m+1, m+2, ...
- de bytes die ik moet ontvangen van u, hebben volgnummers : n+1, n+2, ...
De verbinding kan nu full duplex gebruikt worden.

Een verbinding gebruiken Nemen we aan dat hierboven n en m beide de waarde 0 hebben (in werkelijkheid zijn het willekeurig gekozen getallen). Dan zouden volgende gebeurtenissen kunnen plaatsvinden.
- A verstuurt een segment, bytes met nummers 1 t/m 500; verwacht byte 1 (bevestigingsnummer = 1);
- A verstuurt een segment, bytes met nummers 501 t/m 800; verwacht byte 1 (bevestigingsnummer = 1);
- A verstuurt een segment, bytes met nummers 801 t/m 1200; verwacht byte 1 (bevestigingsnummer = 1);
- B verstuurt een segment, bytes met nummers 1 t/m 600; verwacht byte 801 (bevestigingsnummer = 801);
- A ontvangt (correct) dit segment vanwege B en weet dat zijn bytes tot en met 800 correct ontvangen zijn en past zijn wijzers aan;

- A verstuurt een segment, bytes met nummers 1201 t/m 1700; verwacht byte 601 (bevestigingsnummer = 601);
- Bij A verstrijkt de wachttijd voor het 3° segment (bytes 801 t/m 1200)
- A stuurt het segment opnieuw, bytes met nummers 801 t/m 1200; verwacht byte 601 (bevestigingsnummer = 601);
- B verstuurt een segment, bytes met nummers 601 t/m 1300; verwacht byte 1201 (bevestigingsnummer = 1201);
- A ontvangt (correct) dit segment vanwege B en weet dat zijn bytes tot en met 1200 correct ontvangen zijn;
- Enz...

Stroomregulering Stroomregulering wordt geregeld door een **sliding window** of verstelbaar venster van variabele grootte. De waarde van het window geeft aan hoeveel data er mag worden verzonden beginnend met de byte die bevestigd wordt in het segment. De waarde van het venster kan nul zijn en geeft aan dat de ontvanger nu geen data kan ontvangen. Wanneer een nieuw segment gezonden wordt met hetzelfde bevestigingsnummer een een windows dat niet nul is, geeft de TCP-module aan dat er terug data ontvangen kan worden.

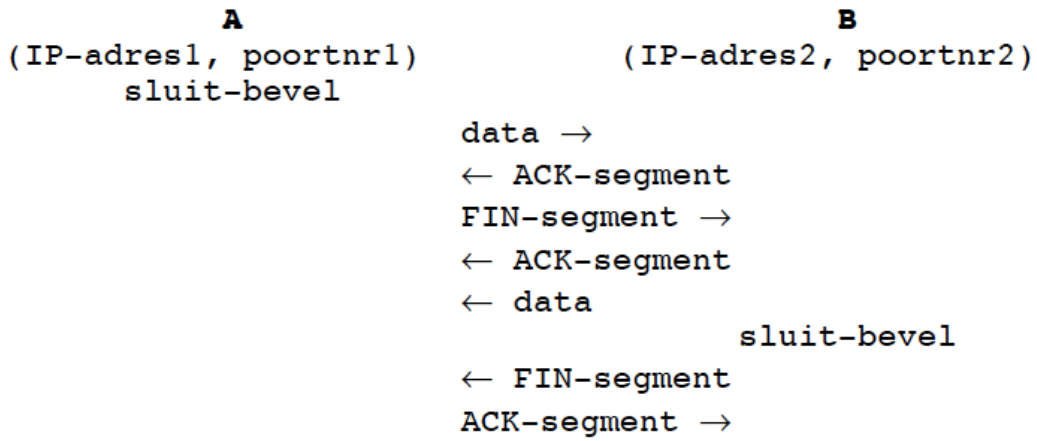
Een verbinding beëindigen Uitzonderlijk kan een verbinding afgebroken worden via een reset. Als één eindpunt een uitzonderlijke voorwaarde vaststelt, voert het een reset-bewerking uit. D.w.z.:
- de eigen toepassing (proces) wordt ervan verwittigd dat de verbinding afgebroken wordt
- er wordt een reset-segment (met de RST-bit =1) naar het andere eindpunt gestuurd
- de buffers worden opgeheven.

Een eindpunt dat een reset-segment ontvangt :
- verwittigt het proces;
- heft de buffers op.

De normale manier van beëindigen is natuurlijk anders. Een TCP-verbinding is een full duplex verbinding die kan worden aanzien als twee simplex verbindingen. Deze twee simplex verbindingen worden onder normale TCP omstandigheden apart opgeheven door de desbetreffende TCP-modules[4].

Het initiatief om te stoppen gaat uit van een proces. In één eindpunt (bvb. A) geeft het proces een sluit-bevel. Zie figuur 4.4.
Na dit sluit-bevel zal de TCP-module (A) :
- de overblijvende data in de buffer versturen;
- wachten tot B de ontvangst bevestigt;
- een FIN-segment sturen (een dergelijk segment heeft de FIN-bit = 1, en bevat een bytevolgnummer, bvb. x);

[4]Deze manier om een TCP-verbinding op te zeggen wordt symmetrisch genoemd.

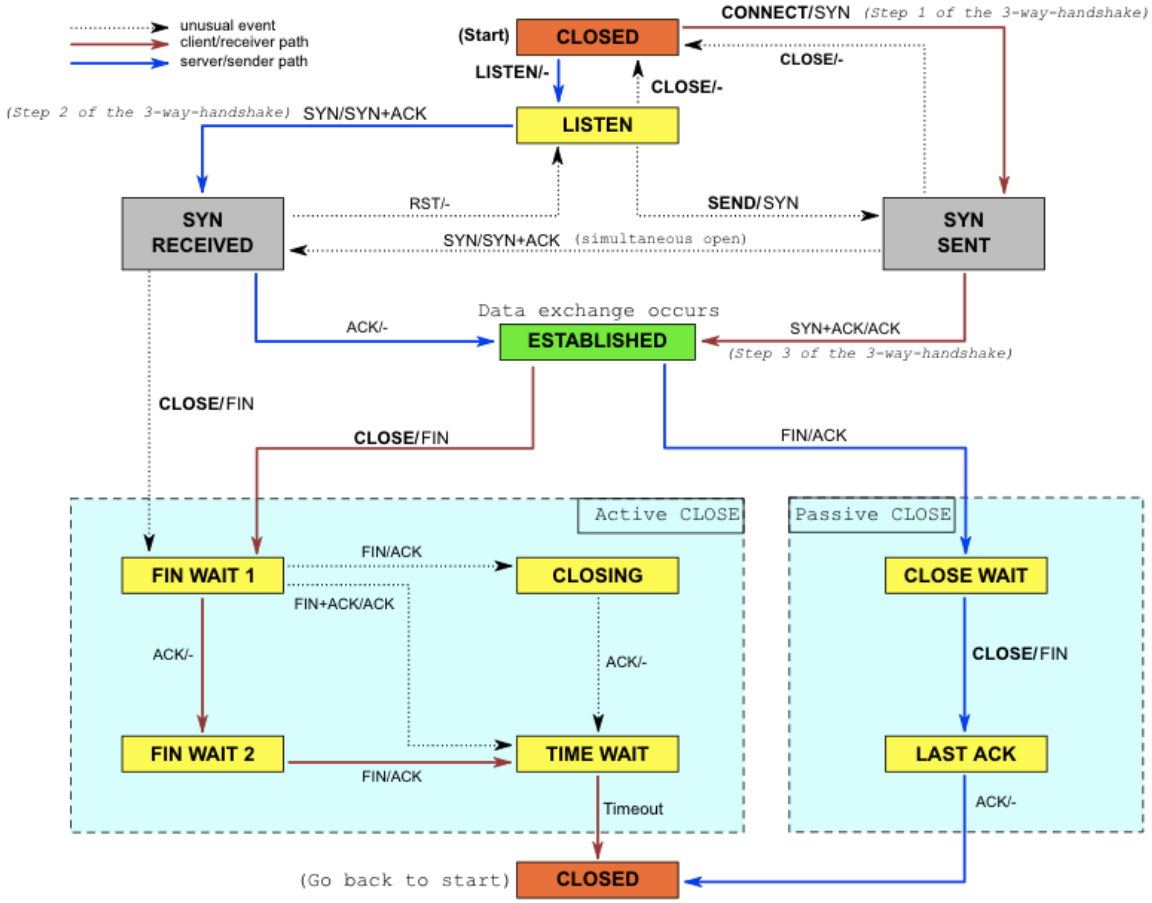


Figuur 4.4: Verbreking van een TCP-verbinding

Als de TCP-module (B) een FIN-segment ontvangt :
- wordt een ACK-segment teruggestuurd (ACK-bit =1 en ik ”verwacht”vanaf byte x+1);
- wordt het eigen proces er van verwittigd dat er geen data meer komen (soort eof);
- worden geen data (van A) meer aanvaard : één kant van de verbinding is gesloten; B kan blijven data sturen; A zal die op de gewone wijze bevestigen. Uiteindelijk zal het proces aan de kant van B ook een sluit-bevel geven.
De TCP-module verstuurt dan een FIN-segment (FIN-bit =1; bytevolgnummer =y; ACK-bit =1 : ik verwacht vanaf byte x+1; nog altijd x+1, want er kunnen geen data meer komen van A).
Als A dit segment ontvangt :
- zendt A een ACK-segment (ACK-bit =1 en ik verwacht vanaf byte y+1);
- heft A de verbinding op.
Als dit ACK-segment ontvangen wordt bij B, wordt de verbinding opgeheven. Normaal zijn er vier TCP-segmenten nodig om een verbinding te verbreken, twee (FIN en ACK) voor elke richting van de connectie. Het is mogelijk om de eerste ACK en de tweede FIN in eenzelfde segment te verzenden, zodoende zijn er maar drie segmenten nodig om een connectie te verbreken. Beide manieren van verbreking zijn toegestaan.

4.3.2 TCP state diagramma

De verschillende fases die nodig zijn om TCP-verbindingen op te zetten en te verbreken, kunnen worden voorgesteld door een eindige-toestandsdiagram met elf toestanden, zie figuur 4.5. In elke toestand zijn bepaalde events of acties toegestaan en kan er overgegaan worden naar een andere toestand.



Figuur 4.5: TCP eindige-toestandsdiagram

4.3.3 TCP tijdsbeheersing

Wanneer een TCP-module een segment zendt wordt er een counter of teller gestart bij de zendende module. Wanneer de ontvangende TCP-module het segment bevestigd heeft alvorens de teller is verlopen, wordt de teller gestopt. Wanneer het segment niet bevestigd is alvorens de teller is verlopen, zal het segment opnieuw verzonden worden en wordt de teller opnieuw ingesteld.
Wanneer de tijd alvorens de teller verstreken is, te kort blijkt te zijn, zullen er onnodige retransmissies plaatsvinden. Als de tijd te groot ingesteld wordt, zal het systeem traag worden omdat er onnodig lang gewacht wordt alvorens een retransmissie gebeurd.

Om bovenstaand probleem op te lossen zal de tijd van de teller dynamisch ingesteld worden door een algoritme dat voortdurend de snelheid van het netwerk evalueert. Het kiezen van een ideale instelling voor de teller is geen eenvoudige taak vooral wanneer de maximale belasting van het netwerk bereikt wordt. Poissonmodellen of wachtrijtheorie voorspelt dat de

Tabel 4.1: De toestanden in het TCP eindige-toestanden diagram om verbindingen op te zetten en te verbreken

Toestand	**Beschrijving**
CLOSED	Geen verbinding actief of hangende
LISTEN	De server wacht op een inkomende aanroep
SYN RCVD	Er is een aanvraag voor een verbinding aangekomen
SYN SENT	De applicatie is begonnen met het openen van een verbinding
ESTABLISHED	De normale toestand voor datatransfer
FIN WAIT 1	De applicatie heeft gezegd klaar te zijn
FIN WAIT 2	De andere kant heeft ingestemd met verbreking
TIMED WAIT	Wacht tot alle pakketten uitgestorven zijn
CLOSING	Beide kanten hebben tegelijkertijd gepoogd te sluiten
CLOSE WAIT	DE andere kant heeft het verbreken gestart
LAST ACK	Wacht tot alle pakketten zijn uitgestorven

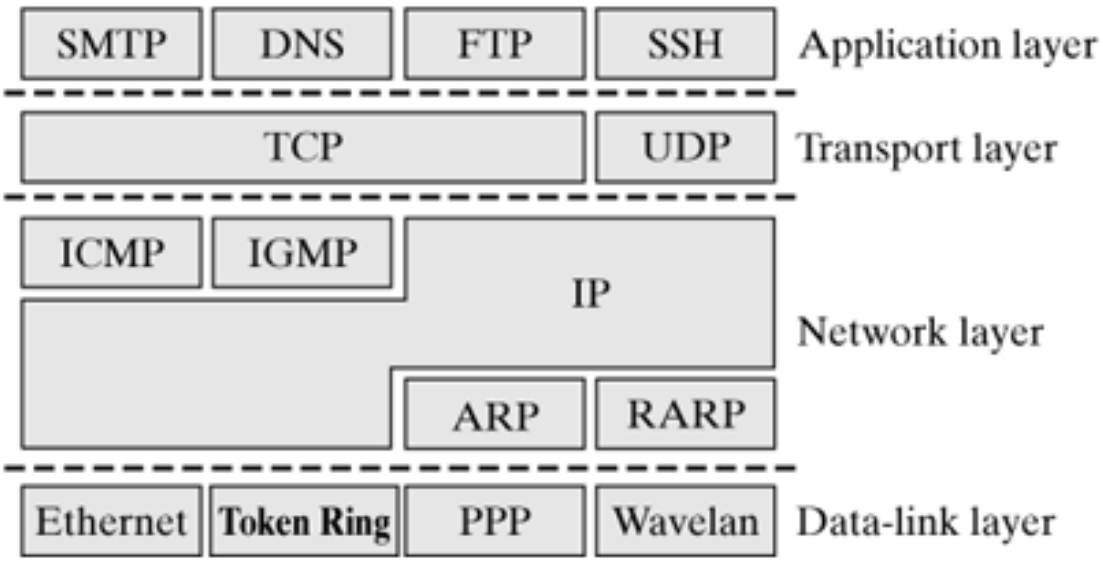
vertragingen groot en sterk variabel zijn wanneer de belasting van het netwerk de capaciteit evenaart.
Verdere bespreking van dit topic valt buiten deze inleidende cursus.

4.3.4 TCP congestie management

Wanneer een netwerk meer pakketten te verwerken krijgt dan dat het afhandelen ontstaat congestie. Internet is geen uitzondering! Weet dat het congestievenster tegelijkertijd wordt bijgehouden samen met het flow-control-venster. Het aantal bytes dat zal worden verzonden, wordt bepaald door het kleinste van de twee vensters. TCP stopt dus met zenden als ofwel het congestievenster ofwel het flow-control-venster vol zijn. Als de ontvangende TCP-module laat weten dat het 64KB kan bufferen en dus ontvangen, maar de zendende TCP-module weet dat bursts van 32KB het netwerk satureren, zal hij 32KB zenden.
Verdere bespreking van dit topic valt buiten deze inleidende cursus.

4.3.5 Protocolstapel

De afkorting TCP/IP wordt vaak gebruikt om het geheel (voor de verschillende lagen) van protocollen aan te geven. Men gebruikt de term : protocolsuite, of protocolstapel. Deze stapel kunnen we als volgt voorstellen:



Bovenstaand plaatje is bijlange niet volledig : er zijn veel meer protocollen voor communicatie tussen computers. Dus, alhoewel TCP en IP slechts 2 protocollen zijn, wordt met de term TCP/IP-suite vaak de ganse verzameling van protocollen bedoeld.

4.3.6 Protocol data unit

We hebben reeds gezien :
- Bij TCP heeft men het over segmenten;
- Bij IP heeft men het over pakketten;
- Bij Ethernet heeft men het over frames;
- Fysisch worden bits verstuurd.

De algemene naam voor de informatie die onder een bepaald protocol verstuurd wordt, heet protocol data unit (PDU). Afhankelijk van het protocol is een PDU dus: segment, pakket, frame of bit. Hoe heten de PDU's voor de toepassingslaag? Die heten simpelweg data. Dit kan een bestand, een e-mail, enz... zijn.

4.4 Adresvertaling: (S/D)NAT, PAT

Voor IP is er een essentiële vereiste dat alle computers in het netwerk een uniek adres hebben. Dit adres is het IPv4-adres van 32 bits. Als verschillende netwerken verbonden worden, dan moet deze voorwaarde nog altijd voldaan zijn. Twee computers zouden hetzelfde IP-adres kunnen hebben als :
- ze zich in verschillende netwerken bevinden;
- en als deze netwerken NIET met mekaar verbonden zijn.

Worden deze netwerken achteraf wel verbonden dan moet één van die computers een ander IP-adres krijgen. De eis om aan verschillende computers een verschillend IP-adres toe te kennen, leidt al snel tot de uitputting van het aantal beschikbare adressen. 4 miljard adressen is wereldwijd niet zo vreselijk veel. Op de invoering van IP-next generation (IPv6) kon niet gewacht worden. Een oplossing is te vinden in een andere richting:de buitenwereld heeft niets te maken met de interne communicatie in een bedrijf (of school). Als de IP-adressen die een bedrijf naar de buitenwereld stuurt, verschillen van de IP-adressen die in de buitenwereld gebruikt worden, is alles o.k.

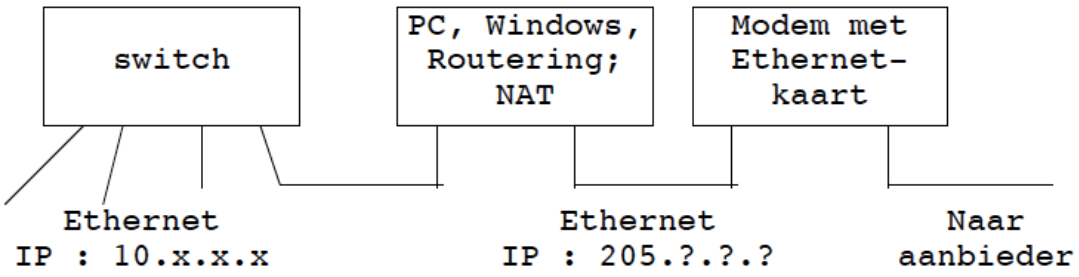
Vandaar volgende oplossing :
- een bedrijf krijgt een aantal IP-adressen toegewezen voor communicatie met de buitenwereld; bvb. de 30 adressen van 198.15.16.1 tot en met 198.15.16.30; deze adressen worden globale adressen genoemd;
- voor interne communicatie worden andere IP-adressen gebruikt; om dit mogelijk te maken zijn de IP-adressen van 10.0.0.0/8[5] gereserveerd voor intern gebruik; ze worden lokale of private adressen genoemd;
- van de IP-pakketten die een computer naar de buitenwereld stuurt, wordt het privaat IP-adres, bvb. 10.18.160.17 eerst vervangen door één van de globale, bvb. 198.15.16.12;
- van pakketten die van de buitenwereld als antwoord terugkomen en als bestemmingsadres 198.15.16.12 hebben, wordt het IP-adres vervangen door het oorspronkelijk adres 10.18.160.17.
Het vervangen van IP-adressen wordt NAT (network address translation) genoemd. Bemerk dat hierdoor het ganse IP-pakket moet bewerkt worden, omdat een nieuwe CRC moet berekend worden. Alleen voor de pakketten die voorbij de router moeten, moet NAT gebeuren.

Meer pc's dan IP-adressen NAT laat toe meerdere pc's een IP-adres te laten delen. Zo bvb. :
- We hebben één echt IP-adres, bvb. 205.205.205.205;
- We laten 4 pc's de adressen gebruiken 10.2.4.1 t/m 10.2.4.4.
- Stel dat elk van deze pc's tezelfdertijd surft naar 130.10.10.10 (ze vragen alle 4 andere pagina's). Hoe kunnen de IP-pakketten die terugkeren van de WWW-server naar de goede pc geleid worden?
Bij het opzetten van de TCP-verbinding naar poort 80 van 130.10.10.10, kiest elke cliënt een vrij poortnummer. Stel dat ze alle hetzelfde poortnummer, 1683, kiezen. De aanvragen voor een verbinding zijn dan:

Bron IP	Bron Poort	Bestemm. IP	Bestemm Poort
10.2.4.1	1683	130.10.10.10	80
10.2.4.2	1683	130.10.10.10	80
10.2.4.3	1683	130.10.10.10	80
10.2.4.4	1683	130.10.10.10	80

[5]Zie RFC 1918

Als de router alleen de IP-adressen 10.2.4.1 t/m 10.2.4.4 zou vervangen door 205.205.205.205, dan zou er maar 1 verbinding zijn met de WWW-server. Daarom wordt ook de bronpoort (die toch willekeurig is) vervangen, bvb. door 1684, 1685, 1686, 1687. Aldus worden 4 verbindingen naar de WWW-server opgezet. Pakketten die terugkeren bevatten ook het poort-nummer van de cliënt (1684, 1685, 1686, 1687). Op basis van dit nummer kunnen ze naar de juiste cliënt gerouteerd worden. Routering en NAT kunnen in een klein netwerk ook gedaan worden door een daartoe geconfigureerde pc, voorzien van 2 Ethernet-kaarten (waarom niet één Ethernetkaart en de modem ook aansluiten op de switch?) Een mogelijke opstelling om één IP-adres (snelle Internet-verbinding) met meerderen te delen :



Opmerking: Volgende groepen IP-adressen zijn gereserveerd als private adressen:
- 10.0.0.0/8 (dus van 10.0.0.1 t/m 10.255.255.254);
- 172.16.0.0/12 (dus van 172.16.0.1 t/m 172.31.255.254);
- 192.168.0.0/16 (dus van 192.168.0.1 t/m 192.168.255.254).

Een router zal nooit een IP-pakket met een bestemmings- of afzenderadres uit deze groepen op het Internet zetten.

Hoofdstuk 5

Applicatielaag en de (ondersteunende) applicaties

De lagen onder de applicatielaag zijn er om transportservices te leveren. Ze doen geen **echt** werk voor de eindgebruiker van concrete applicaties.

5.1 DNS

Een WWW-(of andere) cliënt die een bericht naar een computer wil sturen moet het IP-adres van deze computer opgeven. Bedrijven of instellingen publiceren dit IP-adres niet maar wel de naam van hun web-site (d.i. van hun WWW-server). Hoe kan een cliënt, gegeven de naam van een computer, het IP-adres te weten komen? Om het IP-adres te vernemen van een computer waarvan de naam gekend is, wordt gebruik gemaakt van een cliënt/server-toepassing. Er zijn servers die van een aantal computers de naam en het bijhorende IP-adres kennen. Er is een protocol uitgewerkt dat specifieert:

- hoe namen moeten toegekend worden aan machines
- welke berichten kunnen uitgewisseld worden tussen server en cliënt om antwoord te krijgen op een vragen als: ”Wat is het IP-adres van www.freebsd.org?”

Dit protocol heet: DNS (domain name system). De servers heten DNS-servers. De namen-specificatie is als volgt. De ICANN verdeelt de aangesloten netwerken in domeinen. Aan elk domein wordt een naam toegekend: com, edu, gov, mil, org,..., be, de, fi, fr, nl, ...
Deze domeinen worden ook topdomeinen genoemd. Elk netwerk behoort tot één van deze domeinen. Bij het aansluiten van een netwerk moet een keuze gemaakt worden. Sommige namen duiden op de aard van het bedrijf dat men aansluit : commercieel, onderwijs (E. education),... Andere zijn een code voor de nationaliteit : belgië, deutschland, ... Per topdomein is er weer een organisatie die zorgt voor een verdere indeling. Een domein dat

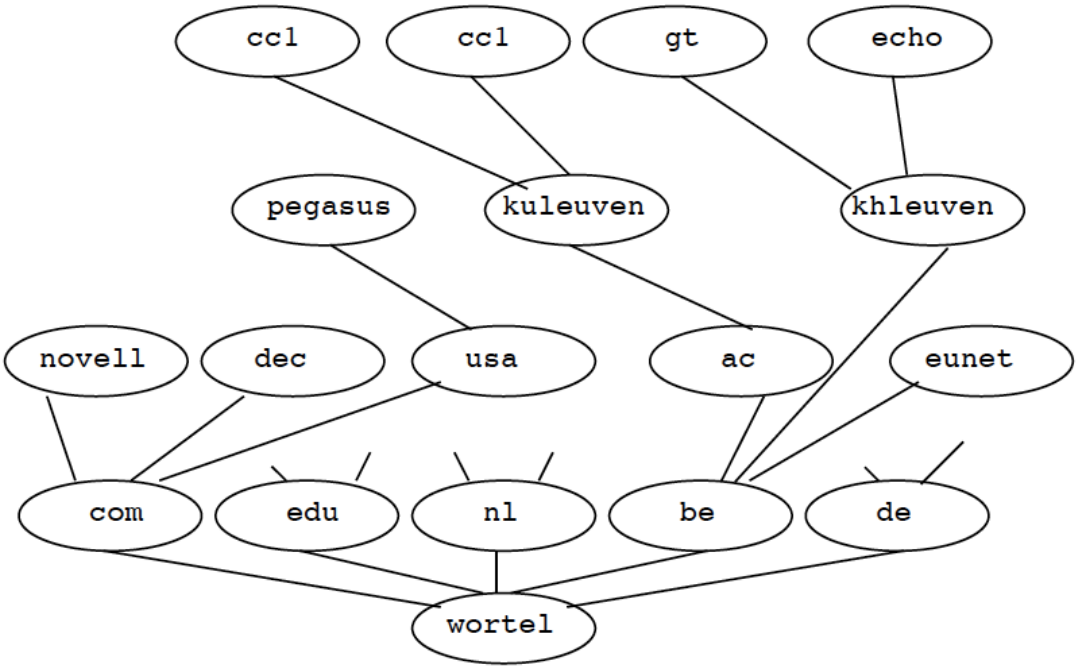
een deel is van een ander domein wordt subdomein genoemd. (subdomein, topdomein zijn geen officiële benamingen, alles heet domein). De regel voor naamgeving is:

subdomein.domein

Bijvoorbeeld: novell.com, dec.com, usa.com, csw.edu, netlib.org, ac.be, htwdresden.de, spt.fi, tech.fr, rug.nl,... .
De organisatie die verantwoordelijk is voor een domein (voor het domein ”beïs dit: DNS Belgium VZW; hierin staat de S voor service) :

- registreert namen
- creëert subdomeinen, bij de naamgeving wordt de bovenstaande regel gevolgd
- delegeert het beheer over deze subdomeinen

Aldus ontstaat de volgende boomstructuur:



Bij deze indeling komen we uiteindelijk uit op een computer, bvb. aha.ment.ucll.be Een computer heet in deze context host; zijn naam is een hostnaam. Na het registreren van een domeinnaam voor een netwerk is de 2° taak van DNS **namen vervangen door IP-adressen**.
De cliënt stuurt een vraag (E. query) naar een naam-server, die dan een antwoord terugstuurt. Een cliënt moet het IP-adres van minstens één naam-server kennen. Een naam-server is een serverproces, maar kan ook betekenen : de computer die dit proces uitvoert. Het poortnummer van het

DNS-serverproces is 53. Eén server kan meerdere subdomeinen beheren. De domeinen die een server beheert, vormen zijn autoriteitsgebied (E. : zone of authority).
De taak van de naam-server is : de vragen van de cliënten beantwoorden. De naam-server die een cliënt als eerste kan raadplegen, wordt lokale naamserver genoemd. Om het aantal keer dat een server gecontacteerd moet worden te beperken, houdt de cliënt en cache bij. De items in deze cache zijn van de vorm:

Domein- (of host)naam, IP-adres

Als een machine een IP-adres nodig heeft, wordt een cliënt-proces gestart. Dit proces wordt ook name resolver genoemd. Het zoekt eerst de domeinnaam in de cache. Staat de domeinnaam er NIET in, dan wordt een zgn. naam-server geraadpleegd.

5.1.1 Hoe wordt de DNS-informatie bewaard?

Oorspronkelijk werd de lijst met namen en bijhorende IP-adressen bewaard in een bestand HOST.TXT. Het beheer er van gebeurde door het Network Information Center van het SRI (Research Instituut van de Universiteit van Stanford). Wie aan een machine een naam wilde toekennen, moest dit melden aan Stanford. Als deze naam nog niet bestond, werd hij aan de lijst toegevoegd. Gebruikers moesten regelmatig de nieuwste versie van HOST.TXT opvragen.

Met de omvang van het huidige Internet, kan deze eenvoudige manier van werken natuurlijk niet meer. Nu wordt de DNS-informatie bewaard in een gedistribueerde databank : wereldwijd zijn er DNS-servers die elk een stukje van de DNS-informatie hebben. Deze servers kunnen onderling communiceren en informatie uitwisselen. Elke computer (die via TCP/IP communiceert) kent het IP-adres van een DNS-server en kan bij deze server terecht met vragen betreffende de DNS-informatie.

Een ander voorbeeld van een gedistribueerde databank is de informatie over namen (incl. adres) en telefoonnummers. De informatie is opgeslagen in diverse telefoonboeken.

5.1.2 Hiërarchische naamgeving

Niet alleen de opslag van de informatie is gedistribueerd, ook het toekennen en beheren van de namen gebeurt niet centraal. Op het hoogste niveau is er het ICANN dat de namen beheert van de topdomeinen:

- com.
- gov.

- ...
- be.
- ...

Wie een topdomein zou willen toevoegen, zou dit moeten vragen aan het ICANN. **Voor elk domein is er een instantie die bevoegd is voor de namen binnen dat domein**. Voor het domein ”be.ïs dit DNS Belgium VZW. Voor het domein ücll.be.ïs de UC Leuven-Limburg de bevoegde instantie. Deze bevoegdheid houdt in:

- aan een computer een naam toekennen, bvb. www.ucll.be
- naar believen subdomeinen creëren, bvb. ti.ucll.be. (ücll.be.ïs dan het ouderdomein of parentdomein)
- al of niet het beheer van een subdomein delegeren aan een ondergeschikt bestuur (bvb. een departement)

De instantie die bevoegd is voor een domein moet een DNS-server installeren. Het IP-adres van deze DNS-server moet bekend zijn bij de bevoegde instantie van het ouderdomein. Dus UC Leuven-Limburg moet een DNS-server installeren voor haar domein ücll.be.”; DNS Belgium VZW moet het IP-adres ervan kennen. (Er wordt ook vereist dat er een tweede DNS-server is, die een kopie van de DNS-informatie heeft).

5.1.3 Zone

De termen zone en domein worden vaak door mekaar gebruikt maar er is een subtiel verschil tussen beide. Als er in een domein geen delegatie gebeurd is, dan is dit domein een zone. Als er hier subdomeinen zijn, behoren deze ook tot de zone. Alle DNS-informatie, ook voor de subdomeinen wordt bewaard door dezelfde DNS-server; zo bvb. de informatie voor ücll.be.”, ”ti.ucll.be.”, ”chem.ucll.be.”, ... wordt bewaard door dezelfde DNS-server. Als er subdomeinen zijn en het beheer ervan, bvb. ”ti.ucll.be.”zou gedelegeerd worden dan behoort dit subdomein niet langer tot de zone ücll.be.”; het wordt een aparte zone en het departement TI moet zelf een DNS-server installeren voor deze zone. Enkele feiten:

- Een DNS-domein is een groep van namen die op hetzelfde suffix eindigen (suffix = achtervoegsel) met een bestuur dat namen toekent aan computers. Bvb. het domein ”be.”
- Domeinen zijn hiërarchisch opgebouwd, zoals weergegeven in de figuur van §1.3. Deze hiërarchische structuur vormt een boom en wordt naamruimte (E. name space) genoemd
- Een zone is een domein maar zonder de subdomeinen waarvan het beheer gedelegeerd is aan een andere organisatie.

FQDN (fully qualified domain name) Een domeinnaam is fully qualified als hij eindigt bij de wortel van de boom (die de naamruimte voorstelt). De wortel wordt genoteerd als een punt. Voorbeelden van FQDN's zijn:

- be.
- khleuven.be.
- kuleuven.ac.be.
- ucll.be.

Men kan een FQDN vergelijken met een volledige padnaam bij een bestandssysteem (de volgorde is wel andersom). De naam rega.khleuven is niet fully qualified. Men kan een dergelijke naam alleen maar gebruiken als duidelijk is wat het ouderdomein is (bvb. be.).

5.1.4 Hoe gebeurt de naam-resolutie?

Als iemand surft naar www.val.be dan moet deze naam vervangen worden door het IP-adres. Dit vervangen heet resolutie (E. resolution). Het cliëntprogramma dat de naam vervangt, heet resolver. Een resolver maakt deel uit van de browser (hij is er wel maar je ziet hem niet). Een vereiste is dat de computer het IP-adres van een DNS-server kent. Men noemt dit de lokale DNS-server. Het adres van de LDS (lokale DNS-server) maakt deel uit van de TCP/IP-configuratie, samen met IP-adres, subnetmasker en IP-adres van de router. De resolver zal eerst de naam aanvullen tot hij "fully qualifiedïs, dus "www.val.be"wordt "www.val.be.". Daarna vraagt de resolver aan de LDS "wat hoort bij de FQDN www.val.be.?"
Als de LDS het antwoord kent, stuurt hij het IP-adres terug. Wat als hij het IP-adres niet kent? De standaard-oplossing is het gebruik van zgn. root-servers. Als een DNS-server geïnstalleerd wordt (praktisch gebeurt dit door het besturingssysteem de DNS-serverprogrammatuur le laten kopiëren naar de harde schijf) dan wordt ook een lijst met root-servers gekopieerd. Deze lijst, de hints-file, bevat o.m.:
a.root-servers.net 198.41.0.4
...
d.root-servers.net 128.8.10.90
...

Alle root-servers kennen voor alle top-domeinen (het IP-adres van) een DNS server.

Voorbeeld De gebruiker die aan pc6.ti.ucll.be zit, surft naar www.wellington.govt.nz. Dan krijgen we volgende stappen:

1. De resolver vraagt aan zijn LDS ”wat hoort bij de FQDN www.wellington.govt.nz.?”D
resolver kan dit omdat hij het IP-adres van de LDS kent. Dit is de DNS-server van de UCLL.
2. De DNS-server van de UCLL kent ”www.wellington.govt.nz.”niet en vraagt één van de root-servers, bvb. d.root-server.net (met ip-adres 128.8.10.90) ”wat hoort bij de FQDN www.wellington.govt.nz.?”
3. d.root-server.net weet het wellicht ook niet, maar als root-server kent bij wel het IP-adres van de (een) DNS-server voor het topdomein ”nz.”. Dit IP-adres wordt naar de DNS-server van de UCLL gestuurd. Het is bvb. 19.146.0.4.
4. De DNS-server van de UCLL vraagt aan 19.146.0.4 (d.i. de DNS-server van ”nz.”) ”wat hoort bij de FQDN www.wellington.govt.nz.?”
5. De DNS-server van ”nz.äntwoordt met het IP-adres van de DNS-server voor ”govt.nz.”. Dit is bvb. 217.22.59.4.
6. De DNS-server van de UCLL vraagt aan 217.22.59.4 (d.i. de DNS-server van ”govt.nz.”) ”wat hoort bij de FQDN www.wellington.govt.nz.?”
7. De DNS-server van ”govt.nz.äntwoordt met het IP-adres van de DNS-server voor ”wellington.govt.nz.”. Dit is bvb. 192.190.108.253.
8. De DNS-server van de UCLL vraagt aan 192.190.108.253 (d.i. de DNS-server van ”wellington.govt.nz.”) ”wat hoort bij de FQDN www.wellington.govt.nz.?”
9. De DNS-server van ”wellington.govt.nz.äntwoordt met het IP-adres van www.wellington.govt.nz.”. Dit is bvb. 192.190.108.36.
10. De DNS-server van de KHLeuven stuurt dit IP-adres (192.190.108.36) naar de resolver.

Lijm Even benadrukken dat -voor stap 5 hierboven- de DNS-server van ”nz.”het IP-adres van de DNS-server voor ”govt.nz.”moet kennen. Deze informatie is essentieel voor de werking van het DNS, het is de lijm waarmee het systeem aan mekaar hangt. Hetzelfde geldt voor stap 7.

Recursief, Iteratief De taak van de naam-server is: de vragen van de cliënten beantwoorden. Dit kan op iteratieve of recursieve wijze. Iteratief wil zeggen: als de server het antwoord zelf niet weet, dan geeft hij aan de cliënt het IP-adres van een andere server. Als de cliënt dan deze server raadpleegt, is hij een stap dichter bij het antwoord. Recursief betekent: als de server het antwoord niet weet, dan raadpleegt hij zelf (een) andere server(s) tot hij het antwoord weet.
Bij stappen 1 en 2 hierboven: is dit een recursieve of een iteratieve werk-wijze?

Forwarders Een DNS-server beschikt over een lijst met root-servers die het IP-adres van DNS-servers voor elk topdomein kennen (er komt wel eens een practicum waarbij je een DNS-server moet installeren; kijk na de installatie eens naar root hints). Toch wordt bij voorkeur geen gebruik gemaakt van de root-servers. De beheerder die een DNS-server configureert, kan één of meer forwarders opgeven. Een forwarder is een andere DNS-server die door een DNS-server moet geraadpleegd worden als hij zelf het antwoord niet weet.
Voorbeeld: op de DNS-server van de UCLL is de DNS-server van ”be.öpgegeven als forwarder. Dan wordt stap 2 van het vorig voorbeeld : de DNS-server van de UCLL kent ”www.val.be.”niet en vraagt aan de DNS-server van ”be.”(dus aan een forwarder) : ”wat hoort bij de FQDN www.val.be.?”. Het gaat dan verder zoals in het voorbeeld vanaf stap 5.

Cache Als een DNS-server een IP-adres verkregen heeft, zal hij dit gedurende een zekere periode in een cache bewaren. Hoelang duurt deze periode?

UDP De berichten die de resolver en de DNS-servers versturen gaan meestal via UDP. Het is immers niet nodig om een TCP-verbinding op te bouwen alleen maar om te vragen ”wat hoort bij de FQDN xx.yy.zz.?”.

5.1.5 Omgekeerde naamresolutie

Het zal iedereen wel duidelijk zijn waarom namen door IP-adressen moeten vervangen worden. Maar ook de omgekeerde bewerking moet kunnen : ook een vraag als ”wat is de naam van 193.190.138.4”moet kunnen beantwoord worden. Hier zijn een aantal redenen voor:

- als een webserver verneemt dat iemand met IP-adres 192.18.6.3 komt surfen, zal de webmaster misschien graag willen weten welke computer dit is (bvb. welke de naam is van het bedrijf)
- tracert is een programmaatje om de route naar een bestemming te traceren. In eerste instantie verneemt het programma alleen de IP-adressen van de routers op weg naar de bestemming. Via een omgekeerde naamresolutie kan het programma ook de namen van de routers tonen.

Om omgekeerde naamresolutie mogelijk te maken, werd een speciaal domein gecreëerd:

in-addr.arpa.

(ärpa.ïs één van de topdomeinen). Dit speciaal domein bevat 256 subdomeinen:

0.in-addr.arpa.
1.in-addr.arpa.
...
255.in-addr.arpa.
Elk van deze subdomeinen bevat op zijn beurt weer 256 subdomeinen; voor bvb. ”123.in-addr.arpa.”zijn dit:
0.123.in-addr.arpa.
1.123.in-addr.arpa.
...
255.123.in-addr.arpa.

Ook deze subdomeinen bevatten 256 subdomeinen; voor bvb. ”148.66.inaddr.arpa.”zijn dit:

0.148.66.in-addr.arpa.
1.148.66.in-addr.arpa.
...
255.148.66.in-addr.arpa.

De vraag ”welke naam komt overeen met IP-adres 193.190.138.4”kan opgelost worden door aan de lokale DNS-server te vragen : ”wat hoort bij de FQDN 4.138.190.193.in-addr.arpa.?”(bemerk de volgorde). Het is interessant om de gevolgen van deze vraag te bekijken.
Via een root-server of een forwarder komt de vraag terecht bij de DNS-server voor het domein ïn-addr.arpa.”. De vraag die deze moet beantwoorden is : ”wie weet aan wie een IP-adres dat begint met 193, toegewezen is. Nu, het ICANN dat de IP-adressen beheert, heeft blokken van IP-adressen uitgedeeld aan (wereldwijd gezien) regionale organisaties. Zo’n organisatie is bvb. het RIPE (Réseaux IP Européens, jaja, oui oui, en français; maar de volledige naam is RIPE NCC, van Network Coordination Centre); ARIN is en andere regionale organisatie. Regionaal moet correct begrepen worden, RIPE heeft betrekking op : Europa, het Midden-Oosten, Centraal Azië, en de Afrikaanse landen van het noordelijk halfrond. Een overzicht van de toewijzing van IP-adressen vinden we op : http://www.iana.org/assignments/ipv4-address-space; zo bvb.:
193/8 May 93 RIPE NCC
194/8 May 93 RIPE NCC
195/8 May 93 RIPE NCC
...
199/8 May 93 ARIN

We zien dus dat IP-adressen die beginnen (o.m.) met 193 (met de notatie 193.0.0.0/8) toegewezen zijn aan het RIPE NCC, of wat betreft DNS : het ICANN heeft het beheer van subdomeinen als ”193.in-addr.arpa.”gedelegeerd aan het RIPE NCC. De vraag ”wat hoort bij de FQDN 4.138.190.193.inaddr.arpa.?”zal dus terecht komen bij de DNS-server voor ”193.in-addr.arpa.”die beheerd wordt door het RIPE NCC.

Ook RIPE heeft het beheer van IP-adressen verder gedelegeerd. Een ledenlijst vinden we op: http://www.ripe.net/membership/indices/BE.html. Via http://www.ripe.net/ en whois (bvb. 193.190.0.0) vinden we o.m.:
- 193.190.0.0/16 en 193.191.0.0/16: BELNET;
- 134.58.0.0/16 : Katholieke Universiteit Leuven;
- 193.190.138.0/24 : Katholieke Hogeschool Leuven V.Z.W. (dit is een verdere delegatie door BELNET).

Wat betreft DNS : het beheer van subdomeinen ”190.193.in-addr.arpa.ën ”191.193.in-addr.arpa.ïs gedelegeerd aan BELNET. De vraag ”wat hoort bij de FQDN 4.138.190.193.in-addr.arpa.?”zal dus terecht komen bij de DNS-server voor ”190.193.in-addr.arpa.”die beheerd wordt door BELNET. BELNET heeft het beheer van 193.190.138.0/24 gedelegeerd aan University Collega Leuven V.Z.W. (zo heten wij officieel). De vraag ”wat hoort bij de FQDN 4.138.190.193.in-addr.arpa.?”zal dus terecht komen bij de DNS-server voor ”138.190.193.in-addr.arpa.”die beheerd wordt door de UCLL. Als het IP-adres 193.190.138.4 niet zou toegekend zijn (en hierin is de UCLL natuurlijk volledig vrij) dan verneemt de vrager dat. Als het IP-adres wel toegekend is verneemt de vrager bvb.:

IP Address 193.190.138.4 resolves to:
moorse.ti.ucll.be

Een organisatie die aansluit op het Internet wordt verantwoordelijk voor 2 domeinen. In het geval van de UCLL zijn dit de domeinen : ucll.be, 138.190.193.in-addr.arpa.

5.1.6 Primaire, secundaire DNS-server

Om de goede werking van het DNS te verzekeren moeten er voor elk domein minstens 2 servers zijn. Aldus wordt het werk verdeeld. En, mocht er één server tijdelijk onbeschikbaar zijn dan blijft een andere zorgen voor naamresolutie. Het verschil in functie tussen beide is:
- wijzigingen aan de zonegegevens moeten gebeuren op de primaire server
- secundaire servers hebben een kopie van de zonedata. Deze kopie wordt regelmatig ververst. Zo’n verversing heet **zonetransfer**.
Een zonetransfer gebeurt over een TCP-verbinding.

5.1.7 De records van de DNS-databank

De informatie in de DNS-databank worden zonedata genoemd. Zonedata zijn opgeslagen als records, de zgn. resource records. Waarom zonedata en niet domeindata? Als het beheer voor een subdomein gedelegeerd is dan behoort dit subdomein niet tot de zone. Juist doordat het beheer gedelegeerd is, staan de gegevens ergens anders. **Men bewaart de gegevens dus niet per domein maar per zone.** Resource records bestaan uit een aantal velden: naam, TTL, klasse, type, data; bvb.:

www.khleuven.be 86400 IN A 193.190.138.71

- Het 1° veld is de naam van een domein of van computer
- Het 2° veld (dat mag weggelaten worden) bevat de duur van een periode, de zgn. time to live, uitgedrukt in seconden. Nadat deze periode verstreken is, moeten andere DNS-servers die een kopie van het record zouden opgeslagen hebben in hun cache, dit record verwijderen. Andere servers wil zeggen : andere dan de server die autoriteit (zie verder) heeft; 86400 seconden = 24 uren
- In het 3° veld staat bijna altijd IN (=Internet)
- Het 4° veld geeft het type weer. Dit type kan o.m. zijn : SOA, NS, A, MX, SRV, PTR, ... (zie verder)
- Afhankelijk van het type volgen een aantal parameters die de data vormen

A-record (A=address) Bvb. : www.khleuven.be. 86400 IN A 193.190.138.71
Een A-record geeft het IP-adres dat met een naam overeenkomt.

PTR-record (PTR=pointer) Bvb. : 4.138.190.193.in-addr.arpa. IN PTR moorse.ti.ucll.be. Een PTR-record geeft de naam voor een IP-adres. Het is het omgekeerde van een A-record. Niet onbelangrijk : het PTR-record bevindt zich een andere databank dan het A-record. Beide hebben immers betrekking op andere domeinen. Wat wel kan, is dat beide databanken op dezelfde server bewaard worden.

NS-record (NS=name server) Bvb.:
khleuven.be. IN NS ns.khleuven.be.
...
ns.khleuven.be. IN A 193.190.138.2

Een NS-record geeft de naam van de DNS-server voor een domein. Veel is men niet met deze naam; het IP-adres van de DNS-server is vereist. Dit IP-adres is te vinden in een A-record.
In principe bevatten de zonedata alleen informatie die betrekking heeft op de zone. Dat is logisch : de organisatie heeft het beheer gekregen voor de zone. Toch is er een uitzondering. Stel dat de KHLeuven het beleid voor een zone ”rega.khleuven.be.”delegeert aan het departement Rega. Alles wat betrekking heeft op deze zone wordt dan bewaard op de DNS-server van Rega. Binnen deze zone kan de beheerder naar believen namen, xxx. rega.khleuven.be toekennen. Maar het IP-adres van de DNS-server van de zone ”rega.khleuven.be.”moet als NS-record voorkomen in de zonedata van khleuven.be. Anders kan de naamresolutie niet werken. Immers,

als ergens een resolver het IP-adres van ”www.rega.khleuven.be.”nodig heeft, komt hij vroeg of laat uit bij de DNS-server van het domein ”khleuven.be.”. Deze moet weten wat het IP-adres van is van de DNS-server voor ”rega.khleuven.be.”, omdat daar het IP-adres van www.rega.khleuven.be staat.

SOA-record (SOA=start of authority) In elke zone moet er een SOA-record zijn. Het geeft aan welke server autoriteit heeft over de zone. (Ook

```
khleuven.be.    IN    SOA    ns.khleuven.be    dnsbaas.khleuven.be. (
                2109       ; serienr
                14400      ; refresh : 4 uren
                18000      ; retry :   5 uren
                8640000    ; expire :  100 dagen
                86400 )    ; default TTL : 1 dag
```

een secundaire server heeft autoriteit vermits deze een kopie heeft van de gegevens van de primaire server.) Van een server die autoriteit heeft wordt gezegd dat hij autoritatief is (gezag heeft in verstaanbaar Nederlands). Het SOA-record bevat gegevens over de zone

- de naam van de zone
- de naam van de primaire DNS-server
- het e-mail-adres van degene die verantwoordelijk is voor de zone (i.p.v. ”@”staat er ”.”)
- het serienummer (2109 in het voorbeeld). Telkens als er gegevens gewijzigd worden, wordt dit nummer met 1 opgehoogd. Als een secundaire DNS-server zijn data wil verversen, gaat hij eerst na of er wel wijzigingen zijn. Heeft de secundaire server hetzelfde serienummer dan weet hij dat er geen verversing nodig is
- het verversingsinterval (14400 sec in het voorbeeld). Hiermee legt de primaire server vast wanneer een secundaire zijn data moet verversen
- als een secundaire server wil verversen en geen contact kan krijgen met de primaire, moet hij niet blijven proberen maar na een zeker tijdsinterval opnieuw verbinding maken; in het voorbeeld is dit na 18000 seconden
- als een secundaire server er steeds niet in slaagt om zijn gegevens te verversen, mag hij na 864000 sec (de expire-waarde) geen gegevens meer verspreiden
- voor resource records waarbij geen TTL opgegeven wordt, wordt de default TTL gebruikt