Welkom bij de cursus

TCP/IP

Uw docent: Erik van Zuuren

Een cursus van IIR Technology
een divisie van IIR Seminars Europe BV

IIR TECHNOLOGY

Een divisie van IIR Seminars Europe B.V.

WELKOM !

IIR Technology heet u van harte welkom op deze cursus. Wij hopen dat aan uw verwachtingen zal worden voldaan.

DE IIR TECHNOLOGY SPELREGELS:

* gelieve niet te roken in de zaal,
* indien u gebruik wenst te maken van een telefoon meldt u zich dan even bij de receptie,
* tijdens de pauzes kan er geen alcohol geschonken worden,
* helpt u mee de afvalberg terug te dringen ? Lever dan uw badge na afloop van de cursus in bij de receptie,
* notities kunt u maken op de blanco pagina's die u achterin het cursusboek aantreft.
* de directie van IIR Technology stelt zich niet aansprakelijk voor vermissingen van eigendommen. Laat dus geen waardevolle spullen in de lokalen achter gedurende de pauzes.

EVALUATIEFORMULIEREN

Aan het eind van elke cursusdag stellen wij u in de gelegenheid om het dagevaluatieformulier in te vullen. Met deze evaluatie kan de docent zich voorbereiden op de volgende dag.

Aan het eind van de cursus wordt u verzocht het eind-evaluatieformulier in te vullen. Dit formulier is voor iedereen van groot belang. Uit dit formulier kunnen wij opmaken hoe u de cursus, de documentatie en de docent beoordeeld. Hiermee stelt u ons in de gelegenheid om onze Customer Service-activiteiten te verbeteren en toekomstige cursussen aan uw wensen aan te passen.

CURSUSINFORMATIE

Al onze cursusbrochures treft u aan in de brochurekast bij de receptie of op de aparte tafel in het lokaal. Neemt u gerust een kijkje in ons cursusaanbod. Indien u inhoudelijke vragen heeft kunt u zich wenden tot de heer Sander van der Meer van de afdeling Customer Service. Voor speciale maatwerk cursussen kunt u terecht bij mevrouw Joke Pikaar onze Incompany Training Manager.

Voor vragen en opmerkingen m.b.t. organisatorische zaken van IIR Technology kunt u altijd terecht bij onze receptioniste. Wij wensen u een prettig cursus toe.

IIR Technology

Hands-On TCP/IP

Ontwikkeld door

ARANEA CONSULT

NETWERKADVIES & ONDERSTEUNING

door

Roeland Ravesteijn
John Lasschuit

INHOUDSOPGAVE

INHOUDSOPGAVE
vervolg

IIR TECHNOLOGY

1. Inleiding

referentiemodel

1.1 OSI / RM
1.2 TCP/IP
1.3 Drie technieken in lagenmodellen
1.4 Samenvatting

IIR TECHNOLOGY

1.1 OSI / RM

- **Communicatie is zinvol uitwisselen van gegevens die tot *informatie* geinterpreteerd kan worden.**

- **Communicatie gaat uit van een *aantal afspraken* die beide partijen zijn overeengekomen: de *protocollen***

Er wordt gesproken van zinvolle communicatie als de gegevens die uitgewisseld worden door beide partijen op gelijke wijze geïnterpreteerd worden.

Daarom is het alleen mogelijk zinvol te communiceren indien het uitwisselen van de gegevens volgens vastomlijnde afspraken gebeurt. Dit worden protocollen genoemd.

Er kunnen bij communicatie meerdere protocollen gelijktijdig worden gebruikt. In conversaties bijvoorbeeld, wordt een protocol gehanteerd voor de wijze waarop de gegevens in signalen omgezet worden (spraaksignalen en de klanken die daarbij horen), wordt een protocol gehanteerd voor de taal waarin de gegevens uitgewisseld worden (bijvoorbeeld nederlands), is er een protocol die de grammatica en zinsopbouw regelt en is er een protocol die de stijl regelt.

Zonder deze protocollen zou het zelfs lastig zijn om met iemand een gesprek te voeren.

IIR TECHNOLOGY

1.1 OSI / RM

- **Protocollen zijn afspraken, op velerlei niveaus**
- **Onhandig om één groot dik protocol te ontwerpen om op twee computers te implementeren**
- **Handiger om verschillende protocollen voor verschillende niveaus te implementeren**

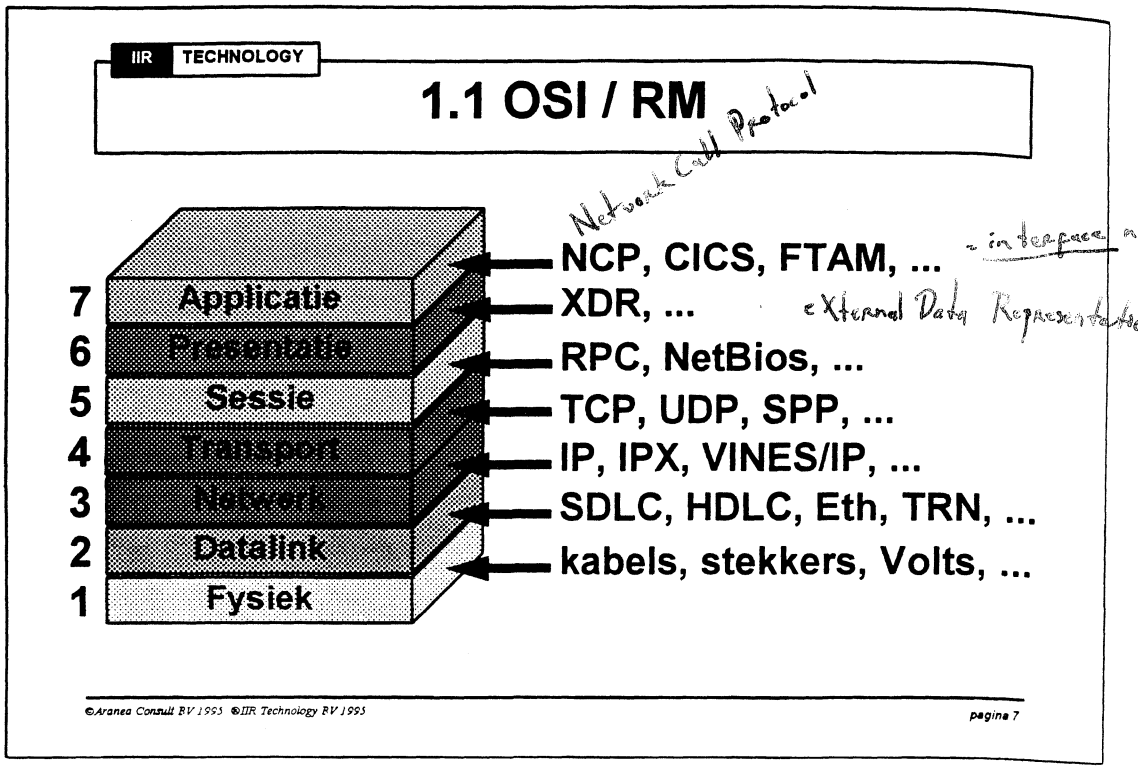
- **Eén model dat als referentie zou moeten dienen voor 'de rest': het OSI model van ISO**

Ook bij computers die onderling communiceren moeten er protocollen afgesproken worden om de communicatie zinvol te laten wezen.

Zo'n verzameling van protocollen zou in zijn geheel als eenheid geïmplementeerd kunnen worden, maar dat maakt het doorvoeren van wijzigingen in één van de protocollen een lastige zaak.

Daarom is het beter om ieder protocol apart te implementeren en af te spreken hoe de verschillende protocollen de gegevens aan elkaar door te geven. De wijze waarop protocollen binnen een verzameling protocollen de gegevens aan elkaar doorgeven, wordt een interface genoemd.

In 1972 is het ISO begonnen met de ontwikkeling van een referentiemodel voor protocollen die gebruikt moeten worden in de communicatie tussen twee computers. Dit referentiemodel is beter bekend als het OSI model. Het doel ervan was dat, als iedere computerfabrikant zijn protocollen analoog zou laten werken aan het referentiemodel, alles met alles zou kunnen communiceren.



Hier is het OSI Referentie Model in beeld gebracht, waarbij voor elke laag een paar voorbeelden van echte protocollen worden genoemd. Want het OSI-project heeft zijn doel niet bereikt: er zijn in de wereld vele tientallen protocollen in gebruik, het computer-Esperanto is nog steeds niet doorgebroken! Hier geven we een korte beschrijving van elke OSI-laag:

Fysieke Laag. Regelt kabels, impedanties, elektrische spanningen of optische niveaus, afmetingen van stekkers, signaleringssnelheid (bitrate), codering, modulatie en bit-synchronisatie. Voor telecom bekend terrein!

Data Link Laag. Byte- en frame synchronisatie, detectie van transmissiefouten (door gebruik van een *checksum*). Eigenlijk heeft deze laag twee functies: toegang regelen tot het gebruikte medium en de besturing van het gegevenstransport tussen twee aangrenzende apparaten.

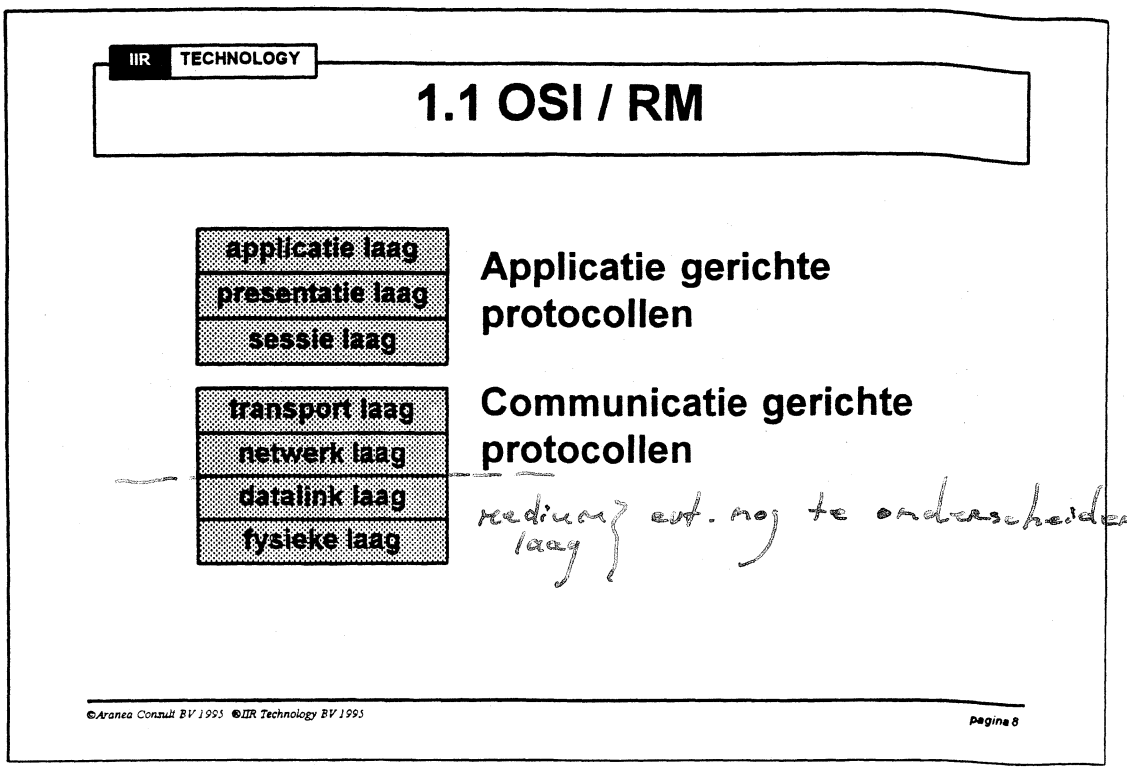
Netwerklaag. Routering door een samengesteld netwerk: van A naar F via B, C en D. Opeenvolgende frames kunnen langs verschillende wegen reizen, waardoor de volgorde kan veranderen!

Transportlaag. Zorgt voor het transport van volledige berichten, detecteert en herstelt fouten van de onderliggende lagen (ontbrekende of dubbele frames, frames die in een andere volgorde binnen komen dan ze werden uitgezonden).

Sessie Laag. Regelt het opzetten, onderhouden en weer afbreken van een logische verbinding.

Presentatie Laag. Zorgt voor 'vertaling', bv. conversie van ASCII naar EBCDIC of het aanpassen van de byte-volgorde tussen verschillende computerarchitecturen.

Applicatie Laag. *Niet* de toepassing zelf, maar het interface tussen de communicatiesoftware en de applicatie. Bevat functies voor het opzetten en afbreken van communicatie, opvragen van statusinformatie, bibliotheek-functies voor verschillende programmeertalen (Fortran, Cobol, Basic, ...).

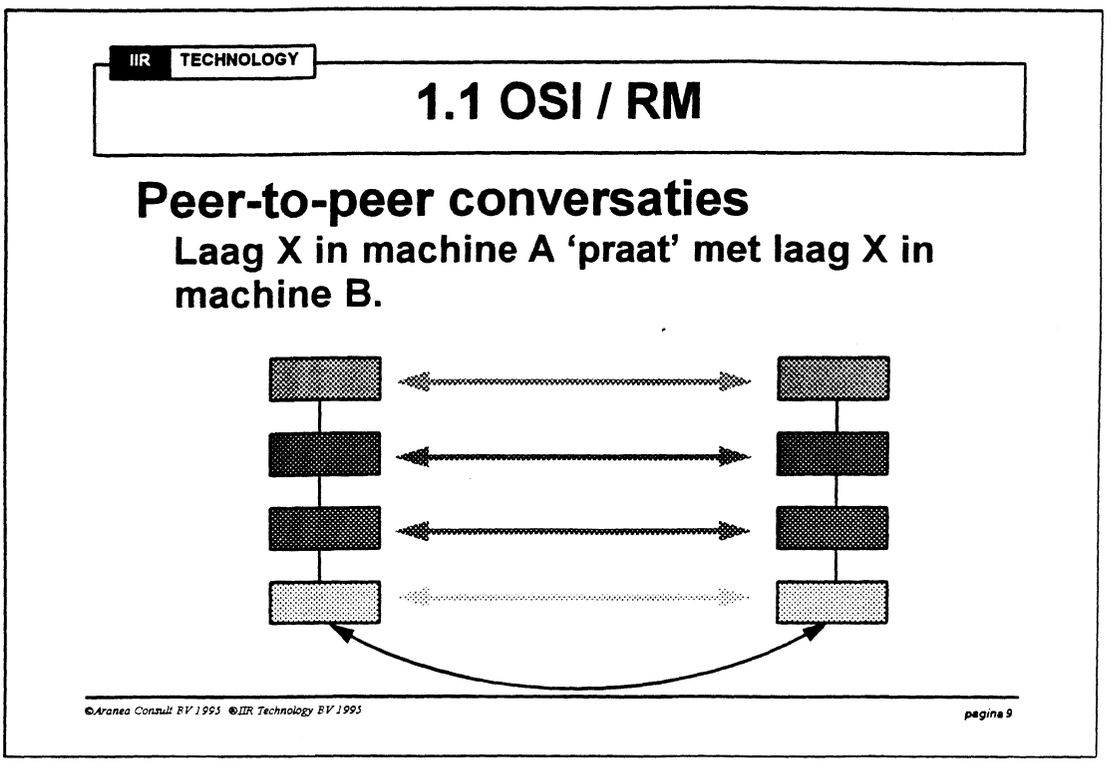


De bovenste drie lagen van het OSI model hebben te maken met de wijze waarop de toepassingen met de te versturen of ontvangen gegevens om moeten gaan. Daarom zijn dit de toepassings-gerichte lagen.

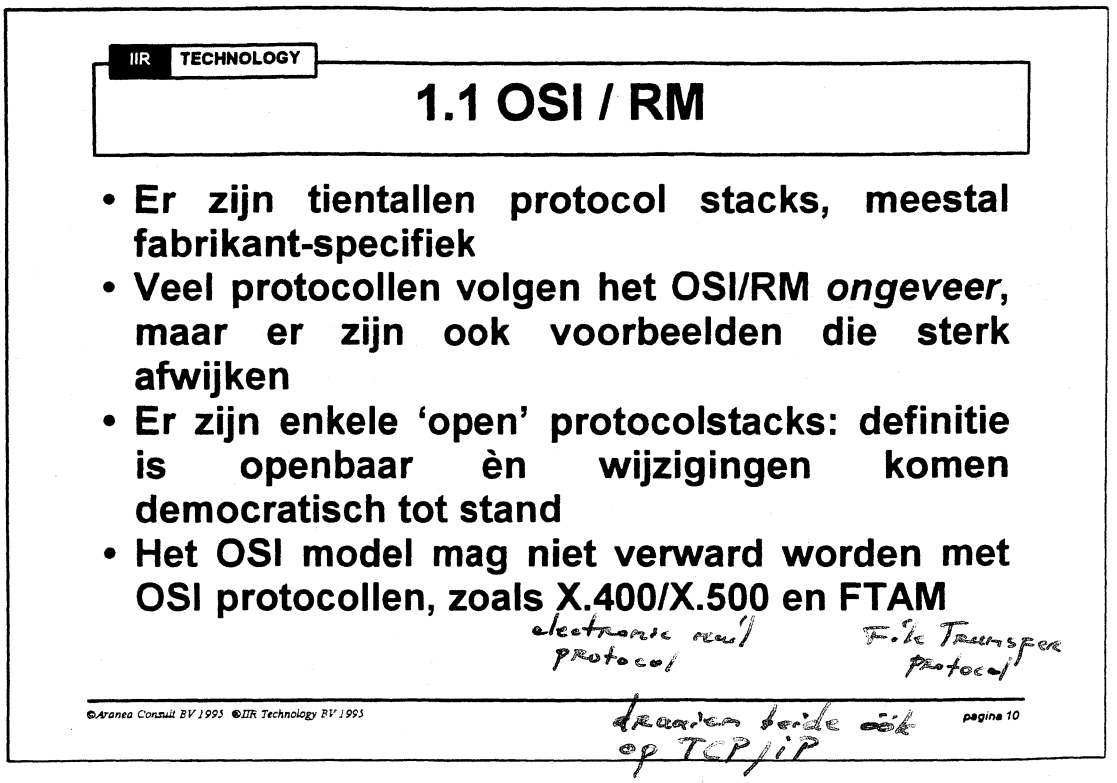
De onderste vier lagen houden zich alleen bezig met het transport van de gegevens tussen de twee toepassingen die aan het communiceren zijn. Daarom worden dit de transport- of communicatie gerichte lagen genoemd.

De bovenste vier lagen zijn alleen aanwezig in zogeheten 'end-systems'. Dit zijn systemen waarop toepassingen actief zijn, de feitelijke gebruikers van het communicatie netwerk.

De onderste drie lagen zijn in alle systemen aanwezig. Een systeem waarop geen toepassingen draaien, maar dat alleen een soort doorgeef functie heeft, wordt een 'intermediate system' genoemd. Hier zijn slechts de onderste drie lagen op geïmplementeerd.



Door het principe van de lagen, zijn er diverse interfaces nodig tussen de lagen onderling. Een willekeurige laag X krijgt een verzoek van de laag die er boven zit (laag X+1). De laag X zal zijn activiteiten uitvoeren, bijvoorbeeld het inpakken en het uitvoeren van checksum controles, en het geheel 'doorgeven' aan de laag die er onder zit (X-1). Laag X kan in feite alleen maar communiceren met eenzelfde laag X' op de machine waarmee gecommuniceerd wordt; laag X weet niet (precies) wat hij aangeleverd krijgt van laag X+1 en weet niet (precies) wat laag X-1 met het door hem aangeleverde zal gaan doen. Laag X maakt zich alleen maar zorgen over de laag X' op de machine waarmee gecommuniceerd wordt; diè moet begrijpen wat laag X bedoeld heeft. Dat wat lagen X+1 en X-1 verder nog met de informatie doen, is voor laag X irrelevant!



Er zijn tientallen fabrikant-specifieke protocollen. Soms zijn de specificaties daarvan geheim, soms worden ze wel gepubliceerd. Maar altijd heeft de fabrikant de macht om het protocol naar eigen inzicht en op het moment dat hem dat goed dunkt, te veranderen. Als een concurrent dat protocol heeft geïmplementeerd, moet hij maar zien dat hij het zo snel weet aan te passen. Voor klanten niet zo'n prettig gevoel.

Bij 'open' protocollen staan de specificaties onder 'democratische' controle; wijzigingen kunnen dus niet eenzijdig en bij verrassing worden opgelegd. Alle fabrikanten krijgen een faire kans om hun produkten aan te passen en kunnen hun klanten daarover geloofwaardige garanties geven.

Standaardisatie door een openbare commissie alléén biedt geen garantie voor openheid: het Token Ring protocol was een keurige IEEE standaard. Toch bracht IBM (marktaandeel 80-90%) eenzijdig en bij verrassing de snelheid van 4 Mb naar 16 Mb. Het kostte de andere fabrikanten *anderhalf jaar* voor ze betrouwbare 16 Mb Token Ring produkten konden leveren.

Een goed voorbeeld van een open protocol familie of *protocol stack* is TCP/IP. Daarnaast voldoen ook de OSI protocollen aan het criterium 'openheid', maar in de praktijk worden ze relatief weinig gebruikt, zodat 'standaardisatie op OSI' in de meeste gevallen geen oplossing is.

IIR TECHNOLOGY

1.2 Drie technieken

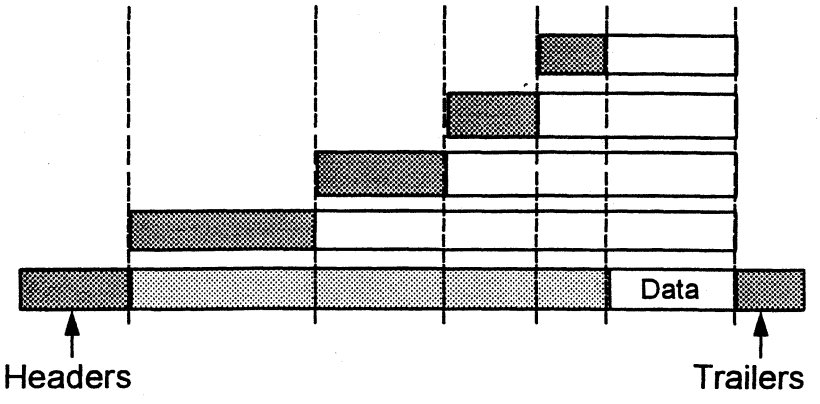
- In gelaagde modellen wordt gebruik gemaakt van 3 'technieken':

  1. (De)encapsulation
  2. (De)fragmentation
  3. (De)multiplexing

IIR TECHNOLOGY

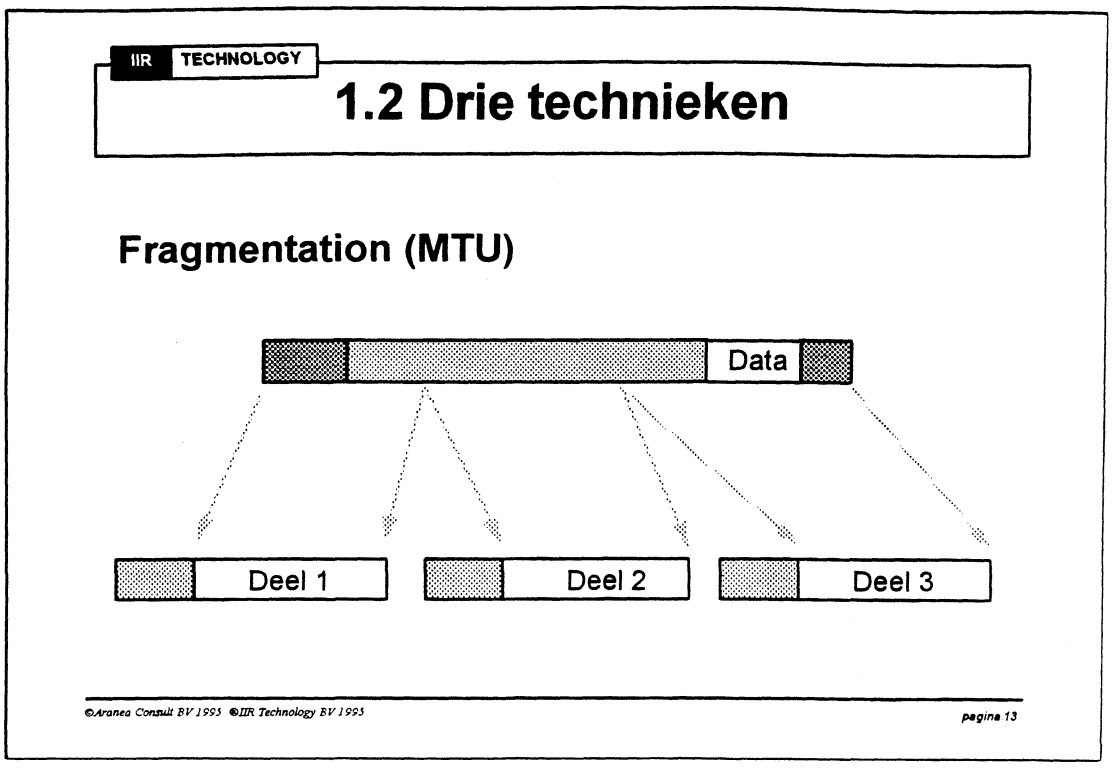
1.2 Drie technieken

Encapsulation



Eén van de gebruikte technieken in de lagen modellen is het inpakken ('encapsulation'). Iedere laag in het OSI model (op de slide zijn er gemakshalve 4 getekend) moet 'iets' uitvoeren met de informatie. Zoals besproken in de slide van de peer-to-peer protocollen, zal laag X met de overeenkomstige laag X' aan de andere kant communiceren. Dat betekent dat laag X een header toevoegt om laag X' op te hoogte te stellen van bijvoorbeeld voor welke toepassing de gegevens bedoeld zijn, het volgnummer, time outs, acknowledgements, etcetera. Laag X geeft het geheel door aan laag X-1 die daaronder zit, maar verder niets met de extra gegevens doet. Deze laag zal met X'-1 communiceren en er dus ook weer de benodigde informatie voor moeten plakken. Als dit laag na laag gebeurt, dan zal iedere laag dus datgene, dat aangereikt wordt door de laag erboven, inpakken en doorgeven aan de laag eronder. Met als resultaat dat er pakketten over het netwerk kunnen gaan met zo'n 58 bytes aan header en trailer informatie en 1 byte aan data!

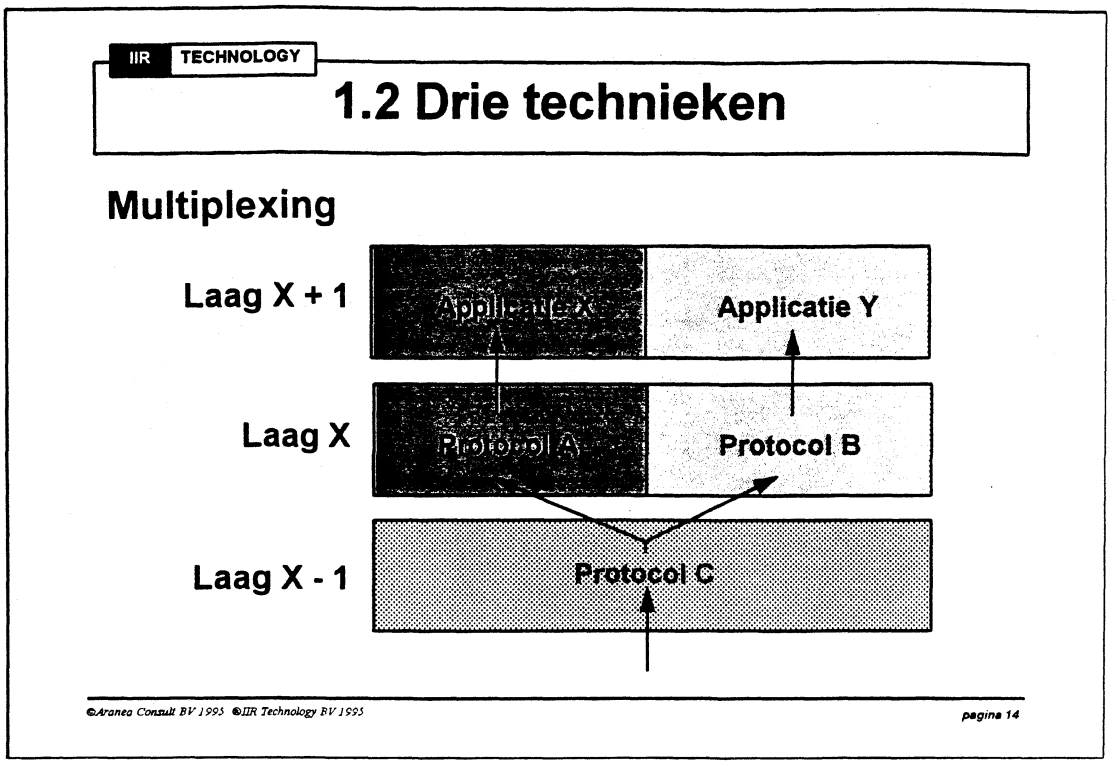
Algemeen kan worden gesteld dat voor laag X de eronder gelegen lagen niet bestaan, en dat laag X gebruikt maakt van de diensten van laag X-1, en diensten levert aan laag X+1.



Fragmentatie heeft te maken met de *grootste eenheid* van te versturen data. Deze grootste eenheid wordt MTU genoemd (=Maximum Transfer Unit). Het mag duidelijk zijn dat er zoiets als een 'grootste pakketje' moet zijn. Anders zou een machine een file van 1 Gb naar een andere machine kunnen verzenden, waarbij waarschijnlijk het onderliggende netwerk een behoorlijke tijd bezet blijft.

De MTU is recht evenredig met de datatransmissiesnelheid voor de betreffende datalink laag. Hoe sneller de datalink laag, hoe groter de eenheid van te versturen data mag zijn. Op 'langzamere' datalink lagen, mogen de frames natuurlijk niet te groot zijn, omdat deze anders de betreffende datalink laag kunnen monopoliseren. Een aantal MTU's als voorbeeld:

Network	MTU (bytes)
16 Mbps TRN	17914
4 Mbps TRN	4464
FDDI	4352
Ethernet	1500
IEEE 802.3/.2	1492
X.25	576
Point-point	296



Een tweede techniek die het gevolg is van het lagenmodel, is (de)multiplexing. Iedere laag kan *meerdere* implementaties bevatten: de transport*laag* beschrijft enkel welke *functionaliteit* zich op die laag moet bevinden. Daadwerkelijke implementaties kunnen besluiten om slechts een subset van deze functionaliteit te implementeren. Zo kan het vervolgens voorkomen dat er op één laag verschillende implementaties kunnen voorkomen. Dat betekent dat een laag X moet *weten* aan welke implementatie op de laag X+1 het binnenkomende pakketje moet worden gegeven.

Concreet voorbeeld: boven op laag 3, het IP protocol, draaien zowel het TCP als het UDP protocol. Dit zijn beide verschillende implementaties van een transportlaag protocol. IP aan de ontvangende kant zal dus moeten weten of hij het moet afleveren bij TCP of UDP. Dit gebeurt op basis van het zogenoemde demultiplexing element.

Een voorbeeld dat niet direct aan TCP/IP gerelateerd is. Wellicht is het bekend dat het mogelijk is om op een PC met zogenaamde 'multi protocol stacks' te werken. Zo kan bijvoorbeeld zowel VINES/IP, als TCP/IP gebruikt worden. Met VINES/IP kan het kantoorautomatiseringsplatform bereikt worden, met TCP/IP kan het RS/6000 platform bereikt worden met daarop bijvoorbeeld Prolin of LIMS.

IIR TECHNOLOGY

1.3 TCP/IP

- **Ontstaan als protocol voor het ARPA netwerk in 1969**
- **Ontworpen voor WAN, later ook LAN**
- **Gratis verspreid via Berkeley Unix in '84**
- **Voor alle computerarchitecturen beschikbaar**
- **Protocol van 'Internet'**
- **Ontwikkeling op basis van RFC's (*Request For Comment*)**

De TCP/IP protocollen werden oorspronkelijk ontwikkeld als 2e generatie protocol voor ARPAnet, een communicatie netwerk dat door het Amerikaanse ministerie van defensie werd aangelegd om de universiteiten en laboratoria waar militaire contract-research werd gedaan, met elkaar te verbinden. Dit netwerk vormde de start van wat nu het Internet is, een wereldwijd netwerk met circa 20 *miljoen* gebruikers. Het Internet verdubbelt zich elk jaar en juist dit jaar begint het ook bij het grote publiek bekend te worden. Waarschijnlijk is over 3 jaar een aansluiting op Internet even normaal als nu een fax apparaat.

Ruim tien jaar geleden werd de UNIX versie die door de University of California in Berkeley werd verspreid, voorzien van een TCP/IP implementatie. Daarmee werd TCP/IP 'gezaaid' op duizenden universiteiten in de hele westerse wereld. Omdat daarbij ook de source (programmatekst) mee werd geleverd, was het plotseling vrij makkelijk om ook op andere computers TCP/IP te implementeren. Als een organisatie vier computers heeft waarvan er drie TCP/IP 'praten', is de druk groot om ook op de vierde TCP/IP te implementeren. Binnen enkele jaren was het dan ook zo ver dat voor vrijwel alle computers TCP/IP beschikbaar was, en voor de populaire typen was er al snel keuze: gratis, goedkoop of wat duurder, eenvoudig, snel of goed ondersteund.

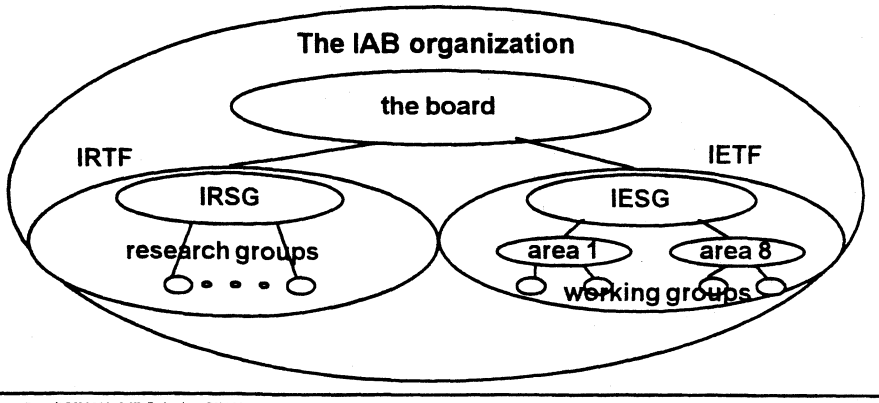
De ontwikkeling van de TCP/IP protocollen gaat nog steeds door. Daarvoor kan iedereen voorstellen doen in de vorm van een Request For Comment (RFC) (dus geen *standaard* (OSI) of *aanbeveling* (CCITT), maar een uitnodiging tot discussie!). Als er voldoende steun voor een RFC is, kan het een voorlopige status krijgen. Gedurende een half jaar wordt er mee geëxperimenteerd. Lijkt het daarna nog steeds een goed idee, dan kan het als een eventueel verplicht onderdeel in TCP/IP worden opgenomen.

Door de enorme groei van het Internet zullen de netwerkadressen over een paar jaar 'op' zijn. Dat zal tot een tamelijk ingrijpende wijziging leiden, waar nu al druk op gestudeerd wordt.

IIR TECHNOLOGY

1.3 TCP/IP

- IRTF = Internet Research Task Force
- IETF = Internet Engineering Task Force
- IRSG = Internet Research Steering Group
- IESG = Internet Engineering Steering Group



Om alle ontwikkelingen rondom TCP/IP in goede banen te kunnen leiden, is in 1992 de Internet Society opgericht (ISOC). Het ISOC houdt zich bezig met de promotie van Internet, het sturen en begeleiden van nieuwe standaards voor Internet, administratieve zaken en samenwerking met andere informatica organisaties, waaronder het ITU.

Eén van de taken van het ISOC, is het ondersteunen van IAB (Internet Architecture Board, voorheen Internet Activities Board) dat zich bezighoudt met de ontwikkeling van het Internet.

Het IAB is verdeeld in twee taskforces en een informatie centrum.

Het IETF verzorgd de standaardisatie en de implementatie van de protocollen, het IRTF onderzoekt nieuwe ideeën.

Het NIC (Network Information Centre) houdt zich bezig met de uitgifte en registratie van IP domains en adressen.

IIR TECHNOLOGY

1.3 TCP/IP

- Protocol: Initial, Proposed, Draft, Standard, Historic

- Status van een protocol:

• Required	Verplicht op alle hosts
• Recommended	Aanbevolen
• Elective	'U ziet maar'
• Limited use	Niet voor general purpose
• Not recommended	Afgeraden

2e dyv. RFC 1149 van 01-04-1990

IIR TECHNOLOGY

1.3 TCP/IP

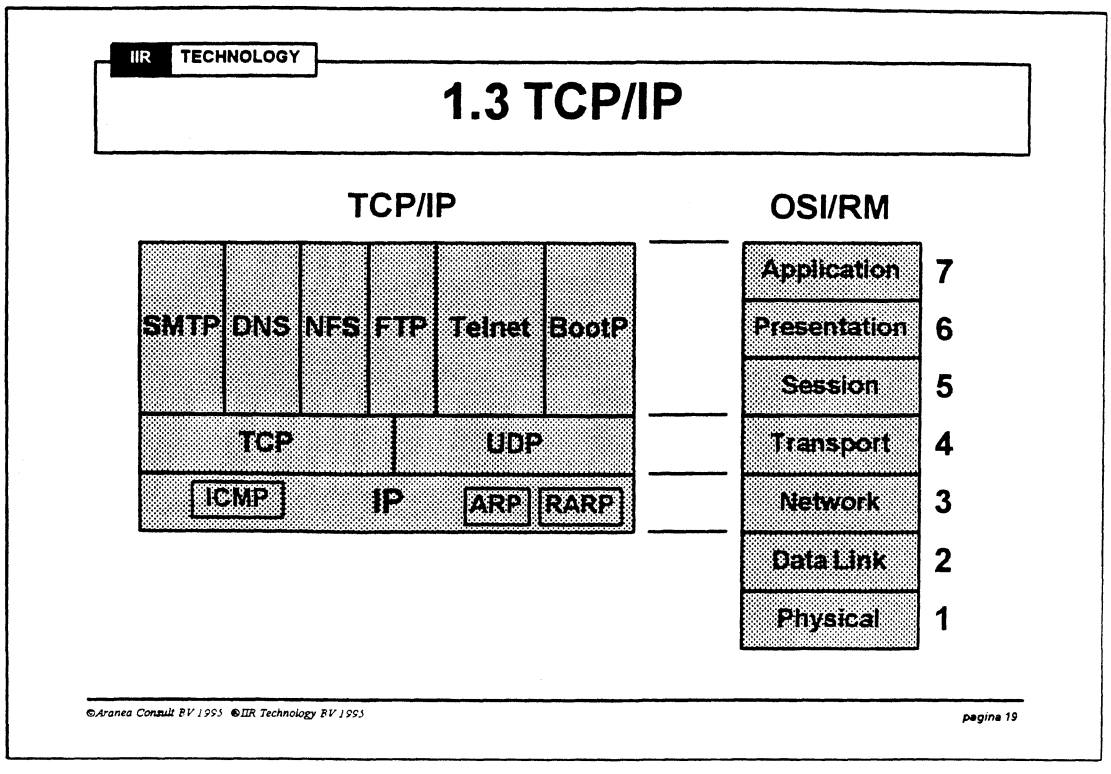
- **Internet**
  Benaming voor de verzameling van netwerken, opgebouwd rondom ARPANET, MILNET en NSFnet, die tezamen de backbone van de academische wereld vormen in USA

- **internet**
  Generieke term om een aantal aan elkaar gekoppelde netwerken te benoemen

Let even op het verschil tussen Internet (met een hoofdletter) en internet (met een kleine letter).

Het Internet is de aanduiding voor het wereldwijde netwerk dat op dit moment zo in de belangstelling staat en zijn oorsprong heeft in ARPAnet en MILnet. military Advanced Research

Een internet daarentegen, is de aanduiding voor een aantal aan elkaar gekoppelde netwerken, openbaar of privé. Het Internet is dus wel een internet, maar omgekeerd is dat zeker niet het geval.



De structuur van de TCP/IP familie past redelijk in het OSI Referentie Model:

De onderste twee lagen maken *geen* onderdeel uit van TCP/IP. De TCP/IP 'gemeenschap' is er altijd vanuit gegaan dat TCP/IP moet kunnen werken op alle bestaande onderliggende datalink lagen, en heeft hier dus geen effort ingestoken.

De netwerklaag kent één protocol:

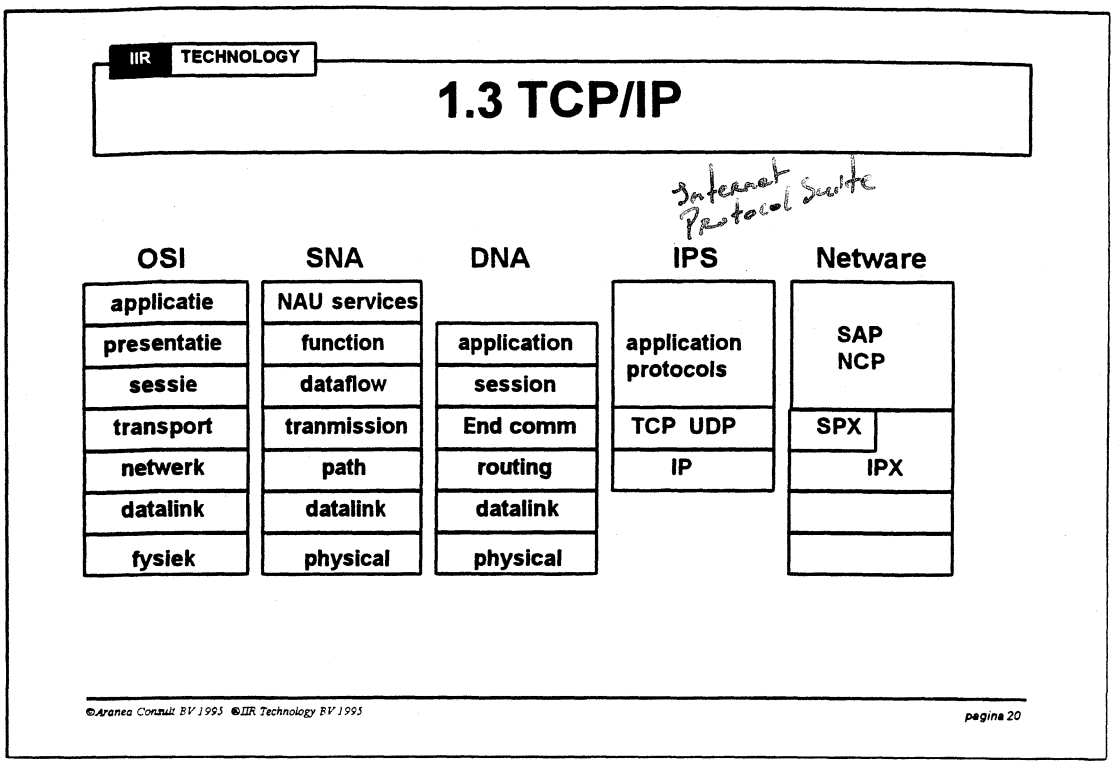
- **IP** - een zogenaamd connection less, best effort delivery system. Kan met vrijwel alle bestaande datalink lagen samenwerken. Dat maakt TCP/IP onafhankelijk van de netwerk techniek.

Er is een keuze uit twee transport lagen:

- **UDP** - voor korte, losse berichtjes (bv. "hoe laat is het?" "10:26").
- **TCP/IP** - voor het 'echte' werk. Een zeer krachtig protocol dat zich automatisch aanpast aan de situatie in het netwerk. Daardoor is het zowel over een lokaal netwerk als over een satelliet-verbinding redelijk efficient.

De OSI lagen 5 t/m 7 zitten min of meer verstopt in de TCP/IP applicaties, bv.:

- **Telnet** - het virtual terminal protocol. Met Telnet kan de gebruiker van systeem A inloggen op systeem B (in de volgende kamer of aan de andere kant van de wereld).
- **FTP** - een programma om bestanden (files) tussen twee computers uit te wisselen. De belangrijkste opdrachten zijn *get* en *put*, waarbij zowel de remote als de lokale filenaam worden opgegeven, maar daarnaast zijn er mogelijkheden om bv. directories te bekijken.
- **NFS** - een protocol waarmee de bestanden van de ene computer door een andere computer kunnen worden gebruikt, precies zoals dat bij een Netware of Vines server het geval is.



Hierboven zijn de huidige, meest gebruikte protocolstapels naast elkaar gezet. Omdat OSI, SNA (Systems Network Architecture van IBM) en DNA (Digital Network Architecture van DEC) architectuur modellen zijn, wordt daarbij geen invulling gegeven aan de lagen met protocollen, maar met een naam die verwijst naar de functie van de betreffende laag.

In SNA worden bijvoorbeeld SDLC, X.25 en 802.5 ondersteunt als datalink protocollen, in DNA DDCMP, 802.3, 802.4 en 802.5 alsmede X.25.

TCP/IP en Netware daarentegen zijn protocolstapels die opgebouwd zijn uit een aantal bestaande protocollen of afgeleid van bestaande protocollen. De meeste protocollen van zowel TCP/IP als Netware vinden hun oorsprong uit XNS, de protocollen die ooit door Xerox ontwikkeld zijn voor de communicatie tussen computers.

IIR TECHNOLOGY

1.4 Samenvatting

- **Protocollen om communicatie mogelijk te maken**
- **Protocollen zijn 'gelaagd' vanwege flexibiliteit**
- **Vanwege deze gelaagdheid drie technieken**
- **Eén 'godfather' van de lagenmodellen: OSI**
- **TCP/IP is ook 'een' lagenmodel**

Samengevat zijn er protocollen nodig om zinvol te kunnen communiceren. Omdat er meestal meerdere protocollen op verschillende niveaus gelijktijdig gebruikt worden, wordt er voor de communicatie tussen computers gebruik gemaakt van een protocol stapel waarin voor elk te onderscheiden niveau één of meerdere protocollen gebruikt kunnen worden.

Het OSI model is een referentiemodel voor protocol stapels, maar de praktische toepasbaarheid ervan is nihil.

Een veel praktischer lagenmodel is TCP/IP, hoewel dit eigenlijk een verkeerde aanduiding is. Officieel moet gesproken worden van de Internet Protocol Suite (IPS), waarvan TCP en IP slechts 2 protocollen zijn.

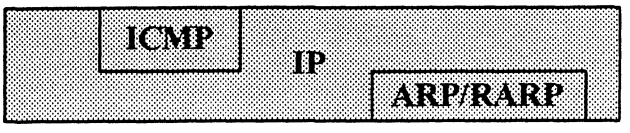
IIR TECHNOLOGY

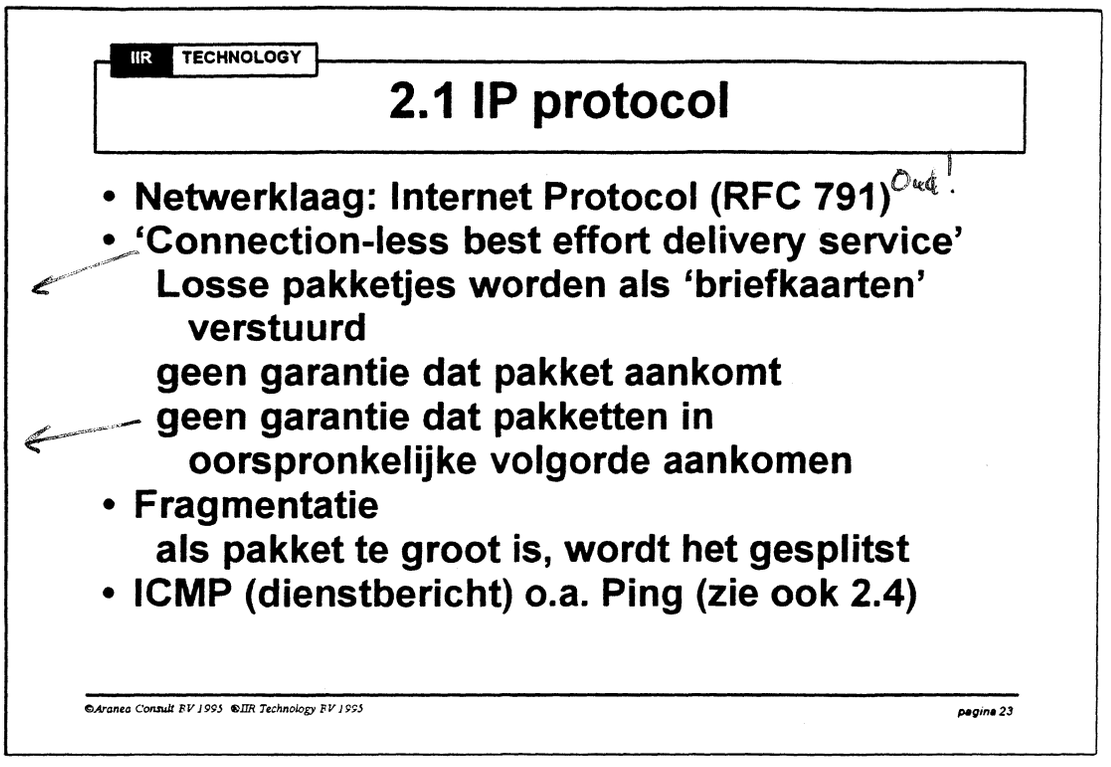
2. De netwerklaag

2.1 IP Protocol
2.2 Op welke onderliggende lagen

Intermezzo: Netwerktechnieken

2.3 Van logisch naar fysiek
2.4 ICMP protocol
2.5 Samenvatting





Het Internet Protocol (IP) heeft als taak 'datagrammen' door het netwerk te loodsen. Daarbij geeft IP geen garanties: het is een 'best effort' service.

Als één van de bovenliggende lagen (UDP of TCP) een bericht wil versturen, geeft hij dat aan IP, samen met het IP-adres van de bestemming. IP behandelt elk bericht onafhankelijk van het voorgaande of volgende; als de volgorde belangrijk is, moet de transportlaag dat maar regelen.

Als een echte pakketdienst plakt IP een vrachtbrief op het bericht: het bericht wordt voorzien van een **header**, waarin de informatie staat die voor IP en zijn collega-IP's in andere computers in het net belangrijk is: de lengte van het bericht, de afzender en de bestemming, voor welk transportprotocol het bericht bestemd is (UDP of TCP) en een identificatienummer.

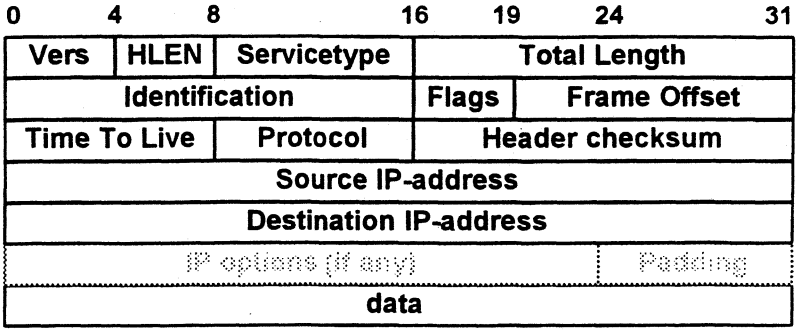
Juist omdat TCP/IP onafhankelijk is van de onderliggende netwerktechniek, is het mogelijk dat een bericht langer is dan het grootst mogelijke frame. In ARCnet mag een frame niet meer dan 512 bytes bevatten, terwijl er bij Token Ring meer dan 4000 bytes in één frame mogen worden verstuurd. Als het bericht te lang is, wordt het door IP gefragmenteerd: het bericht wordt in willekeurige stukken gebroken, die elk als afzonderlijk datagram worden verstuurd. In de header wordt dan wel aangeven welk deel van het oorspronkelijke bericht het is. De IP in de bestemmingscomputer moet immers de brokstukken weer aan elkaar lijmen om tenslotte het oorspronkelijke bericht aan de 'klant' (UDP of TCP) te overhandigen.

Een belangrijk onderdeel van TCP/IP is het Internet Control Message Protocol (ICMP), waarmee 'dienstberichten' tussen de IP's worden uitgewisseld. Zo wordt er, als de bestemming van een bericht niet bereikbaar is, een ICMP bericht 'host unreachable' naar de afzender gestuurd.

IIR TECHNOLOGY

2.1 IP protocol

- **IP netwerklaag doet aan encapsulation**



De IP header bevat de volgende velden:

Version - het versienummer van het IP protocol: dit veld maakt het mogelijk geleidelijk naar een nieuwe versie van IP te migreren. De huidige versie is 4, de volgende versie (IP next generation) zal 6 zijn.

HLEN - length van de IP-header, uitgedrukt in 32-bits woorden.

Service type - prioriteit (0-7) en de opties *low delay*, *high throughput* en *high reliability*.

Total Length - lengte van het hele datagram (header + data) in bytes, maximaal 64k bytes (indien fragmentatie: lengte van het fragment).

Identification - een unieke code die gebruikt wordt om fragmenten van een datagram weer bij elkaar te passen.

Flags - Twee bits: More geeft aan dat dit *niet* het laatste fragment is, Don't Fragment geeft aan dat een datagram niet gefragmenteerd *mag* worden (3e bit gereserveerd).

Fragment Offset - geeft aan welk deel van een gefragmenteerd datagram dit is. Fragmentatie vindt altijd plaats op een veelvoud van 8 bytes, daarom bevat dit veld 1/8e van de byte-offset.

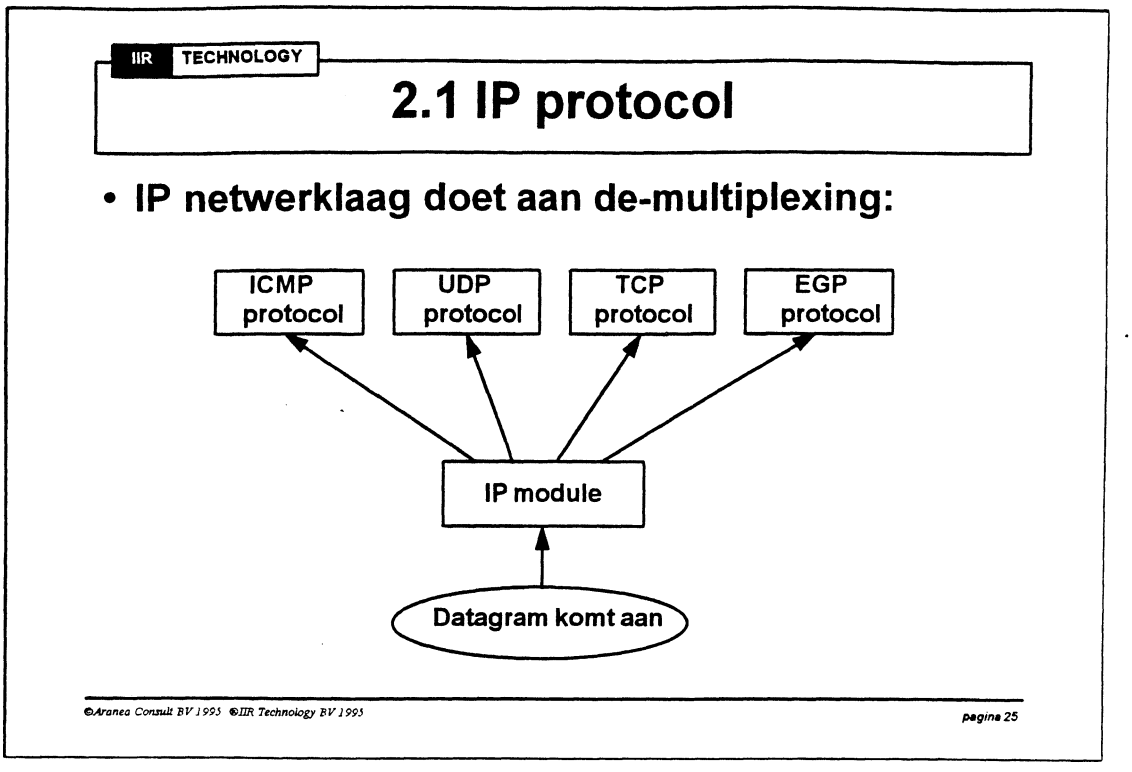
Time To Live - een 'tijdbom' die zorgt dat een datagram niet eindeloos door het netwerk kan zwerven.

Protocol - een code voor het hogere-laag protocol dat dit datagram heeft verstuurd (code 6 voor TCP en 17 voor UDP).

Header Checksum - controle op de consistentie van de IP-header.

Source en Destination IP address - zie verder.

Options en Padding - Optionele informatie in header: Record Route (bv PING -R), Loose/Strict Source Routing, TimeStamp, etcetera

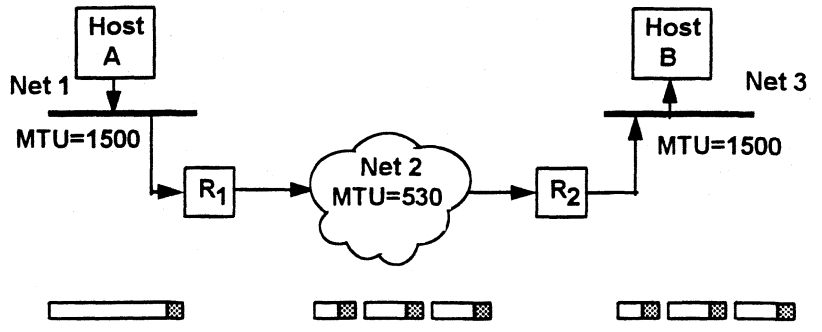


Op basis van het protocol number kan IP (de)multiplexen. Protocol nummer 1 is voor ICMP, protocol nummer 6 is TCP, protocol nummer 17 is UDP, protocol nummer 8 is voor EGP.

IIR TECHNOLOGY

2.1 IP protocol

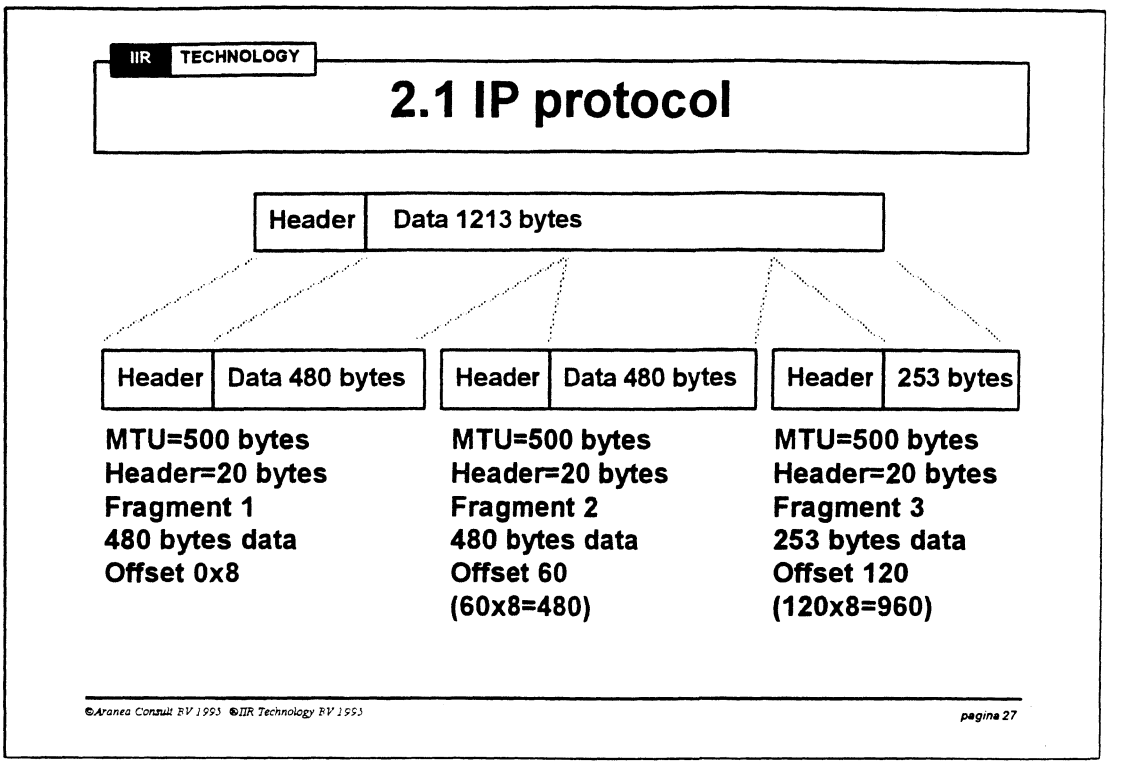
- **IP netwerklaag doet aan fragmentatie**



IP is ook verantwoordelijk voor het fragmenteren van datagrammen die te groot zijn voor het onderliggende netwerk. Elke netwerk technologie kent een *Maximum Transfer Unit* (MTU). Voor Ethernet is dat circa 1500 bytes, voor ARCnet maar iets meer dan 500, etc.

Elk fragment krijgt een kopie van de IP-header (met kleine wijzigingen). Elk fragment reist vervolgens als een zelfstandig datagram door het internet. Fragmenten kunnen dus langs verschillende wegen reizen en in een afwijkende volgorde aankomen. Ook is het mogelijk dat een fragment opnieuw wordt gefragmenteerd als het een netwerk tegenkomt met nog kleinere MTU.

Merk op dat de fragmenten pas op de uiteindelijke bestemming weer worden samengevoegd!



Op de slide een voorbeeld van een mogelijke fragmentatie, uitgevoerd door de IP laag. Het oorspronkelijke datagram van 1213 bytes is opgedeeld in 3 fragmenten van respectievelijk 480, 480 en 253 bytes. Let op de waarden voor het veld 'fragment offset'.

IIR TECHNOLOGY

2.1 IP protocol

- **Fragmentation control wordt gerealiseerd met behulp van 4 velden in de IP header:**
  1. ***Identification***
  2. ***Fragment Offset***
  3. ***Flags***
  4. ***Total length***
- **Soms is fragmentatie niet gewenst: 'Don't fragment' bit in het flags field**
- **Fragmenten worden *nooit* onderweg weer aan elkaar geplakt, maar door de eindbestemming!**

De velden op de sheet zijn de velden die gebruikt worden in de IP header om ervoor te zorgen dat fragmentatie goed uitgevoerd kan worden.

Identification - ieder fragment van een te verzenden datagram wordt voorzien van eenzelfde ID. Dit is een random gegenereerd, 16 bits getal. Alle binnenkomende fragmenten met hetzelfde ID behoren tot hetzelfde IP datagram.

Frags - Met behulp van een bit in het flags field kan de zendende IP host aangeven of er 'more fragments' verstuurd zijn. Met behulp van het bit 'Don't fragment' kan de zendende IP host aangeven dat het betreffende datagram onderweg niet gefragmenteerd mag worden. Want, zoals we al eerder zagen, ook internetworking devices moeten rekening houden met MTU's, en als zij te grote pakketten moeten routeren, zullen ook internetworking devices gaan fragmenteren! (nb het derde bit in het flags field is reserved).

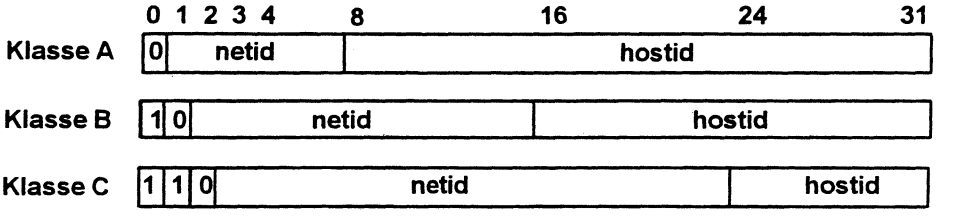
Fragment offset - dit 16 bits veld geeft de offset van het betreffende fragment aan ten opzichte van het oorspronkelijke datagram.

Total length - dit veld wordt aangepast als het datagram moet worden gefragmenteerd, zodat ieder fragment een total length waarde bevat voor dat betreffende fragment.

IIR TECHNOLOGY

2.1 IP protocol - adressen

- **IP adressen bestaan uit 32 bits, 4 bytes**
- **deze bytes worden decimaal weergegeven, gescheiden door een punt, bijv. 145.46.203.254**
- **afhankelijk van de *klasse* is er een scheiding in *netwerk-gedeelte* en *host-gedeelte***



Een heel belangrijk onderdeel van IP is de adressering.

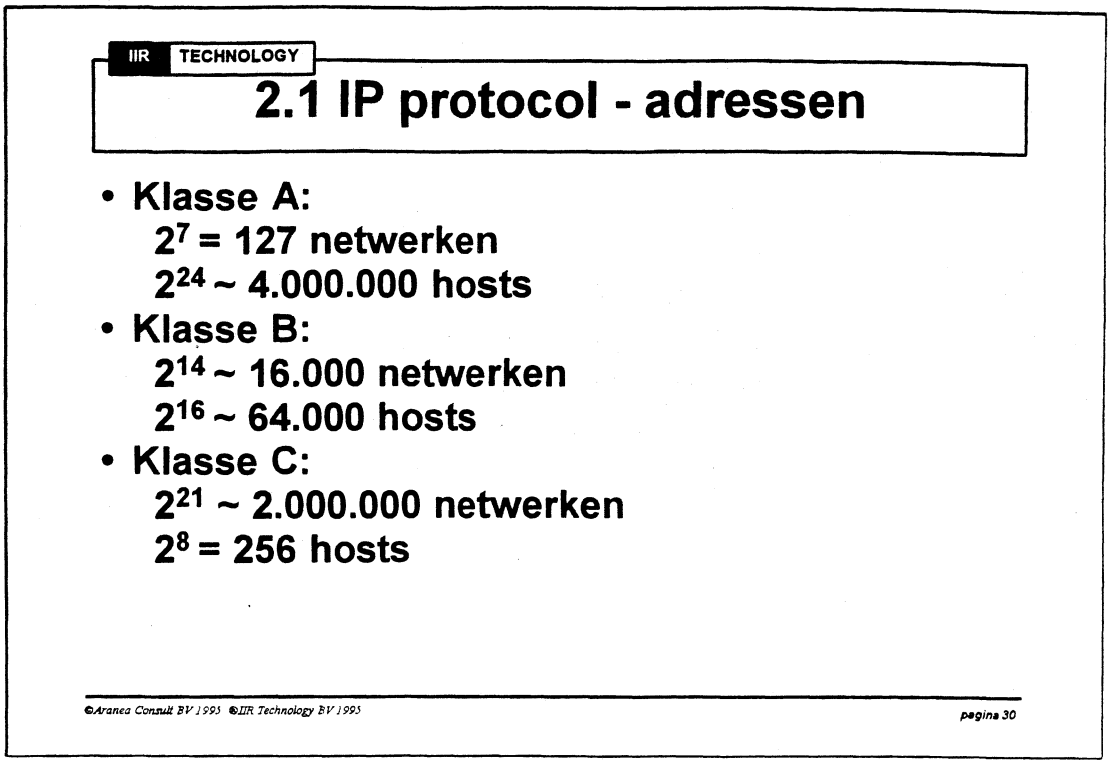
Een IP-adres bestaat uit 4 bytes (32 bits). Als een IP-adres moet worden opgeschreven, schrijven we de decimale waarde van elk byte, gescheiden door punten. Een byte (8 bits) kan een waarde tussen 0 en 255 hebben.

We maken eerst een vergelijking met het telefoonnet. Een telefoonnummer (telefoon-adres) bestaat uit twee niveaus: het netnummer en het abonneenummer. Bij het opzetten van een gesprek kijken de telefooncentrales eerst naar het netnummer en pas als er een verbinding naar de juiste stad is gemaakt naar het abonneenummer.

Ook het IP-adres bestaat uit twee delen, het netwerk-nummer en het host-nummer (in TCP/IP is elke computer een 'host'). Net zoals er bij telefoonnummers korte en lange netnummers zijn (040 en 01180), zijn er IP-adressen met verschillende verdelingen in netwerknummer en hostnummer.

- Als het meest significante bit van het eerste byte 0 is, hebben we met een klasse A adres te doen. De laatste 3 bytes vormen dan het hostadres. Het eerste byte heeft een waarde tussen 0 en 127.
- Als de twee meest significante bits van het eerste byte 10 zijn, hebben we met een klasse B adres te doen. De laatste 2 bytes vormen dan het hostadres. Het eerste byte heeft een waarde tussen 128 en 191.
- Als de drie meest significante bits van het eerste byte 110 is, hebben we met een klasse C adres te doen. Het laatste byte vormt dan het hostadres. Het eerste byte heeft een waarde tussen 192 en 223.

Er bestaan ook nog klasse D adressen. Deze adressen worden als multicast adressen gebruikt en kunnen interessant zijn om door routers gebruikt te worden om routeringsinformatie uit te wisselen.

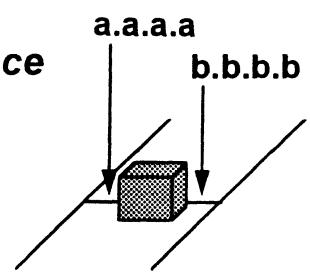


Bovenstaand een indicatie over de IP netwerken. Klasse A adressen zijn al lang niet meer te krijgen, klasse B sporadisch, en klasse C netwerken beginnen ook schaars te worden.

IIR TECHNOLOGY

2.1 IP protocol - adressen

- Alle bits 0 -> '*dit*'; 145.46.0.0
- Alle bits 1 ->'*alle*'; 145.46.255.255
- Adres 0.0.0.0 = deze host op dit netwerk
- 127.x.x.x is het loopback adres
- IP-adres: het adres van een *interface*
- Speciale netwerkadressen:
  - 10.*.*.*
  - 172.16.*.* .... 172.31.*.*
  - 192.168.*.*
  - Te gebruiken als 'private address space', worden niet in Internet doorgegeven!



Een zéér belangrijke opmerking: IP adressen zijn adressen van interfaces, niet van hosts! Als een bepaalde host dus met twee Ethernet kaarten aan (twee verschillende) netwerken hangt, dan is deze host dus ook met twee adressen te bereiken! Een voorbeeld hiervan zijn de routers.

Een aantal adressen is als 'speciale' adressen te betitelen. Op de sheet is een aantal voorbeelden van deze adressen opgenomen.

Als alle bits '0' zijn, betekent dat 'dit' of 'deze'. Zo betekent 145.46.0.0 zoveel als '*deze* host op netwerk 145.46'. Dit had overigens ook met het adres 0.0.0.0 bereikt kunnen worden (wat zoveel betekent als '*deze* host op *dit* netwerk' (zie ook slide). Dit adres wordt vaak gebruikt door BootP/DHCP clients. Zie ook hoofdstuk 6.

Alle bits op '1' (in de literatuur ook wel aangegeven met decimale waarde -1), betekent 'alle'. Zo betekent 145.46.255.255 een broadcast naar *alle* hosts op netwerk 145.46. Dit wordt overigens een *directed broadcast* genoemd; er wordt expliciet een netwerk geadresseerd. Als het adres 255.255.255.255 wordt gebruikt, dan wordt dit een *limited broadcast* genoemd. Dit adres wordt gebruikt om te broadcasten naar '*alle* hosts op dit netwerk'. Dit wordt dus ook niet door een router doorgegeven.

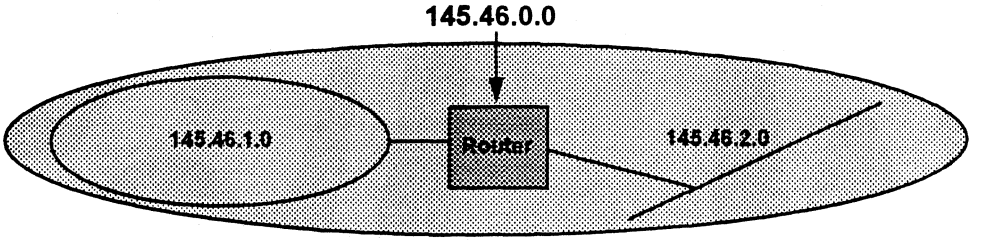
Er is een speciaal adres 127.x.x.x gedefinieerd. Dit is de zogenaamde loopback driver. Deze driver wordt gebruikt om de *eigen* IP stack te testen. Als het mogelijk is om met test programmatuur adres 127.0.0.0 te bereiken, dan is het in ieder geval zeker dat de eigen IP stack goed functioneert. Overigens zijn er maar weinig PC implementaties die dit speciale adres ondersteunen!

Een aantal IP adressen is gedefinieerd als IP adres dat niet middels het wereldwijde Internet doorgegeven zal worden.

IIR TECHNOLOGY

2.1 IP protocol - subnetting

- **Bij subnetting wordt er een gedeelte van het *host-gedeelte* van het IP adres gebruikt als *sub-netwerk* nummer**
- **Noodzakelijk bij meerdere fysieke netwerken**
- **Aangeven met behulp van 'subnet mask'**



Indien in een organisatie meerdere fysieke netwerken voorkomen, bijvoorbeeld een Token-Ring netwerk en een Ethernet netwerk, dan kunnen deze netwerken alleen maar goed aan elkaar gekoppeld worden met behulp van een router. Dit impliceert echter dat aan beide kanten van de router verschillende logische IP netwerken worden gedefinieerd. Immers, routers zijn internetworking devices die verschillende (logische) netwerken aan elkaar koppelen (zie ook hoofdstuk 3). Daarnaast zijn er wellicht ook nog andere redenen aan te voeren om intern met meerdere logische netwerken te gaan werken, en 'naar buiten toe' als één netwerk op te treden. Dit kan gerealiseerd worden met behulp van subnetting.

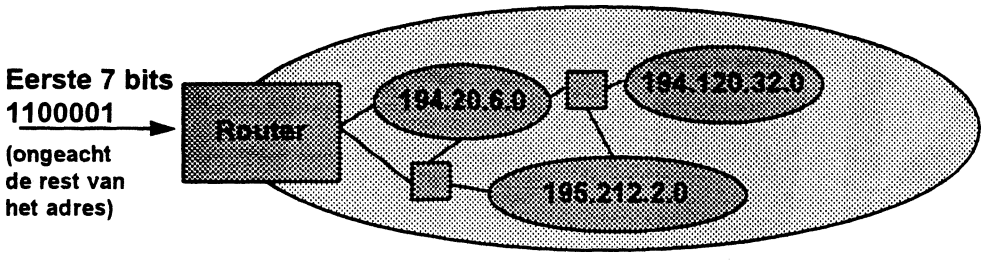
Bij subnetting wordt een gedeelte van het host-nummer van het IP adres gereserveerd als sub-netwerk nummer. Dit wordt aangegeven met behulp van een subnet masker. Een voorbeeld. Een klasse B adres gebruikt de eerste twee bytes als netwerk-nummer, de laatste twee bytes als host-nummer. Het masker 255.255.255.0 specificeert nu dat het volledige 3e byte als subnetwerk-nummer moet gaan fungeren (door een logische AND operatie te doen met het betreffende IP adres blijven alle bits die meedoen in het (sub)netwerk-gedeelte hun oorspronkelijke waarde behouden). Dit betekent dat bij een subnet masker van 255.255.255.0, de machines 145.46.3.46 en 145.46.3.201 tot hetzelfde subnetwerk behoren. Immers, volgens het masker is het volledige 3e byte het subnetwerk-nummer, en dat is gelijk bij beide adressen.

Zo is het ook mogelijk om een subnet mask van 255.255.252.0 te hebben. Als we dit uitschrijven, krijgen we 11111111.11111111.11111100.00000000. Met andere woorden, alleen de eerste 6 bits van het 3e byte behoren tot het subnetwerk-gedeelte. Dit betekent dat bij het gebruik van subnet masker 255.255.252.0, de hosts 145.46.201.20 en 145.46.203.254 tot hetzelfde subnetwerk behoren. Immers, het derde byte van deze adressen is respectievelijk 110010|01 en 110010|11. De eerste 6 bits zijn identiek, en dat waren volgens het masker ook de bits die identiek *moesten* zijn!

IIR TECHNOLOGY

2.1 IP protocol - supernetting

- Naast *opdelen* ook *aggregeren*: supernetting (ook wel 'CIDR' genoemd, RFC's 1518, 1519)
- achterliggende gedachte is om routing tabellen zo klein mogelijk te houden
- Nog maar op kleine schaal toegepast



Naast subnetting, is ook supernetting gedefinieerd. In de RFC's wordt er gesproken van Classless InterDomain Routing. In plaats van het opdelen van een bestaand IP adres in 'kleinere' subnetwerk adressen, wordt in het geval van supernetting een aantal netwerken geaggregeerd. Hiermee wordt geprobeerd routing tabellen kleiner te houden. Immers, van de samengevoegde netwerken is nog maar één entry nodig, namelijk de entry naar de router die precies weet wat er allemaal 'achter' hem ligt. In RFC 1466 wordt gesuggereerd om alle klasse C netwerken die er voor Europa worden uitgegeven, te voorzien van adressen in de range van 194.0.0.0 tot 195.255.255.255. Op die manier zou alle verkeer voor Europa via één centrale, supernet router kunnen lopen. De netwerken uit deze range hebben namelijk gemeen dat ze dezelfde eerste 7 bits van het eerste byte hebben ('1100001', zie ook slide).

Er is een aantal voorwaarden alvorens aan supernetting gedaan kan worden:

1. De IP netwerken die geaggregeerd moeten worden, dienen in een range te zitten waarvan de high-order bits overeenkomen (hoeveel bits is vrij).

2. De routing tabellen en routing algorithmes moeten zowel met 32 bits adressen, als 32 bits maskers kunnen omgaan.

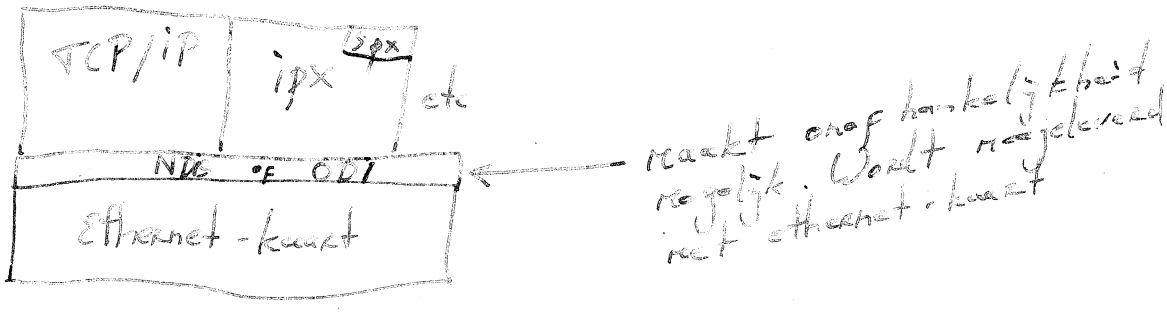
3. De routing protocollen moeten zodanig uitgebreid worden dat ze 32 bits maskers, naast 32 bits adressen, kunnen vervoeren (zowel RIP-2, als OSPF voldoen hieraan, zie hoofdstuk 5).

IIR TECHNOLOGY

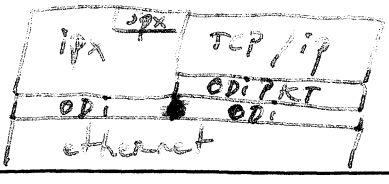
2.2 Op welke onderliggende lagen

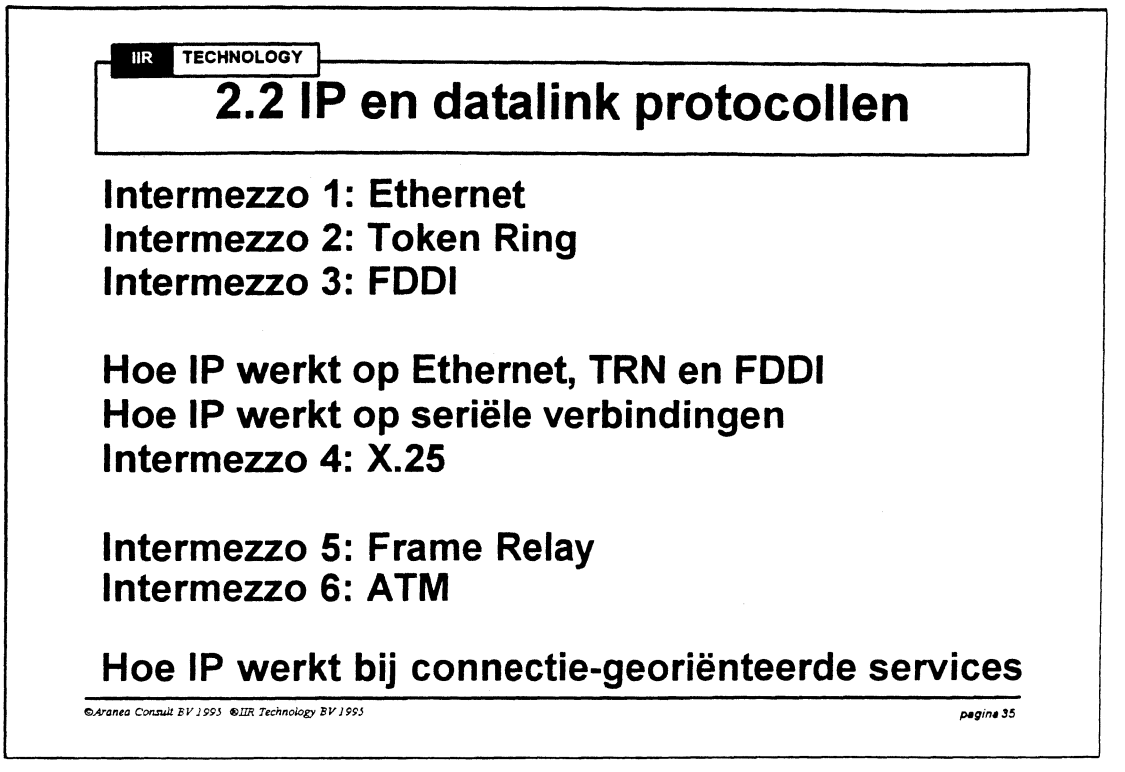
- **TCP/IP is gedefinieerd vanaf laag 3**
- **Dit betekent dat TCP/IP zich niet bekommert om lagen 1 en 2: 'die zijn er al genoeg'**
- **In separate RFC's is beschreven hoe TCP/IP kan werken over laag 2 protocollen**
- **Bekendste:**
  - **TCP/IP over Ethernet, Token Ring, FDDI**
  - **TCP/IP over LLC**
  - **TCP/IP over X.25**
  - **TCP/IP over SLIP/PPP**
  - **etcetera**

TCP/IP is ontwikkeld, onafhankelijk van de onderliggende datalink- en fysieke laag. Dit betekent dat er diverse interfaces zijn ontwikkeld om TCP/IP op voorkomende onderliggende lagen te kunnen laten werken. Het meest gebruikt is ongetwijfeld DIX Ethernet (Ethernet versie II). Dit zijn veelal rechttoe, rechtaan implementaties. Er zijn echter ook wat minder voor de hand liggende implementaties ontwikkeld, zoals TCP/IP over X.25. Hier wordt immers een connection less best effort delivery system over een connection oriented packet switched netwerk gebruikt!



odi wordt min of meer vereist door Novell
Oplossing naar TCP/IP:





Om TCP/IP te kunnen beheren, is ook grondige kennis van het gebruikte medium of de gebruikte toegangsmethode nodig.

Daarom zal nu eerst ingegaan worden op de globale werking van de bekendste toegangsmethoden waarop IP gebruikt kan worden.

IIR TECHNOLOGY

Intermezzo 1: Ethernet

- **1976 Experimental Ethernet:**
  Xerox PARC
  3 Mbps, coax kabel
- **Digital, Intel, Xerox (DIX)**
  Ethernet I, II (blue book)
  10 Mbps, coax kabel, glasvezel
- **IEEE 802.3, ISO 8802.3**
  breedband, twisted pair kabel (UTP CAT 3 en CAT 5)

Ethernet, de LAN techniek met verreweg de grootste *installed base*, werd in de eerste helft van de jaren '70 ontwikkeld in het Xerox Palo Alto Research Centre (PARC). Daar werd, lang voor de PC op de tekentafel stond, gewerkt met 'persoonlijke' minicomputers. Voor de onderlinge communicatie zocht men naar een snel en betrouwbaar medium als alternatief voor een stervormig netwerk van seriële verbindingen. Electronica was in die dagen duur en niet zo betrouwbaar als nu. Daarom werd gekozen voor een passief medium: een coaxkabel, die langs alle stations liep. In 1976 werd dit 'experimentele' Ethernet beschreven in een wetenschappelijke publicatie. Enkele jaren later staken Digital, Intel en Xerox (DIX) de koppen bij elkaar om een standaard op te stellen op basis waarvan de deelnemers vervolgens produkten zouden kunnen ontwikkelen. Ethernet I kreeg een signaleringssnelheid van 10 Mbps (i.p.v. het experimentele Ethernet dat op 3 Mbps draaide). Kort daarna werden in de Ethernet II standaard (het Blue Book) de laatste puntjes op de *i* gezet. Naast coaxkabel werd ook glasvezel als verbinding tussen twee repeaters beschreven.

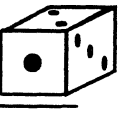
Binnen de IEEE werden publieke standaarden voor LANs besproken. Op basis van de DIX standaard, maar met verfijningen (en een andere terminologie) werd daar de 802.3 CSMA/CD Medium Access Control standaard vastgelegd. In de praktijk is het verschil tussen Ethernet II en 802.3 (of de gelijkwaardige ISO 8802.3 standaard) niet belangrijk voor de hardware; er is wel verschil in de manier waarop informatie wordt ingepakt, maar beide standaarden kunnen naast elkaar (op dezelfde kabel en door dezelfde stations) gebruikt worden. Inmiddels zijn deze standaarden uitgebreid met nieuwe media: breedband (FDM) en twisted pair (de binnen enkele jaren immens populair geworden 10BaseT standaard).

IIR TECHNOLOGY

I1: Ethernet principe

- **Broadcast netwerk**
  bus topologie

- **CSMA/CD**
  Carrier Sense : Luisteren of het stil is
  Multiple Access: Spreken als het stil is
  Collision Detection: Stop met praten als er iemand anders doorheen praat
- **Binary backoff**
  wacht 'willekeurige' tijd met nieuwe poging



Ethernet is (net als Token Ring, FDDI en alle andere LAN technieken) een *broadcast* netwerk; dat wil zeggen dat een uitgezonden frame fysiek bij alle stations langskomt. Elk frame draagt een bestemmingsadres en alleen het geadresseerde station of, in het geval van een multi-cast bericht, de geadresseerde stations lezen het bericht. Dat principe geeft een grote flexibiliteit, maar levert wel een beveiligingsprobleem op en geeft ook veel verkeer en een beperking in aantal aangesloten stations en reikwijdte.

Oorspronkelijk had Ethernet een bus-topologie, al zijn veel moderne implementaties grotendeels stervormig (met behulp van multi-poort repeaters) opgebouwd.

Het Carrier Sense Multiple Access with Collision Detection (CSMA/CD) toegangsprotocol is bekend: het werkt ongeveer zoals tussen mensen op een verjaarspartij. Wie wil spreken, wacht eerst tot het stil is en steekt dan van wal. Blijkt iemand anders tegelijkertijd het woord genomen te hebben, dan stop je beleeft.

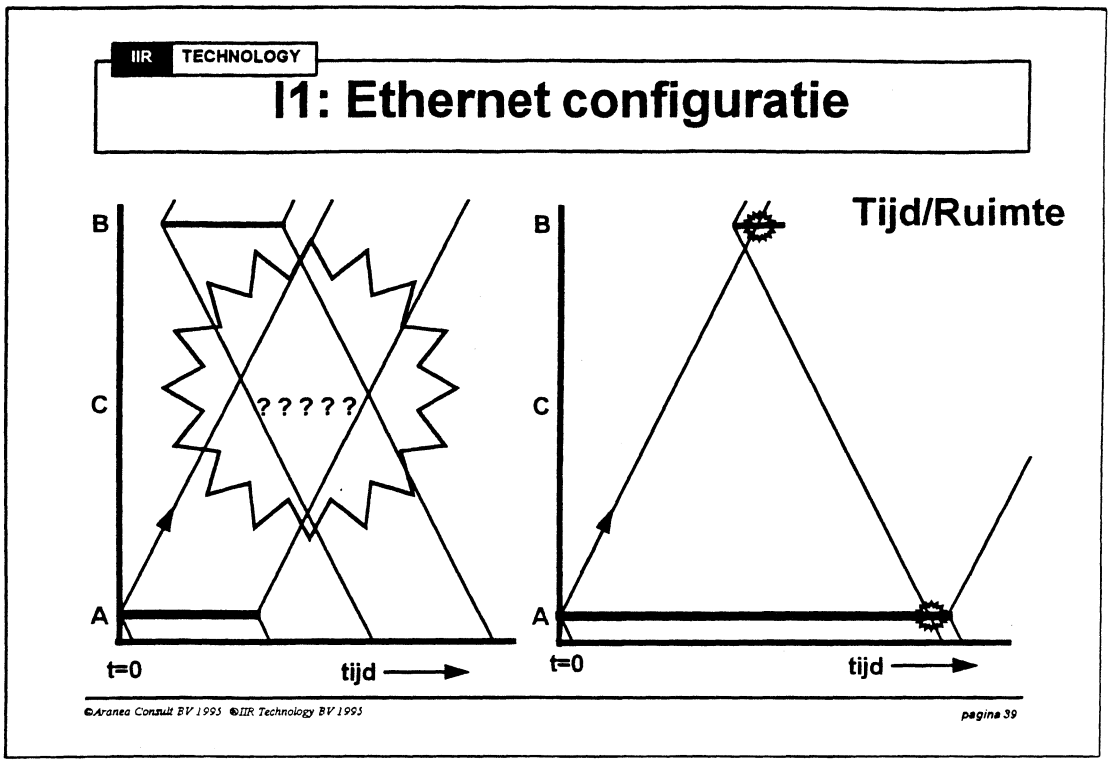
Hier houdt de vergelijking tussen Ethernet en de verjaarsvisite op: mensen kunnen door 'bodylanguage' signaleren wie de voorrang krijgt of neemt, en hanteren een zekere 'pikorde' om de prioriteit van sprekers te bepalen. Netwerkstations hebben buiten het netwerk geen communicatiekanaal. Om de impasse te doorbreken gebruikt Ethernet een statistische benadering: elk station gooit na een collision een electronische munt en bepaalt aan de hand van de worp hoe lang hij wacht met een nieuwe transmissiepoging. Meestal zal één van de twee stations als eerste een nieuwe poging doen. Als de kabel op dat moment vrij is, begint het station zijn uitzending. Het andere station zal kort daarna zijn hertransmissie willen beginnen, maar dan merken dat de kabel al bezet is en wachten tot dat bericht voorbij is.

IIR TECHNOLOGY

I1: Ethernet Binary Backoff

- **Na elke collision wordt het interval verdubbeld: 2, 4, 8, ... 512, 1024**
- **Totaal 16 pogingen**
- **Collision is niet het probleem, het is het arbitrage mechanisme van Ethernet**

Het is natuurlijk mogelijk dat beide stations dezelfde worp met de dobbelsteen doen en dus opnieuw botsen. Ook kan het zijn dat er nog meer stations trachten te zenden. Na een tweede botsing wordt opnieuw een willekeurig interval gekozen, maar nu met een 'dobbelsteen' die twee maal zoveel kantjes heeft. Treedt er telkens opnieuw een botsing op, dan wordt een willekeurige keuze uit 8, 16, 32, ... 1024 intervallen gekozen. Hoe groter het aantal intervallen, hoe kleiner de kans dat er twee stations toevallig hetzelfde interval kiezen. Zelfs als er een groot aantal stations (maximaal 1024) tegelijk proberen te zenden, zal er al snel één station met het kortste interval uit de bus komen, die dan zijn bericht met succes kan versturen. Daarna gaat de competitie verder tussen de overige stations. Totaal doet een station 16 pogingen, de laatste zes telkens met een keuze uit 1024 intervallen. Zowel de theorie als praktijk bewijzen dat met dit eenvoudige, volledig gedistribueerde *binary backoff* algoritme een betrouwbaar en stabiel netwerk wordt verkregen.



Ethernet (CSMA/CD) bevat een intrinsieke relatie tussen de grootste afstand in het netwerk, de signaleringssnelheid (bitrate) en de lengte van het kortste bericht. Dat is aan de hand van deze tijd-ruimte diagrammen eenvoudig te zien.

In het diagram staat vertikaal de afstand langs de kabel (voor het gemak doen we alsof het netwerk uit een enkele kabel bestaat). Langs de kabel bevinden zich drie stations: A, B en daartussen C. De horizontale as stelt de tijd voor. Het linker diagram laat zien wat er mis gaat als de relatie tussen afstand en minimale berichtlengte niet goed is.

Op tijdstip 0 begint A een frame uit te zenden. Het eerste bit loopt langs de kabel (schuine lijnen naar boven en onder), passeert C en bereikt na enige tijd station B. Maar nog vóór B het begin van het frame van A heeft gezien, start hij met zenden: dat mag B, want hij ziet immers geen signaal op de kabel. B's frame is zo kort, dat hij klaar is met zenden voordat de kop van A's frame bij hem arriveert. B denkt dus dat alles goed is gegaan: hij heeft eerst een frame verstuurd en korte tijd daarna een frame van A langs zien komen. Ook voor A lijkt alles in orde. Helaas, beide frames waren bestemd voor station C, en die heeft ze tegelijkertijd (en dus niet) gekregen!

In de rechter figuur is de situatie gecorrigeerd. A zendt een frame dat voldoet aan de eis dat het minstens zo lang moet zijn als de *roundtrip* tijd van het netwerk. Als B op het laatst mogelijke moment voor het arriveren van het frame van A begint te zenden, ontdekt A de collision nog voor dat hij klaar is met zenden. Het *binary backoff* mechanisme zorgt nu voor een snelle hertransmissie, veel sneller dan wanneer via een protocol *timeout* zou worden ontdekt dat het frame verloren is gegaan.

IIR TECHNOLOGY

I1: Ethernet karakteristieken

- **Round trip delay: max 51,2 microseconde**
- **Minimum packet size: 64 bytes**
- **Maximum packet size: 1518 bytes**
- **Long term average load: < 50 %**
- **Peak load: 97%**
- **Overhead: 26 bytes/packet (preamble, header en checksum)**

Enkele karakteristieken van Ethernet:

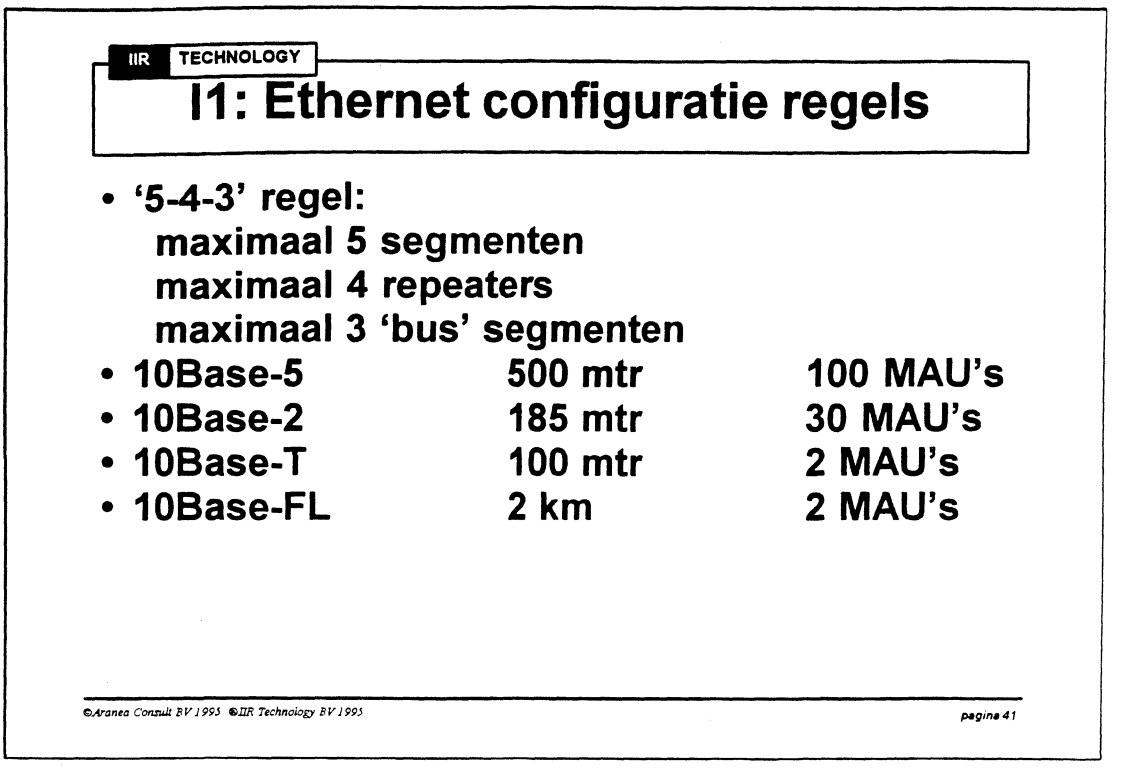
Zoals gemeld, moet er een minimum frame lengte zijn om collisions te kunnen detecteren tijdens het verzenden van frames. Er is gekozen voor een minimum lengte van 64 bytes (512 bits). Om deze 512 bits te verzenden, zijn 512 bittijden nodig. Bij een transportsnelheid van 10 Mbps gebeurt dat in 51,2 microseconden.

Hiermee is de maximale omvang van een collision domain bepaald: 51,2 microseconden. Voor de goede werking wordt daarom aanbevolen om de 'round-trip delay', de tijd die een signaal nodig heeft om heen en terug over het segment te gaan, kleiner dan 51,2 microseconde te laten zijn.

De maximale frame grootte is 1518 bytes, exclusief de preamble die nodig is voor bit-synchronisatie. De preamble is minimaal 64 bits groot.

Over Ethernet belasting worden altijd de meest vreemde verhalen verteld. Het is mogelijk om Ethernetten tot bijna 100% te belasten zonder dat het netwerk 'down' gaat: de wachttijd neemt dan wel toe. Algemeen kan gezegd worden dat in een collision domain de gemiddelde belasting over langere periode niet boven de 40% uit moet komen, anders is er niet voldoende bandbreedte over om de pieken op te vangen. Bovendien wordt anders de wachttijd te groot.

Inclusief de preamble heeft Ethernet een overhead van 26 bytes/frame. Bij een gemiddelde frame grootte van 200 bytes dus 13%.

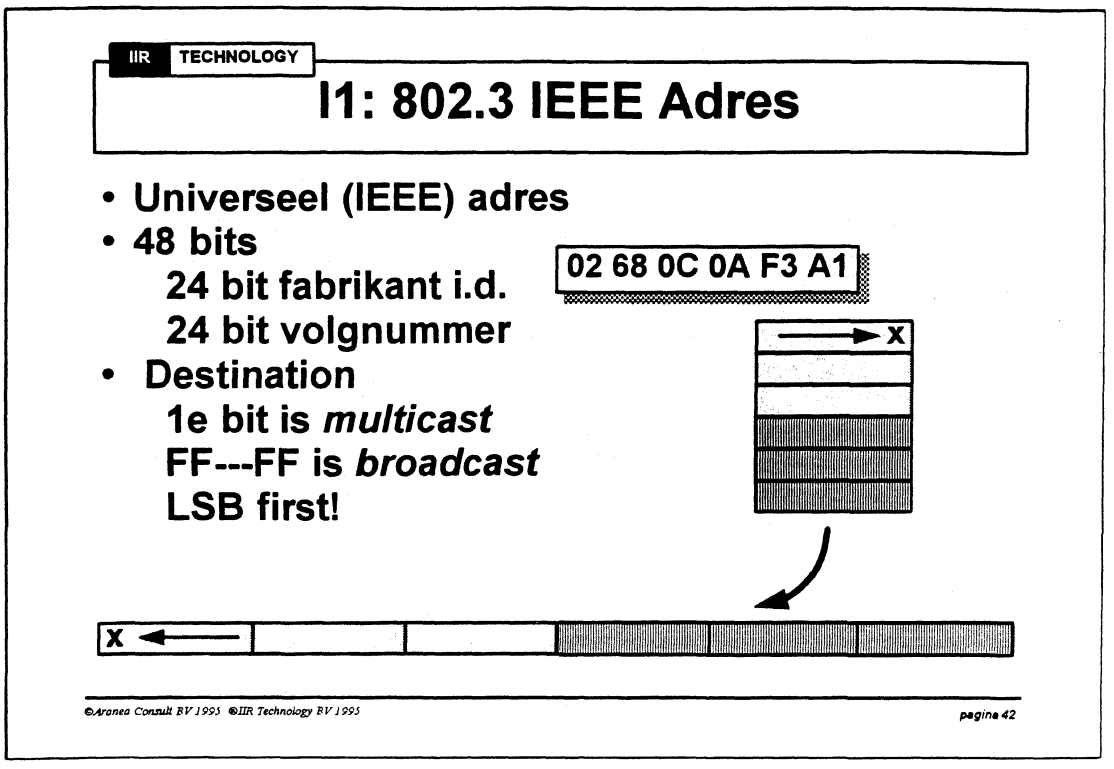


Voor een juiste werking van het CSMA/CD principe moet het Ethernet goed geconfigureerd zijn. Dit is altijd te bepalen door de round-trip delay te meten, maar er is een simpele vuistregel voor het configureren van een collision-domain:

Zorg ervoor dat het communicatie pad tussen de twee verst verwijderde stations uit maximaal 5 segmenten bestaat, gekoppeld met maximaal 4 repeaters en dat er op maximaal 3 segmenten, behalve de repeaters, ook andere apparatuur is aangesloten.

Zorg er verder voor dat de maximale lengtes en het maximaal aantal stations op een segment, niet groter is dan volgens de standaard is toegelaten.

In de praktijk blijkt het overgrote deel van de performance en andere problemen met Ethernet veroorzaakt te worden door een niet juist geconfigureerd collision domain.



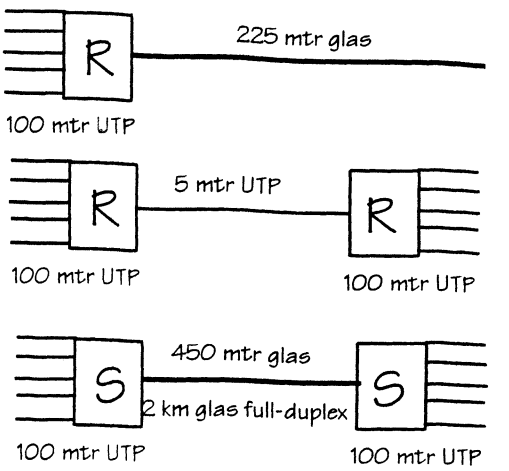
Ethernet gebruikt universele adressen: er zijn geen twee Ethernet interfaces op de wereld die hetzelfde adres hebben. Dat wordt bereikt door elke fabrikant een eigen pre-fix te geven: de eerste drie van de 6 adresbytes geven aan wie de fabrikant is. De laatste 3 bytes geven dan het serienummer van de kaart aan. Heeft een fabrikant 2^{24} interfaces gemaakt, dan kan hij van de IEEE een nieuwe prefix krijgen. Het gevolg is dat je bij het installeren van Ethernet kaarten nooit rekening hoeft te houden met het adres.

Het eerste uitgezonden bit van het bestemmingsadres (dat is het minst-significante bit van het eerste byte) bepaalt of het een multi-cast adres is. Daarmee kan een logische groep stations (bv 'alle fileservers') worden aangesproken. Er is één multicast groep die alle stations omvat: als het bestemmingsadres uitsluitend '1'-en bevat (FF FF FF FF FF FF hex), worden alle stations aangesproken: dit heet een broadcast bericht.

IIR TECHNOLOGY

I1: 100Base-T

- **Andere configuratie regels**
- **100Base-TX**
- **100Base-T4**
- **100Base-FX**
- **Repeaters**
- **Switches**



pagina 43

Voor 10 Mb Ethernet geldt de 5-4-3 regel: tussen twee stations mogen maximaal 5 segmenten, 4 repeaters en 3 bus-segmenten zitten. Regels die er onder meer voor zorgen dat een collision altijd ontdekt wordt vóór het einde van het frame. Vandaar dat een Ethernet frame (pakketje) een minimale afmeting (64 bytes = 51,2 microseconde) moet hebben. Wat verandert er nu bij de overgang van 10 naar 100 Mb? Aan het collision mechanisme of de minimale framelengte wordt niet getornd. Maar een 64 byte frame duurt bij 100 Mb maar 5,12 microseconde! Dus moet de afstand tussen twee stations ook (ongeveer) een factor 10 kleiner worden. Feitelijk wordt de maximale afstand bepaald door de looptijd van het signaal (ongeveer 200 meter per microseconde, afhankelijk van het gebruikte kabeltype), de vertraging in transceivers en repeaters en de collision detectie tijd. Maar ook de transmissie-technische problemen zijn bij 100 Mb groter en leggen extra beperkingen op. Dat levert de volgende regels op:

- 1. maximaal 100 meter voor een twisted pair link (100BaseTX of 100BaseT4).
- 2. maximaal 450 meter voor een half-duplex glasvezel link (100BaseFX) zonder repeaters.
- 3. 1 repeater met maximaal 100 meter twisted pair plus 225 meter glasvezel.
- 4. maximaal 2 repeaters met 100+5+100 meter twisted pair.
- 5. maximaal 2 km voor een full-duplex glasvezel link.

IIR TECHNOLOGY

I1: 100VG-Anylan

- **Nieuw protocol voor fysieke laag**
- **'Demand Priority Multiple Access'**
- **'Encapsulation' van Ethernet of TRN frames**
- **Aparte bandbreedte reservering voor isochrone diensten**
- **Geen collisions**

100VG-Anylan is de IBM/HP tegenhanger van 100Base-T. Ooit is 100VG-Anylan ontwikkeld als moderne Ethernet versie. Omdat echter van een geheel ander toegangsmechanisme dan CSMA/CD wordt uitgegaan, heeft 100VG-Anlylan niets meer met Ethernet te maken. Vandaar dat hiervoor een nieuwe IEEE standaard is bedacht: 802.12.

100VG-Anylan gebruikt een MAC-onafhankelijke methode voor toekenning van de gewenste bandbreedte. Hierdoor kunnen zowel Ethernet frames als Token-Ring frames getransporteerd worden over een 100VG-Anylan netwerk. Omdat er een demand-priority algoritme gebruikt wordt, is het ook mogelijk om bij 100VG-Anylan bandbreedte te reserveren voor kritische toepassingen zoals video of audio signalen.

Het ziet er op dit moment niet naar uit dat 100VG-Anylan door zal breken. IBM laat al weten eerder met 100Base-T produkten op de markt te komen dan 100VG-Anylan, zodat alleen HP nog overblijft als 100VG-Anylan promotor.

IIR TECHNOLOGY

Intermezzo 2: Token Passing Ring

- **IBM Token Ring Network**
- **IEEE 802.5**
- **ISO 8802.5**
  Token Ring Access Method and Physical Layer Specification

In de 70-er jaren werden er op tal van plaatsen experimenten met Local Area Netwerk technieken gedaan. Eén van de richtingen waarin men zocht, was de netwerken waarbij de stations in een ring geschakeld zijn en de informatie langs de ring van station naar station loopt. Om de toegang tot de ring te regelen wordt 'de beurt' van station naar station doorgegeven, en wel in de vorm van een speciaal signaal dat meestal het *token* heet. Van de vele ontwerpen die op dit principe gebaseerd waren, is alleen Token Ring Network als belangrijk produkt overgebleven (met FDDI als afstammeling).

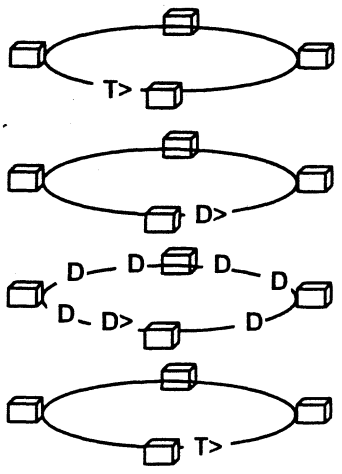
Token Ring Network werd door IBM geïntroduceerd en wordt tot op de dag van vandaag door IBM gedomineerd. Het werd geïntroduceerd als tegenhanger van Ethernet, dat destijds in de IBM literatuur als hoogst onbetrouwbaar werd afgeschilderd (de als treintjes getekende netwerkframes knalden tijdens *collisions* op elkaar, waarbij de brokstukken *corporate data* je om de oren vlogen. Wilt u uw bedrijfsgegevens aan zo'n netwerk toevertrouwen?).

IBM bracht Token Ring in bij de standaardisatiecommissies van de IEEE en zo ontstond IEEE 802.5 als één van de alternatieve MAC standaarden.

IIR TECHNOLOGY

I2: Token Ring principe

- **Broadcast netwerk**
  ring topologie
- **Token Passing:**
  wacht op het 'token'
  zend je bericht
  ontvang het bericht
  zend het token door
- **Voor alles is een regel**

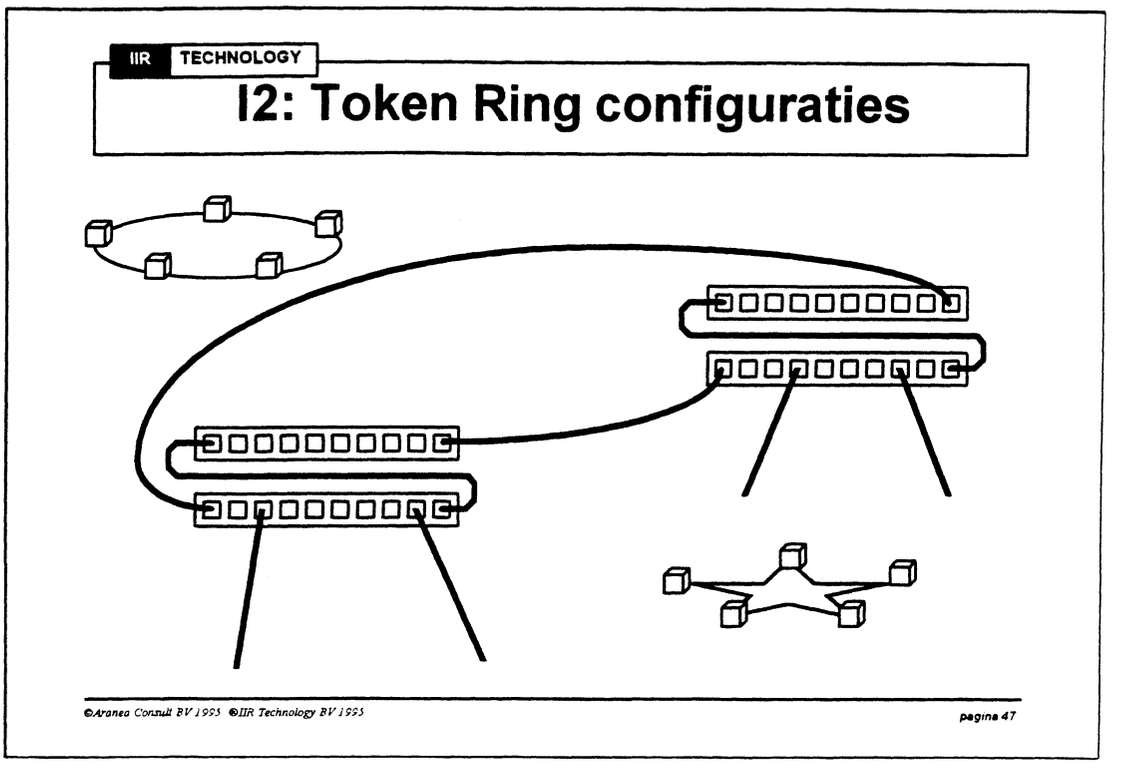


pagina 46

Token Ring is, net als Ethernet, een broadcast netwerk, maar nu met een ring structuur. Als er geen verkeer over de ring gaat circuleert er een kort frame: het token. Als een station het token ontvangt en niets te versturen heeft, geeft het het token onmiddellijk door aan zijn buurman.

Heeft een station wel een frame te verzenden, dan moet het wachten tot het token langs komt. Het token wordt van de ring gehaald en het frame verstuurt. Het frame doorloopt de ring, waarbij elk station het in de header opgenomen bestemmingsadres bekijkt: herkent een station zijn adres (of één van zijn multicast adressen), dan kopieert het de inhoud van het frame. Intussen vervolgt het frame zijn weg langs de ring, totdat het bij de oorspronkelijke zender terug is. Deze neemt het frame van de ring en zet tenslotte het token weer op de ring.

Token Ring is vanuit een totaal andere filosofie ontwikkeld als Ethernet: waar Ethernet probeert met een minimum aan complexiteit en regels te volstaan, is Token Ring gebaseerd op een boek vol regels en afspraken. Elke denkbare uitzonderingssituatie is beschreven en in het MAC-layer protocol vastgelegd. Zo kent een Token Ring netwerk een aantal standaard functionele adressen (Ring Monitor, Ring Error Monitor, etc) en sturen Token Ring interfaces onder verschillende omstandigheden dienstberichten aan deze adressen.



De feitelijke implementatie van een Token Ring netwerk heeft toch weer een ster-structuur. Vanuit de kabelkast worden kabels naar alle werkplekken getrokken, waarop telkens één station wordt aangesloten. In de kabelkast worden Multi-station Access Units (MAUs) geplaatst, waar de stationskabels (*lobe*-kabels) op worden aangesloten. De electrische ring loopt via de lobekabel naar een station en weer terug, door de MAU naar de volgende poort en zo voort, van MAU naar MAU. De (passieve) MAUs bevatten een relais voor elke poort dat door het aangesloten station kan worden omgeschakeld; op die manier kan een station zich in de ring schakelen of juist weer losmaken, zonder de ring te verbreken.

De eerste versie van Token Ring had een signaleringssnelheid van 4 Mbps en gebruikte de hier beschreven passieve MAUs. Naar keuze kon afgeschermde (STP) of onafgeschermde twisted pair (UTP) kabel worden gebruikt. In een STP ring konden maximaal 260 stations worden opgenomen, maar een UTP ring was tot 72 stations beperkt. Dat had te maken met het optreden van *jitter*. Een Token Ring is eigenlijk een aaneenschakeling van repeaters (elk station fungeert als repeater). Elke link voegt een klein beetje jitter toe, wat minder voor STP en wat meer voor UTP kabel. Na een zeker aantal stations is de geaccumuleerde jitter zo groot, dat de ontvangst onbetrouwbaar wordt.

Later heeft IBM een 16 Mbps versie van Token Ring uitgebracht en daarvoor een actieve MAU ontwikkeld, de z.g. Controlled Access Unit ofwel CAU. Aanvankelijk werd 16 Mbps (dat gevoeliger is voor jitter) alleen over STP kabel ondersteund, maar onder druk van de markt heeft IBM in samenwerking met Synoptics een speciale schakeling ontwikkeld, waarmee de jitter ook in UTP ringen in de hand kan worden gehouden.

IIR TECHNOLOGY

I2: Token Ring karakteristieken

- **Minimum packet size: 21 bytes**
- **Maximum packet size: 10 ms (~4K/16K)**
- **Long term average load: < 50 %**
- **Peak load: 100%**
- **Overhead: 21 bytes/packet**
- **Bandwidth consumption: MAC frames**
- **Jitter gevoelig**

Enkele karakteristieken van Token Ring:

In tegenstelling tot Ethernet is er geen minimum frame grootte. Het kleinste frame dat gebruikt wordt in Token Ring, is het Token zelf; hiervan is de frame grootte drie bytes (het Token moet volledig op de fysieke ring passen, hiermee is dus de fysieke ring omvang bepaald).

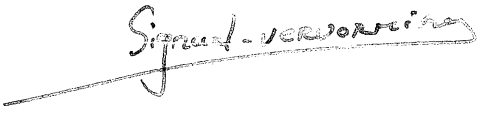
Omdat Token Ring in tegenstelling tot Ethernet een synchroon netwerk is (er gaat immers altijd een Token rond), is er geen bitsynchronisatie nodig en dus ook geen preamble.

In Token Ring is afgesproken dat een station het Token nooit langer dan 10 ms vast mag houden. Dat levert een maximale framegrootte van 4.5k voor 4 Mb en ruim 17k voor 16 Mb ringen.

Ook kan TR tot 100% belast worden, waarbij de wachttijd overigens nooit een bepaalde limiet kan overschrijden (aantal stations * 10 ms). In een maximaal Token Ring netwerk (260 stations) kan het dus theoretisch ruim 2,5 seconde duren voordat het voorste frame in de transmit-queue van een station verstuurd is.

Vanwege het ontbreken van een preamble is er minder frame overhead in TRN dan in Ethernet, slechts 21 bytes. Wordt er echter rekening gehouden met de frames die uitsluitend dienen voor de besturing van de ring, dan is de overhead veel groter. Ook al vindt er geheel geen data-uitwisseling tussen twee stations plaats, dan gaan er al heel veel frames over de ring.

Bij TR is het optreden van jitter een probleem. Er moeten speciale voorzieningen getroffen worden om de jitter zo klein mogelijk te houden.



IIR TECHNOLOGY

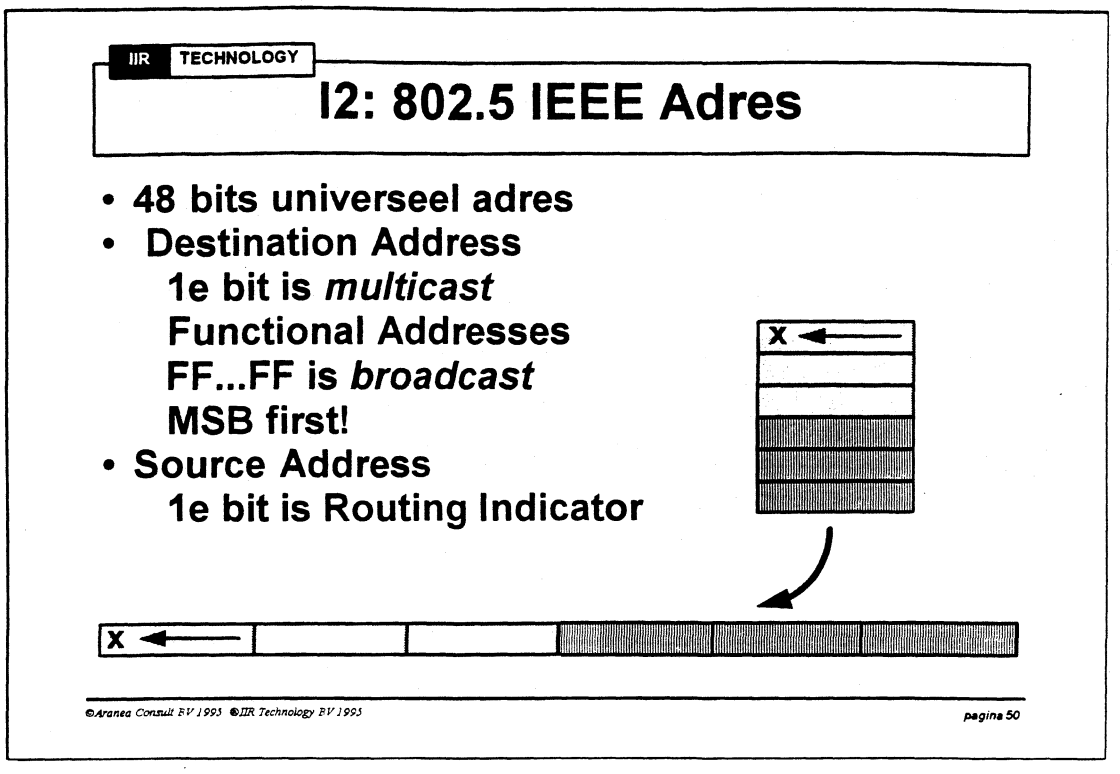
I2: Token Ring configuratie regels

- **Afhankelijk van soort bekabeling**
- **Afhankelijk van aantal patchkasten**
- **Via tabellen van IBM**
- **Omvang in aantal aangesloten stations wordt bepaald door optredende jitter**
- **Omvang in kabellengte wordt bepaald door gebruikte kabel karakteristieken**

Bij TR netwerken is er geen eenvoudige configuratieregel zoals bij Ethernet. IBM heeft allerlei tabellen waarin vermeld wordt hoeveel stations er aangesloten mogen worden bij een gegeven aantal MAU's, patchpanelen en een gegeven ring lengte.

Dat betekent in de praktijk dat eerst het aantal MAU's bepaald moet worden, dan het aantal pacthpanelen, en vervolgens moet de ringlengte (Adjusted Ring Length) uitgerekend worden.

Afhankelijk van het type kabel dat gebruikt wordt, kan daarmee in de tabellen worden opgezocht of de configuratie voldoet, of dat er additionele repeaters of converters gebruikt moeten worden.



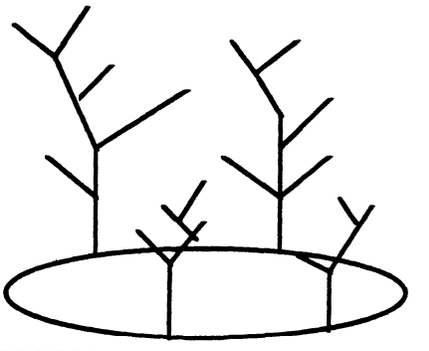
De Token Ring adressen zijn op dezelfde manier opgebouwd als bij Ethernet. Het enige verschil is, dat de bits van een byte in de omgekeerde volgorde worden uitgezonden: het MSB wordt het eerst verzonden. We vinden het multicast-bit dus als het meest significante bit van het eerste byte van het bestemmingsadres. Om praktische redenen moet het multicast bit immers altijd het eerste verzonden bit zijn, zodat de ontvanger direct weet hoe hij de rest van het adres moet interpreteren.

De afzender kan natuurlijk nooit een multi-cast adres zijn. Dit bit wordt in Token Ring dan ook voor iets anders gebruikt: het wordt gezet om aan te geven dat achter het afzenderadres een Routing Information (RI) veld is opgenomen. Dit veld wordt gebruikt om in netwerken die uit meerdere ringen bestaan, de route van afzender naar bestemming aan te geven. Het RI veld bevat dan een rij ring- en bridge-nummers, ongeveer zoals de adreslabels van KLM of SABENA vluchtnummers en luchthavens van de route van uw bagage aangeven.

IIR TECHNOLOGY

Intermezzo 3: FDDI

- Fiber Distributed Data Interface
- ANSI X3T9.5 standaard
- Token Passing access methode
- 100 Mbps
- Media:
  - multimode glasvezel
  - singlemode glasvezel
  - twisted pair kabel
- Ring van bomen



Fiber Distributed Data Interface (FDDI) is een door ANSI ontwikkelde standaard waarmee zowel LAN, MAN als WAN toepassingen zijn te realiseren. FDDI gebruikt een vorm van token passing toegangsregeling, maar is niet direct compatibel met Token Ring. De signaleringssnelheid is 100 Mbps. Als medium kan multi mode glasvezel (tot 2 km), single mode glasvezel (40-60 km) en nu ook afgeschermde of unshielded twisted pair gebruikt worden. In het laatste geval spreken we van CDDI (Copper Distributed Data Interface) of TP-PHY (Twisted Pair PHYsical layer).

De topologie van een FDDI netwerk is een ring van bomen: op de hoofdring kunnen *concentrators* worden aangesloten met een boomstructuur van stations. De ring (die net als bij TRN vaak als stervormige bekabeling wordt opgezet), kan bv. de gebouwen op een campus, bedrijfsterrein of industriegebied verbinden, terwijl de bomen de stations binnen de gebouwen verbinden. Bij totale uitval van een gebouw (stroomstoring, brand, etc), blijft de ring intact.

IIR TECHNOLOGY

I3: FDDI

- **4B/5B codering: 100 Mbps, 125 MBaud**
- **Maximaal 500 stations, 2 km tussen twee stations over multi-mode vezel, totale omtrek 100 km**
- **Standaard bevat 4 delen:**
  PMD: Physical Medium Dependent
  PHY: Physical Sublayer
  MAC: Medium Access Control
  SMT: Station Management

FDDI gebruikt een 4Bit/5Bit codering, zodat de signaleringssnelheid op de kabel 125 MBaud is. In een net mogen maximaal 500 stations worden opgenomen met (op multimode glasvezel) een maximale onderlinge afstand van 2 km. De totale ring omvang wordt, enigszins willekeurig, op 100 km begrensd, maar er zijn netwerken met een aanzienlijk grotere omtrek. Daarmee gaat wel een stukje capaciteit verloren.

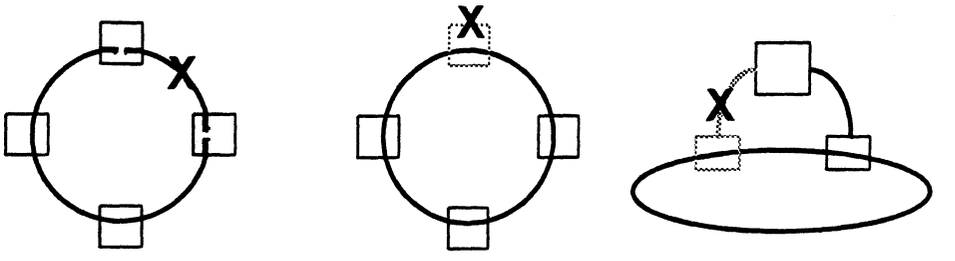
De FDDI standaard valt in 4 delen uiteen:

- Physical Medium Dependent (PMD) bevat alle zaken die direct afhankelijk zijn van het gebruikte fysieke medium (bv. stekkers)
- PHYsical sublayer (PHY) beschrijft de signalering en codering
- Medium Access Control (MAC) definieert het token passing mechanisme
- Station ManagemenT (SMT) legt de parameters van de overige lagen vast en hoe die via het netwerk kunnen worden beheerd

IIR TECHNOLOGY

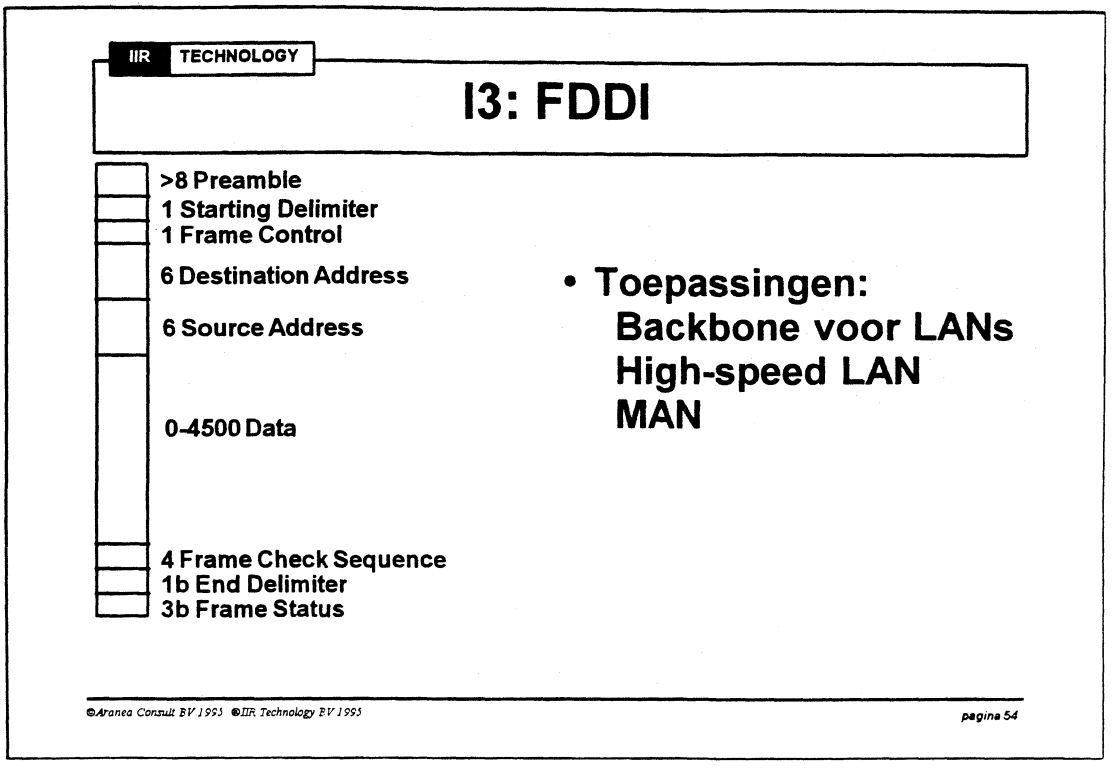
I3: FDDI

- **Fail-safe mechanismen:**
  Electronische 'wrap around'
  Opto-mechanisch 'bypass' relais
  Dual-homing



FDDI bevat verschillende mechanismen die de beschikbaarheid van het netwerk verbeteren.

- De ring is dubbel uitgevoerd, zodat bij uitval van een segment de aangrenzende stations in *wrap-mode* kunnen gaan en het verkeer via de backup ring rond kunnen sturen.
- Stations kunnen voorzien worden van een electro-mechanische *bypass* schakelaar, die bij uitval van het station de inkomende en uitgaande glasvezel aan elkaar koppelt.
- Stations kunnen met een dubbele aansluiting op twee concentrators worden gekoppeld (*dual-homing)*, wat hen beschermt tegen uitval van een kabelsegment of een concentrator.



De FDDI frames lijken op de Token Ring frames, maar bevatten geen Routing Information. Ook FDDI gebruikt standaard 48-bit IEEE adressen. Elk FDDI station bevat een elastisch buffer en filtert daarmee alle jitter uit het ontvangen signaal. Daardoor - en door het gebruik van een preamble voor ieder frame - kan een FDDI ring meer stations bevatten dan een Token Ring netwerk.

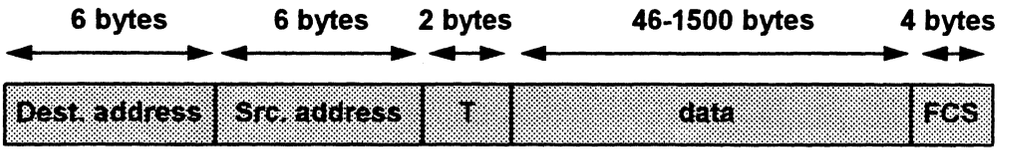
De belangrijkste toepassingen voor FDDI zijn:

- Backbone netwerk voor 'klassieke' LANs. Wie enige tientallen LANs (Ethernet en/of Token Ring) onderling wil koppelen, heeft daarvoor een medium nodig dat ten minste een ordegrootte sneller is dan de LANs zelf; immers, als elk LAN 10% interlokaal verkeer genereert, leveren 10 LANs potentieel al genoeg verkeer om een standaard backbone te verzadigen.
- High-speed LAN. Voor sommige toepassingen zijn Ethernet en Token Ring eenvoudig niet snel genoeg, zelfs als het om een klein aantal stations gaat. Een voorbeeld is een netwerkje van snelle grafische werkstations, die gebruikt worden om seismografische beelden te bewerken en te bekijken. De beelden zijn 3 Mbyte per plaatje. Zelfs op een privé Ethernet duurt het al gauw 6 seconden om een plaatje van de server naar het werkstation te sturen. Dus als je wilt kunnen bladeren...
- Metropolitan Area Network. Een FDDI ring kan voldoende groot worden om een industrieterrein en stadsdeel te bestrijken. Door een aantal ringen aan elkaar te koppelen, kan een netwerk worden gebouwd dat de hele Randstad of de Brusselse agglomeratie bestrijkt.

IIR TECHNOLOGY

2.2 IP bij LAN technieken

- IP datagrammen kunnen 'rechtstreeks' op DIX Ethernet (Ethernet II) worden 'ingepakt', beschreven in de Host Requirements RFC
- Encapsulation beschreven in RFC 894
- Het IP protocol heeft type aanduiding 0x800



Het meest gebruikt is ongetwijfeld TCP/IP over Ethernet. Uiteraard vanwege de grote penetratiegraad van TCP/IP in de Unix wereld. Het inpakken van TCP/IP in Ethernet is vrij rechttoe, rechtaan. Ethernet kan pakketten van 1500 bytes vervoeren (MTU=1500). IP datagrammen die groter zijn, zullen derhalve gefragmenteerd moeten worden. Ethernet doet aan multiplexing (denk aan multiple protocol stacks op PC's) aan de hand van het 'Type'-veld. Als in een Ethernet frame een IP datagram zit, wordt dit aangegeven met type 0x0800.

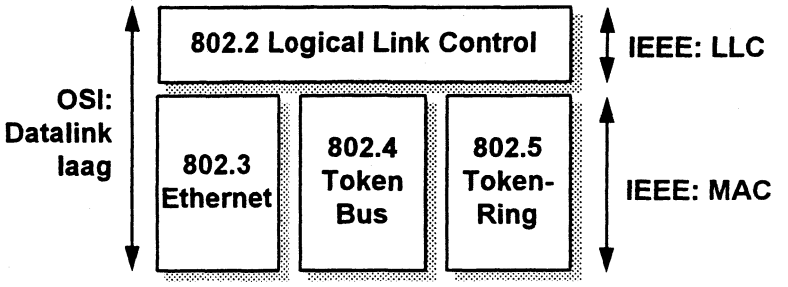
Let op! Er zijn twee soorten Ethernet, het 'oorspronkelijke' Ethernet (DIX Ethernet, ontwikkeld door Digital, Intel en Xerox) en de later door het IEEE committee gestandaardiseerde IEEE 802.3 Ethernet. De verschillen tussen de beide Ethernet varianten zijn niet substantieel, maar wel subtiel genoeg om verschillend te zijn! Daar waar Ethernet spreekt over een 'Type'-veld, gebruikt IEEE 802.3 een 'length'-veld. De structuur van de frames (en ook de daadwerkelijke fysieke codering en dergelijke) is volledig identiek; het verschil zit puur in de interpretatie van het 13e en 14e byte!

NB. Om volledig te zijn zou in bovenstaande figuur ook nog een 64 bits *preamble* moeten worden opgenomen. Dit draagt echter niet bij aan de functionaliteit van het Ethernet protocol en is derhalve achterwege gelaten.

IIR TECHNOLOGY

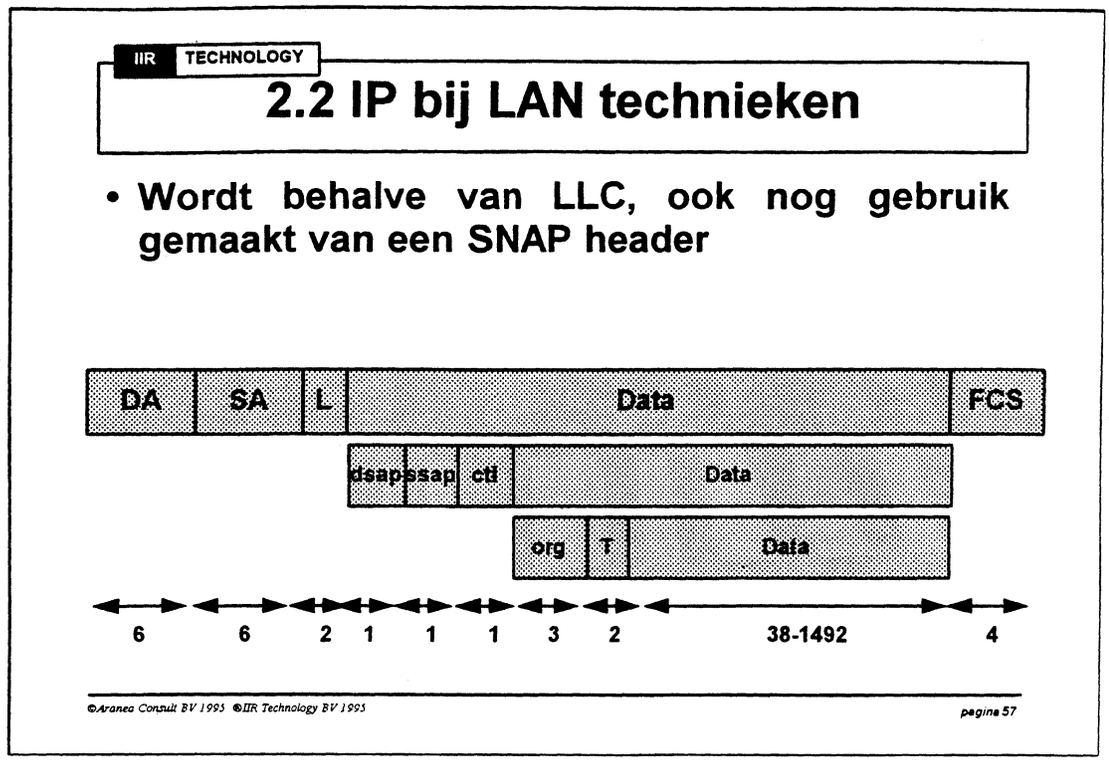
2.2 IP bij LAN technieken

- **IP kan op 802.x netwerken gebruikt worden**
- **Niet beschreven in Host Requirements RFC**
- **Maakt gebruik van 802.2 Logical Link Control, beschreven in RFC 1042:**



De IEEE 802 commissie, de commissie die zich bezighoudt met LAN standaarden, heeft een ietwat afwijkende benadering dan het ISO gekozen. Daar waar het ISO spreekt over één datalink laag, spreekt het IEEE over twee sublagen, de LLC laag (logical link control) en de MAC laag (medium access control). Door verschillende MAC lagen te definiëren en één LCC laag, is het mogelijk om in de toekomst met nieuwe MAC lagen te komen waar bovenliggende netwerklagen (de laag onmiddelijk boven de datalink laag/LLC laag) niet op aangepast hoeven te worden. Immers, de netwerklaag hoeft enkel rekening te houden met de LLC laag, en deze kan, bij het ontstaan van nieuwe MAC implemenaties, voor wat betreft het interface naar boven toe, ongewijzigd blijven!

Het interface tussen IP (de netwerklaag in TCP/IP) en 802.2 (de officiële commissie naam van LLC) is beschreven in RFC 1042.



Bovenstaande het voorbeeld van IEEE 802.3 encapsulation uitgewerkt (NB. 802.5 encapsulatie, Token-Ring encapsulatie dus, werkt vergelijkbaar).

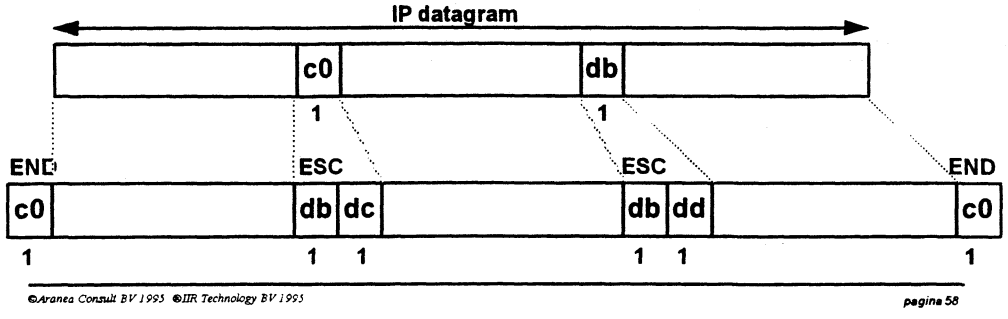
Als eerste het 802.3 formaat, dat slechts op één punt verschilt van het eerder besproken Ethernet II formaat. Het 'type' veld in Ethernet is vervangen door een 'lengte' veld in 802.3. De IEEE commissie gebruikt dus geen type veld als demultiplexing element, maar iets anders: de *dsap* en *ssap* velden in de LLC header. Voor de diverse protocollen is er een destination service access point (zeg maar, het protocol waar het pakketje voor bestemd is) en een source service access point (het protocol waar het pakketje vandaan komt: vaak zal dit identiek zijn aan het *dsap*). Aangezien dit echter niet overeenkomt met de 'type' aanduiding in Ethernet II, moest er nog een extra header ontwikkeld worden. Dat is gebeurd in de vorm van een SNAP header (*Sub-network Access Protocol*). Daarin is een type veld opgenomen, dat *volledig identiek* is aan het type veld uit Ethernet II. Als er nu in de LLC header, in zowel het dsap als ssap veld de waarde 'AA' is opgenomen, dan betekent dat *automatisch* dat er ook een SNAP header is opgenomen met daarin de type aanduiding zoals we die ook uit Ethernet II kennen. Omdat deze types niet door een instituut bewaakt worden (in ieder geval niet allemaal) is er nog een extra veld aan toegevoegd: de organisation code. Deze organisation codes worden uniek gehouden door het IEEE (onder andere door ook gebruik te maken van de eerste 3 bytes van de MAC adressen van Ethernet kaarten).

Bovenstaande betekent dat een 802.3 Ethernet frame effectief *minder* data zal vervoeren dan een Ethernet II frame.

IIR TECHNOLOGY

2.2 IP op seriele verbindingen

- **Voor transport over seriële interfaces wordt gebruik gemaakt van SLIP en CSLIP**
- **CSLIP kan de header informatie reduceren van 40 bytes tot ongeveer 3 tot 5 bytes**
- **Beschreven in RFC's 1055 (SLIP)/1144 (CSLIP)**



Encapsulation in SLIP pakketten is enorm populair geworden dankzij Internet thuiswerkers. SLIP dateert al vanuit 1984, toen het meegeleverd werd met 4.2BSD Unix, met de uitdrukkelijke vermelding dat het *geen* standaard was!

SLIP frames voldoen aan de volgende regels:

- IP datagrammen worden beëindigd met het zogenaamde 'END' character, 0xc0. Veel implementaties laten het IP datagram hier ook mee beginnen.
- Als in het IP datagram 0xc0 characters voorkomen, wordt hiervoor in de plaats het two-byte character 0xdb, 0xdc overgestuurd. 0xdb wordt het SLIP ESC character genoemd.
- Als in het oorspronkelijke IP datagram het SLIP ESC character voorkomt, wordt het two-byte character 0xdb, 0xdd verstuurd.

SLIP is slechts een eenvoudig serieel protocol met een aantal beperkingen:

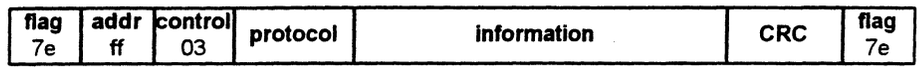
1. Beide kanten van de verbinding moeten elkaars IP adres kennen.
2. Aangezien er geen demultiplexing element aanwezig is, kan er maar één protocol op enig moment worden verzonden.
3. Het SLIP frame heeft géén controle getal. Met andere woorden; ruis op telefoonlijnen kan het frame verminken en fouten zullen dus pas door hogere lagen protocollen ontdekt worden.

CSLIP maakt gebruik van het van Jacobsen header compression algorithme.

IIR TECHNOLOGY

2.2 IP op seriele verbindingen

- **PPP beschreven in RFC's 1331 en 1332**
- **Niet specifiek voor IP, ook voor bv. DECnet**
- **Het PPP protocol beschrijft:**
  - **een manier om datagrammen in te pakken, zowel op asynchrone (81n) als synchrone lijnen;**
  - **een link control protocol (LCP), om een connectie op te zetten en te onderhouden;**
  - **een network control protocol (NCP), om informatie over het netwerklaag protocol uit te wisselen, bijvoorbeeld of header compressie is toegestaan;**



Het PPP protocol is afgeleid van het OSI protocol HDLC, High-level Data Link Control protocol. Het protocol is beschreven in RFC 1331 (encapsulation en het link control protocol, LCP) en RFC 1332 (het network control protocol, NCP).

Ieder PPP frame begint en eindigt met het *flag byte*, waarde 0x7e, gevolgd door een *addres byte* waarvan de waarde altijd 0xff is, en een *control byte* met waarde 0x03. Hierna volgt het *protocol veld*, vergelijkbaar met het type veld in Ethernet. Behalve IP, kan bijvoorbeeld ook DECnet of IPX over PPP vervoerd worden.

- De type-waarde voor IP is 0x0021;
- De generieke waarde 0xc021 duidt op LCP informatie;
- De generieke waarde 0x8021 duidt op NCP informatie.

Deze data, LCP of NCP informatie, is in het *informatie veld* opgenomen. Daarna volgt een *CRC veld* om eventuele errors op de telefoonlijn te detecteren.

Evenals in SLIP, is er een aantal uitzonderingssituaties gedefinieerd in de vorm van escape characters:

- Een byte met waarde 0x7e, die dus overeenkomt met het flag byte, wordt verzonden als de two-byte sequence 0x7d, 0x5e.
- De escape 0x7d wordt verzonden als de two-byte sequence 0x7d, 0x5d (dit is de escape van de escape!).
- Per defintie worden bytes met ASCII waarde kleiner dan 0x20 ook als uitzonderingsbytes verzonden. Als voorbeeld: byte 0x01 wordt verzonden als het two-byte sequence 0x7d,0x21. Dit om te voorkomen dat ASCII control characters worden geïnterpreteerd door de seriële software of de modems.

IIR TECHNOLOGY

2.2 IP op seriele verbindingen

- **Voordelen van PPP t.o.v. SLIP:**
  1. **support voor meerdere protocols**
  2. **CRC in het frame zelf**
  3. **onderhandeling over IP adressen (NCP)**
  4. **TCP en IP header compressie (vgl. CSLIP)**
  5. **een apart LCP om over instellingen te onderhandelen**
- **toch wordt SLIP veel meer gebruikt...**

Het PPP protocol is zonder meer geavanceerder dan het SLIP protocol. De voordelen zijn evident:

- Eén PPP link, bijvoorbeeld tussen twee routers, kan gelijktijdig meerdere laag 3 protocollen vervoeren, zodat het mogelijk is om over één seriële verbinding gelijktijdig meerdere protocollen te vervoeren.
- Het PPP frame bevat een controle getal, zodat laag 2 al kan detecteren of er transmissiefouten zijn geweest en het pakket dus niet hoeft worden doorgegeven aan laag 3.
- PPP kan zelf vaststellen wat het IP adres van 'de andere kant' is; dit hoeft dus niet keihard te worden geconfigureerd.
- Het link protocol kan gebruikt worden om over bepaalde instellingen te onderhandelen.

Maar juist omdát SLIP zo eenvoudig is, wordt dit veel vaker gebruikt dan PPP, temeer daar een aantal TCP/IP implementaties het SLIP protocol bijleveren.

IIR TECHNOLOGY

Intermezzo 4: X.25

- CCITT standaard voor pakket-geschakelde data netwerken 1980, 1984, 1988
- Datanet/1 is X.25 net van PTT Telecom
- DCS is het X.25 net van BELGACOM
- X.25 is *interfacespecificatie*
  - snelheden 2.4 tot 64 kbps
  - Connection oriented: PVC en SVC
  - Error detection en recovery, flow control
  - 5 byte header, tot 128 byte payload (in publieke netwerken)

X.25 netwerken vormen al jaren een aantrekkelijke mogelijkheid voor wie weinig verkeer tussen vele locaties wil uitwisselen en daarbij zo min mogelijk zorgen over het communicatienet wil hebben. Bekende voorbeelden zijn netwerken tussen garagebedrijven en de importeur; via het netwerk kunnen onderdelen en complete auto's besteld worden, voorraden en levertijden worden opgevraagd en soms ook administratieve diensten worden gebruikt.

De X.25 netwerken van de verschillende carriers zijn wereldwijd gekoppeld, zodat X.25 ook voor zeer internationale organisaties een aantrekkelijk aanbod vormt; alle lokale technische en organisatorische verschillen worden door X.25 opgelost (vergelijk met een netwerk van kies-modems).

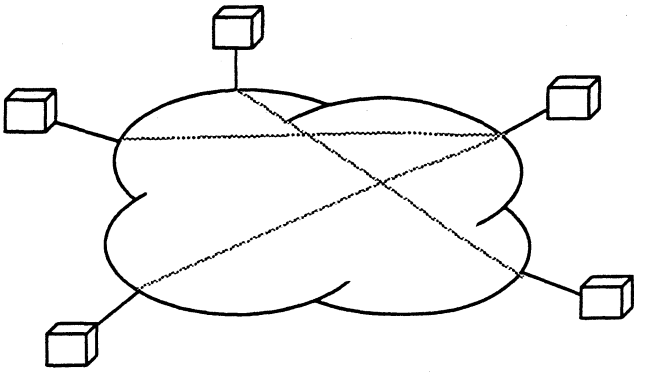
X.25 wordt ook wel toegepast in privé netwerken. Men huurt dan vaste lijnen waarover, vaak naast andere verkeersvormen (telefonie, SNA), ook een X.25 faciliteit wordt gelegd.

Het is van groot belang zich te realiseren dat X.25 slechts een interface specificatie is; hoe het achterliggende netwerk feitelijk werkt, is niet gedefinieerd. Zo zijn er TDMs (Time Division Multiplexers) die aan de gebruikerskant een X.25 interface bieden. Het opzetten van een X.25 SVC (Switched Virtual Circuit) betekent dan niets anders dan het toewijzen van een TDM kanaal.

IIR TECHNOLOGY

I4: X.25

- **Voorbeeld van *cloud* (wolk) techniek**
- **N logische verbindingen**
- **Eén fysieke link**



Een belangrijk voordeel van een X.25 aansluiting is dat, via de enkele fysieke lijn een (groot) aantal logische verbindingen naar vele verschillende bestemmingen kan worden onderhouden. Daarom worden X.25 netwerken en nieuwere structuren die deze eigenschap delen, zoals Frame Relay, vaak als wolk *(cloud)* getekend. Wat zich binnen de wolk afspeelt is niet gedefinieerd, maar het biedt virtuele verbindingen (Virtual Circuits) tussen vele bestemmingen.

Er wordt onderscheid gemaakt tussen:

- Permanent Virtual Circuits (PVCs) worden door de netwerkbeheerder opgezet en blijven gedurende lange tijd (weken, maanden) actief; te vergelijken met een huurlijn, maar het tarief is voor een deel afhankelijk van de hoeveelheid verzonden informatie.
- Switched Virtual Circuits (SVCs) worden door de gebruiker opgezet voor de duur van een conversatie (minuten of uren); vergelijkbaar met een gekozen modemverbinding, maar het tarief is afhankelijk van de hoeveelheid verstuurde informatie. Een SVC kan veel sneller worden opgezet dan een gekozen telefoon(modem)verbinding.

Het X.25 netwerk biedt een kwalitatief hoge functionaliteit: error detectie en correctie, flow control, aanpassing tussen de snelheden van zender en ontvanger (een 2400 bps terminal kan op een 64 kbps host worden aangesloten), de mogelijkheid van *closed user groups*, etc.

Omdat de publieke X.25 netwerken vooral ook voor terminal-host verkeer ontworpen zijn, gebruiken zij vrij kleine frames (128 byte *payload* plus een 5-byte *header*).

IIR TECHNOLOGY

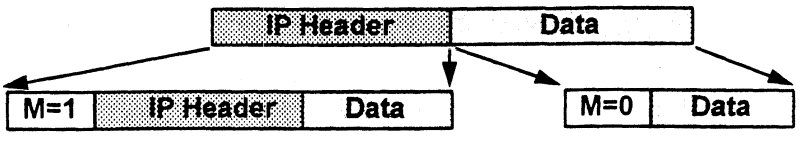
I4: X.25

- X.25 (1984) en X.25 (1988) definieert een connection-oriented service gebaseerd op datagrammen
- 'IP over X.25' beschrijft hoe IP over een connection-oriented netwerk gebruikt kan worden
- Dynamisch opzetten van SVC als IP datagram verstuurd moet worden
- Na 'inactivity timer' wordt SVC afgebroken

IIR TECHNOLOGY

I4: X.25

- IP maakt gebruik van karakteristieken van het X.25 netwerk:
  - (De)multiplexing m.b.v. X.25 Call User Data Field (CUDF): waarde voor IP is 204
  - (De)fragmentatie m.b.v. X.25 fragmentatie faciliteiten



IIR TECHNOLOGY

I4: X.25

- Bij X.25 netwerken wordt gebruik gemaakt van de zogenaamde X.121 NUA's (Network User Adresses)
- Er wordt gebruik gemaakt van adrestabellen om IP adressen op X.121 adressen te mappen

IP address	X.121 address
125.2.3.212	23424354657687
89.0.0.1	23424565432123
145.46.2.1	23427645665543
140.150.1.1	23426435676654

IIR TECHNOLOGY

Intermezzo 5: Fast Packet Switching

- **Variabele Lengte:** **Frame Relay**
- **Vaste Lengte:** **Cell Relay**
- **Cell Relay volgens CCITT:** **ATM**

Binnen Fast Packet Switching vinden we systemen die pakketten van een wisselende lengte (*frames*) transporteren en systemen die met pakketten met een vaste lengte (*cellen*) werken; die systemen heten dan ook *frame relay* en *cell relay*.

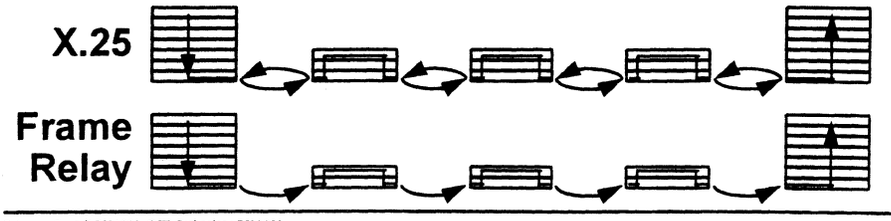
Van alle denkbare frame relay systemen is er één (bijna) gestandaardiseerd: Frame Relay.

Van alle denkbare cell relay systemen zijn er twee (bijna) gestandaardiseerd: ATM (CCITT) en DQDB (IEEE).

IIR TECHNOLOGY

I5: Wat is Frame Relay

- **X.25**
  zonder Error Recovery
  zonder Flow Control
  sneller: 64 kbps - 2 Mbps en meer...
  multiplexing op laag-2 (DLC) i.p.v. laag-3
- **"Laag 2,5"**
  LAP-F, sub-set van LAP-D



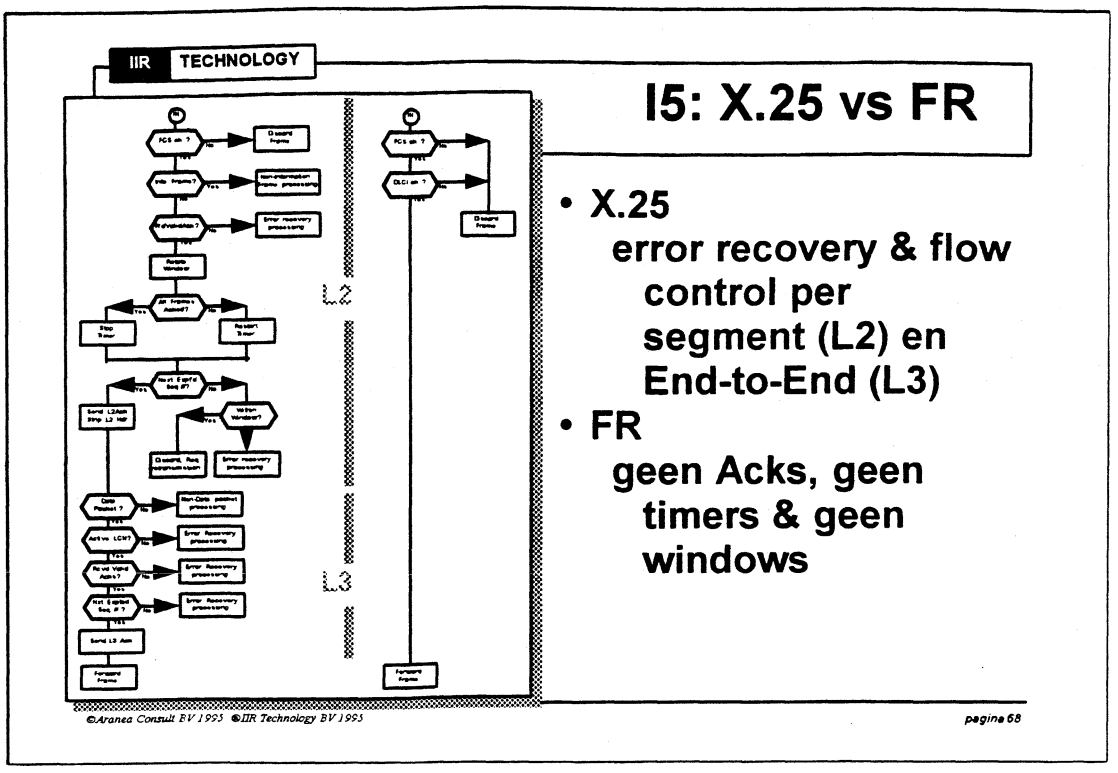
Frame Relay is X.25, maar...

- zonder error recovery
- zonder flow control
- met hogere snelheden (tot 2 Mbps en meer)

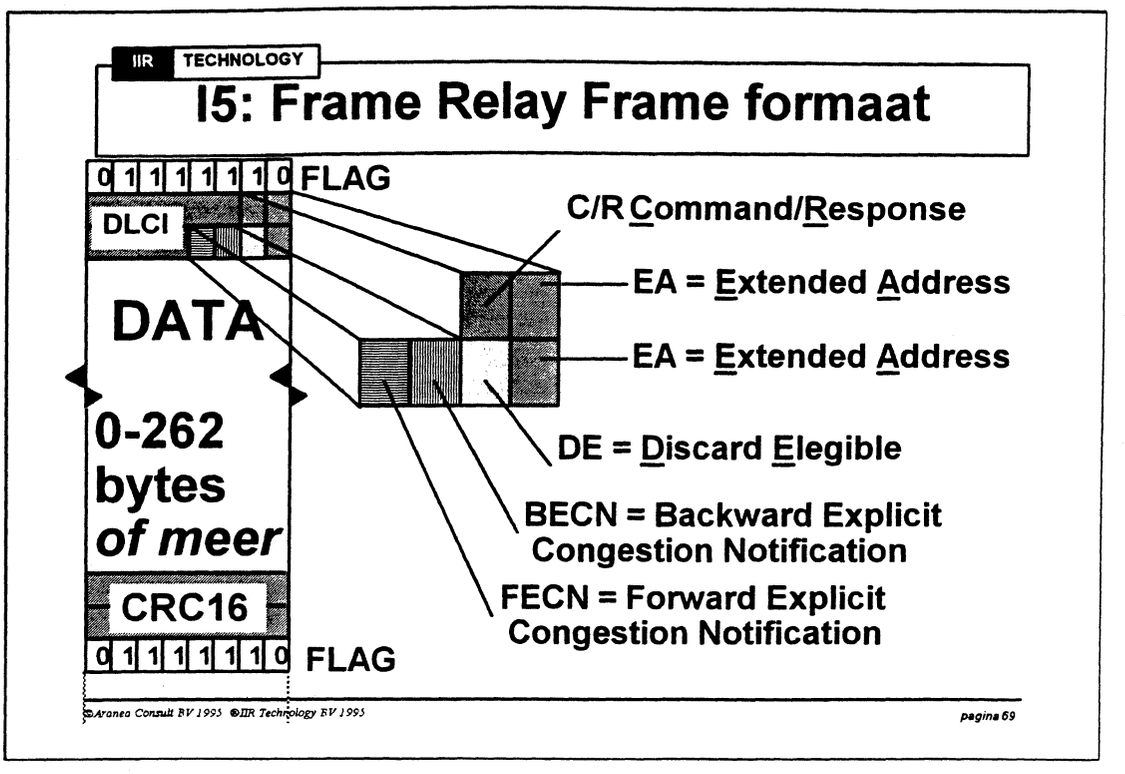
De netwerklaag (OSI laag 3) ontbreekt bij Frame Relay, maar in plaats daarvan worden er op laag 2 verschillende logische kanalen gemultiplext.

LAP-F (Link Access Protocol for Frame Relay) is een subset van LAP-D.

Frame Relay is vooral een pragmatische benadering van het Fast packet probleem: er hoeft geen nieuwe hardware voor ontworpen te worden en voor de belangrijkste toepassing (LAN-LAN koppeling via bridges of routers) kan worden volstaan met een software update.



Wanneer we de stroomdiagrammen van een X.25 switch en een Frame Relay switch naast elkaar leggen, wordt het dramatische verschil in complexiteit direct duidelijk.



FR frames worden begrensd door twee HDLC *flag* bytes ('01111110'). In de rest van het frame mag dit patroon niet voorkomen, wat met de bekende *bitstuffing* techniek wordt gegarandeerd.

De header beslaat 2 bytes en bevat:

- een 10-bits Data Link Connection Identifier (DLCI)
- een Command/Response bit (een overblijfsel van het HDLC protocol, maar niet gebruikt door het Frame Relay protocol)
- een Forward Explicit Congestion Notification bit (FECN)
- een Backward Explicit Congestion Notification bit (BECN)
- een Discard Elegible bit (DE)
- twee Extended Address bits, waarvan het eerste '0' is en het tweede '1'. Het is mogelijk het aantal adresbits in de toekomst uit te breiden. In dat geval worden alle EA-bits '0', behalve die van het laatste header byte.

Daarna volgen de data bytes, gevolgd door een CRC-16 en de afsluitende *flag*. Deze kan meteen ook de start van een nieuw frame zijn.

IIR TECHNOLOGY

I5: Frame Relay DLCI

- **Data Link Connection Identifier**
  10 bits, 1024 mogelijkheden
  Uitbreiding met 8 of 16 bits mogelijk

0	**Call Control Signaling**
1-15	***gereserveerd***
16-1007	**PVCs**
1008-1022	***gereserveerd***
1023	**Consolidated Link Layer Management òf Local Management Interface (FR Forum)**

De DLCI speelt de rol van 'adres' binnen het Frame Relay netwerk. Maar anders dan bij Ethernet of Token Ring heeft de DLCI alleen betekenis op de link (de verbinding) waarop hij voorkomt. Het is dus heel normaal dat een frame wordt verstuurt met DLCI=234, waarna de eerste FR switch verder gaat met DLCI=876, en zo verder tot hij bij de eindbestemming aankomt met DLCI=148.

De standaard DLCI is 10 bits lang, goed voor 1024 verschillende logische verbindingen per link. Juist omdat de betekenis strikt lokaal is, is een Frame Relay netwerk daarmee niet beperkt tot 1024 stations.

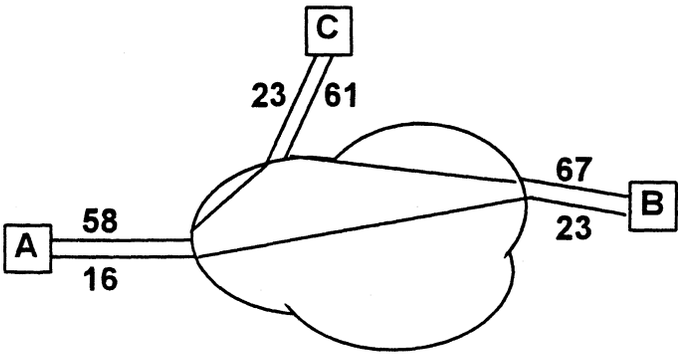
Van de 1024 mogelijke DLCIs (0-1023) zijn er een aantal gereserveerd:

DLCI 0	wordt gebruikt voor signalering tussen het netwerk en het aangesloten apparaat
DLCI 1-15	zijn voorlopig gereserveerd
DLCI 16-1007	worden gebruikt voor Permanent Virtual Circuits (PVCs)
DLCI 1008-1022	zijn voorlopig gereserveerd
DLCI 1023	wordt gebruikt voor het Consolidated Link Layer Management protocol of het Local Management Interface van het Frame Relay Forum

IIR TECHNOLOGY

I5: Frame Relay Lokale Adressering

- **DLCI heeft alleen lokale betekenis**

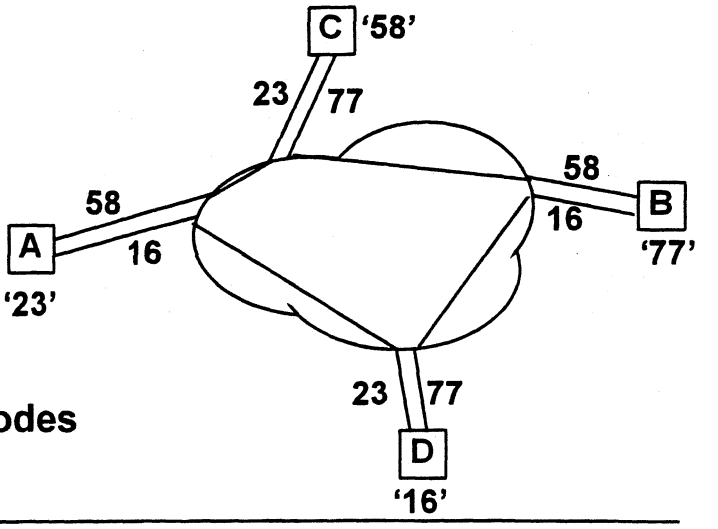


Dit plaatje geeft nog eens weer hoe de DLCIs in een standaard Frame Relay netwerk een willekeurige waarde hebben. Maar let op: in plaats van een willekeurige waarde kunnen we het ons makkelijk maken en een keuze doen die tenminste voor de netwerkbeheerder makkelijk is...

IIR TECHNOLOGY

I5: Frame Relay Globale Adressering

- 'Afspraak'



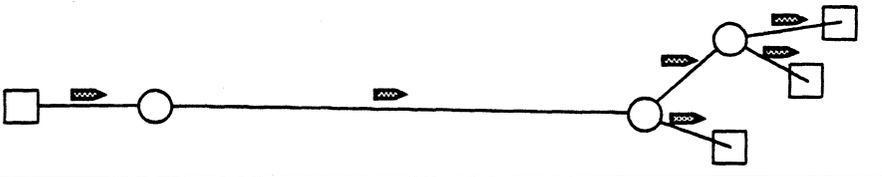
- Makkelijk
- Max 992 nodes

We kunnen bij verschillende implementaties kiezen voor *Global Addressing*. Daarbij geven we elk station een nummer (uit de reeks 16-1007) en kiezen voor de DLCIs steeds een waarde die overeenkomt met het nummer van het station aan het andere eind van de verbinding. Een PVC van station 23 naar station 16 heeft dus op de uitgang van station 23 een DLCI 16 en op het interface van station 16 een DLCI van 23. Dat maakt het een stuk makkelijker te volgen wat er precies gebeurt. Het stelt wel een maximum aan het aantal stations in ons Frame Relay netwerk (992), maar dat zal voor weinigen echt een bezwaar zijn.

IIR TECHNOLOGY

I5: Frame Relay Adressering

- PVCs
  - local addressing
  - global addressing
- SVCs
  - ANSI/CCITT standaard
  - ISDN Digital Subscriber Signaling No. 1
- Multicast



PVCs worden opgezet door de beheerder van het Frame Relay netwerk. Hoe dat gebeurt, valt buiten de scope van Frame Relay, zoals bijna alles wat er zich binnen het Frame Relay netwerk afspeelt: FR is, net als X.25, een interface specificatie en wat er binnen de *cloud* gebeurt blijft buiten beeld. Het is volstrekt legaal om een Frame Relay service aan te bieden, gebaseerd op een TDM netwerk met ergens één grote switch. Maar het is ook mogelijk verspreid opgestelde switches op basis van Frame Relay onderling te koppelen en zo een 'echt' packet switching netwerk te maken.

In de toekomst zullen Frame Relay services ook Switched Virtual Circuits (SVCs) bieden, maar daarvoor zijn de standaarden nog niet afgerond. ANSI en CCITT werken aan een standaard, die gebruik zal maken van delen van ISDN's Subcriber Signalering #1.

Tenslotte zijn er ook specificaties in de maak voor een multicast functie. Het is de bedoeling dat daarbij de switches voor het dupliceren van de frames zorgen, zodat een minimum aan bandbreedte wordt gebruikt.

IIR TECHNOLOGY

Intermezzo 6: ATM

- Nieuwe standaard voor hoge-snelheids-netwerken (45 Mbps - 200 Gbps)
- Cell-relay: celgrootte = 53 bytes
- Zowel voor LAN als WAN
- Stelt eisen aan gebruikte protocollen voor windowing en delay
- Biedt connectie-georiënteerde services
- Wordt gewerkt aan RFC voor IP over ATM
- Ondersteund geen broadcasting

IIR TECHNOLOGY

2.2 IP en WAN technieken

- **Vele manieren om IP te vervoeren**
- **Geen fragmentatie nodig zoals bij LAN technieken**
- **Voor IP is WAN verbinding een punt-punt verbinding**

Bij LAN technieken wordt zoals eerder vermeld een IP datagram verpakt in het data gedeelte van een MAC frame. Hierbij kan fragmentatie optreden. De MAC frames worden vervolgens volgens de LAN specificaties verzonden naar ieder in het LAN aanwezig station.

Voor WAN's gaat dit anders. Hier moet zonodig eerst een virtueel-circuit opgebouwd worden en op het moment dat dat tot stand is gebracht, is er eigenlijk een punt-punt verbinding ontstaan tussen de twee communicerende stations.

Deze verbinding kan beschouwd worden als een gewone seriële verbinding, waarbij dus PPP of SLIP gebruikt kan worden, maar ook kan gebruik worden gemaakt van de frame formaten van het gebruikte netwerk, waarbij IP datagrammen verpakt worden in bijvoorbeeld een X.25 dataveld of Frame-Relay dataveld. Fragmentatie zal nooit plaatsvinden (althans niet op IP niveau), omdat er geen andere frames tussendoor kunnen komen.

Ook is de vertaling van IP adres naar fysiek adres anders bij de genoemde WAN technieken dan bij de LAN technieken, omdat er geen gebruik kan worden gemaakt van broadcasts.

IIR TECHNOLOGY

2.3 Van logisch naar fysiek

- ***IP adressen* zijn logische adressen*: fysieke adressen* nodig om frame ècht te versturen**
  - ***Tabel***
    - **veel werk, veel fouten (zie X.121 mapping)**
  - ***Direct mapping***
    - **'bereken' fysiek adres uit logische adres**
  - ***Dynamic binding***
    - **Fysiek adres vragen aan host**

Toch nu toe hebben we enkel gesproken over logische adressen, bijvoorbeeld adres 145.146.203.254. Deze IP datagrammen moeten echter verzonden worden over een fysiek netwerk, en dus ook geadresseerd worden aan (bijvoorbeeld) fysieke Ethernet adressen. Er zijn drie mogelijkheden om logische adressen te laten matchen met fysieke adressen:

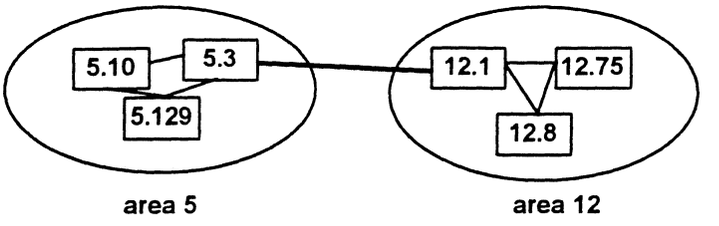
- tabellen - Het is mogelijk om machines uit te rusten met tabellen, met daarin logische adressen en de bijbehorende fysieke adressen. Het mag duidelijk zijn dat dit een arbeidsintensief proces is, waarbij de foutkans erg groot is.
- direct mapping (static binding) - Als tweede optie kan ervoor gekozen worden om een bepaalde 'match' te maken tussen logische adressen en fysieke adressen. Dit is de manier waarop onder andere DECnet te werk gaat. Als een DECnet node opstart, zal deze het oorspronkelijke Ethernet adres overschrijven met een nieuw Ethernet adres, dat hij afleidt van zijn eigen DECnet adres. Andere DECnet nodes die met deze node willen communiceren kunnen uiteraard hetzelfde doen: uit het logische DECnet adres van de node het bijbehorende fysieke Ethernet adres bepalen.
- dynamic binding - Op basis van een logisch adres een protocol gebruiken om het bijbehorende fysieke adres te achterhalen. Dit is de manier waarop IP werkt.

NB. Wellicht ten overvloede, *host* is de generieke term voor een willekeurige TCP/IP pratende machine (PC, router, Unix hosts, minicomputer, etcetera).

IIR TECHNOLOGY

2.3 Van logisch naar fysiek

- **Intermezzo: DEC's 'direct mapping' algorithme**



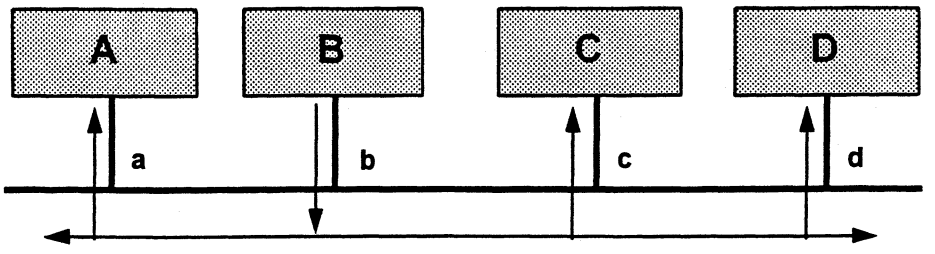
'Nieuwe' MAC-adres als volgt berekend:

1. (area_nr * 1024) + host_nr
2. Dit 16 bit decimale getal -> hexadecimaal
3. Bytes van hex getal omdraaien
4. Dit vastplakken aan AA00.0400

IIR TECHNOLOGY

2.3 Van logisch naar fysiek

- **Address Resolution Protocol (ARP): de vraag**

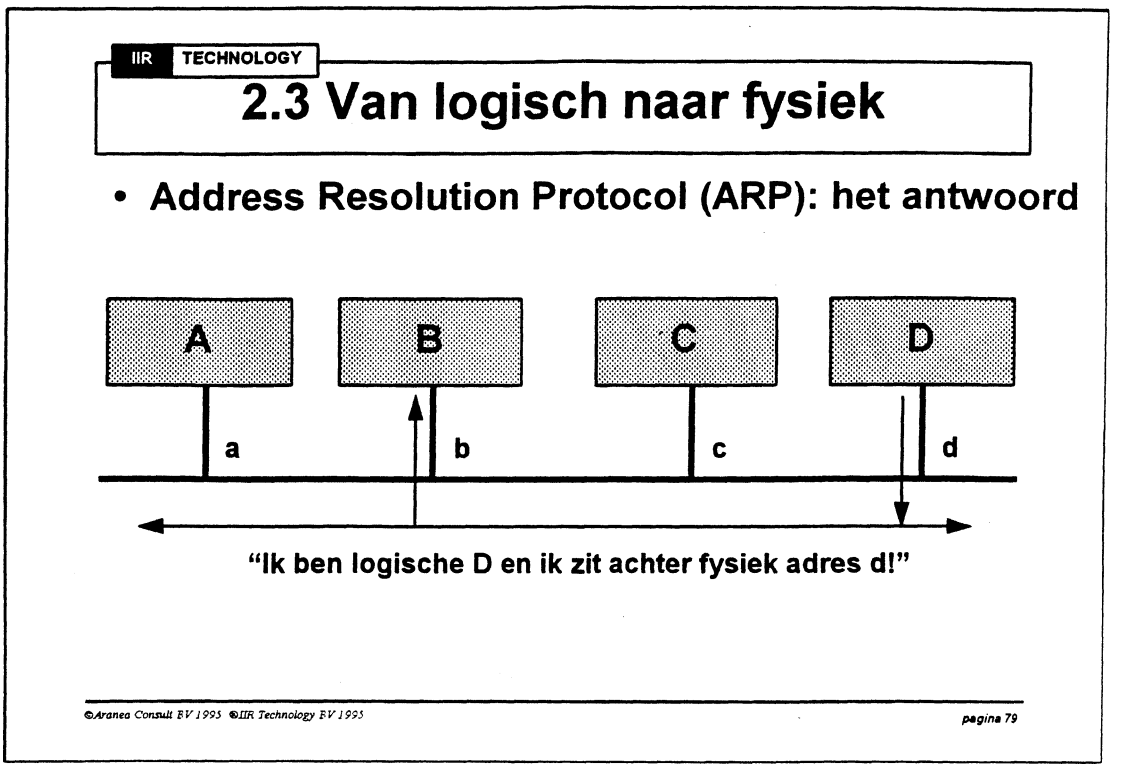


"Ik ben logische B en ik zoek logische D, melden aub!"

Het ARP protocol is een protocol dat 'rechtstreeks' tegen de onderliggende laag (datalink laag) 'praat' en gebruik maakt van de broadcast mogelijkheden die geboden worden door die laag.

In bovenstaand voorbeeld zal machine B een Ethernet broadcast over het netwerk versturen waarin staat aangegeven dat hij op zoek is naar logische machine D. Dit Ethernet frame zal door alle aangesloten machines worden opgepakt (A, C en D).

Dit kan dus alleen gebruikt worden op netwerken waarop broadcasts mogelijk zijn.



Machine D is de enige machine die op de ARP reageert. Machine D herkent namelijk zijn eigen logische adres. In het ARP pakket staat de afzender vermeld. Machine D zal nu met een gericht Ethernet frame ('unicast') zijn adres bekend maken aan machine B. Machine B zal de *Ethernet adres-IP adres* combinatie opnemen in de zogenoemde 'ARP-cache'. Een volgende keer dat B iets te versturen heeft naar D, zal B het bijbehorende adres kunnen opzoeken in de ARP-cache. De ARP-cache is een *dynamische tabel*. Na verloop van tijd (default 20 minuten) zullen entries verwijderd worden uit deze tabel, om te voorkomen dat er entries blijven bestaan die al lang niet meer valide zijn (bijvoorbeeld omdat host D een nieuwe Ethernet kaart gekregen heeft en nu dus niet meer Ethernet adres 'd' maar Ethernet adres 'e' heeft).

IIR TECHNOLOGY

2.3 Van logisch naar fysiek

- **PDU beschrijving van het ARP protocol:**

0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1
Hardware Type		**Protocol type**	
HW size	**Prot size**	**Operation code**	
Source Hardware address			
(continued)		**Source Protocol address**	
(continued)		**Dest. Hardware address**	
(continued)			
Destination Protocol address			

Bovenstaand de PDU-indeling van een ARP pakket. ARP pakketten worden 'rechtstreeks' ingepakt in Ethernet pakketten, met type veld 0x0806. RARP pakketten (zie ook verderop) worden ingepakt met type aanduiding 0x8035. De betekenis van de verschillende velden:

Hardware type - Dit duidt op het type hardware adres. In het geval van Ethernet is dit waarde 1.

Protocol type - Dit duidt op het gebruikt (hogere lagen) protocol. In het geval van IP is dit waarde 0x800.

HW size - Dit duidt op de lengte van het gebruikte hardware adres. In het geval van Ethernet is dat '6' (bytes).

Prot. size - Dit duidt op de lengte van het gebruikte hogere lagen protocol. In het geval van IP is dit waarde '4'.

Operation - Er zijn vier type operations gedefinieerd:

- ARP request (operation code 1)
- ARP reply (operation code 2)
- RARP request (operation code 3)
- RARP reply (operation code 4)

Source Hardware Address - Ethernet adres van de afzender (in het geval van Ethernet).

Source Protocol Address - IP adres van de afzender (in het geval van IP).

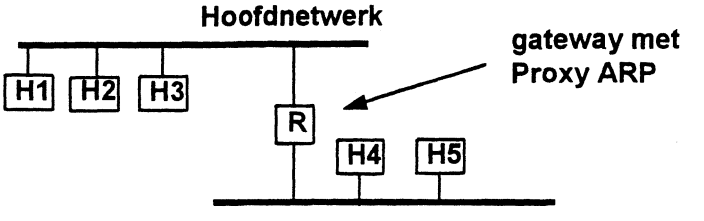
Destination Hardware Address - Ethernet adres van de geadresseerde (in het geval van Ethernet).

Destination Protocol Address - IP adres van de geadresseerde (in het geval van IP).

IIR TECHNOLOGY

2.3 Van logisch naar fysiek

- ***Proxy ARP*** **wordt gebruikt om stations op netwerken te laten 'geloven' dat bepaalde andere stations te bereiken zijn.**
- ***Gratuitous ARP*** **wordt gebruikt om te controleren of het IP adres van een host niet al in gebruik is ('*kostenloos*', '*overbodig*').**



Er bestaat ook nog zoiets als Proxy-ARP. Dit kan gebruikt worden om ARP requests, die eigenlijk door hosts op een ander netwerk afgehandeld zouden moeten worden, te laten afhandelen door een router (wat precies de functionaliteit van routers is, wordt besproken in hoofdstuk 4). Als H3 op zoek gaat naar H4, dan kan de router R zodanig geconfigureerd worden, dat H3 al zijn pakketten voor H4 naar router R zal sturen. Router R zal namelijk op de ARP requests van H3 reageren, waardoor H3 denkt dat het Ethernet adres van de router R, het Ethernet adres van H4 is! Op die manier kunnen machines die zich eigenlijk niet op hetzelfde *fysieke* netwerk bevinden, stiekem toch door dat netwerk benaderd worden!

Een andere 'speciale' vorm van ARP is 'gratuitous ARP'. 'Gratuitous' staat voor kostenloos. Het kan gebruikt worden door IP hosts tijdens booten om te controleren of er al een host met hun IP adres in de lucht is. Mocht er inderdaad een reply komen op het ARP request, dan betekent dat, dat een host met het betreffende IP nummer voorkomt . Op dat moment zal de host die aan het booten is hiermee stoppen en een melding geven. Gratuitous ARP is (helaas) niet verplicht...

IIR TECHNOLOGY

2.3 Van logisch naar fysiek

- **ARP gebaseerd op principe van broadcasting**
- **Connectie-georiënteerde verbindingen ondersteunen alleen unicasting of multicasting**
- **Gewijzigd ARP mechanisme nodig voor dynamische adres mapping**
- **Statische adres mapping via tabellen**

Als het onderliggende netwerk geen broadcasting mogelijkheid heeft (en dat geldt in principe voor alle WAN technieken), dan kan er geen gebruik worden gemaakt van de standaard ARP methode.

Wel kan er gebruik worden gemaakt van statische binding door het bijhouden van tabellen.

Een andere mogelijkheid is een kleine modificatie van het ARP mechanisme. Hiervoor is een RFC ontwikkeld, zodat ARP ook gebruikt kan worden op Frame Relay netwerken. Voor X.25 netwerken is het gebruikelijk om tabellen te gebruiken.

IIR TECHNOLOGY

2.3 Van logisch naar fysiek

- **Multiprotocol over Frame-Relay**
  - **Gestandaardiseerd in RFC1490**
  - **Niet alleen voor IP, kan ook voor andere protocollen gebruikt worden, alsmede voor bridging over Frame Relay**
  - **Kan gebruikt worden met LLC, SNAP, IPX en IP**
  - **Kan gebruikt worden voor ARP, RARP and IARP**
  - **Herdefinieert het dataveld van een FR pakket (niet het adres deel !)**

RFC 1490 beschrijft hoe op een Frame-Relay netwerk logische adressen (IP adressen) vertaald kunnen worden naar fysieke adressen (de DLCI's aan de andere kant).

De RFC is niet alleen voor IP gemaakt, maar beschrijft tevens hoe LLC, SNAP en IPX over een frame-relay netwerk getransporteerd kunnen worden, of gebridged kunnen worden over een frame-relay netwerk.

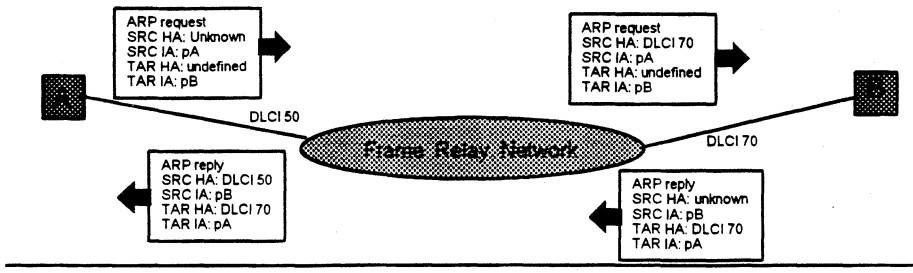
Hiervoor is een speciaal soort frame ontwikkeld, waarbij het data gedeelte van het frame relay pakket op een speciale manier ingedeeld wordt.

Hierdoor is het mogelijk om zowel ARP, RARP als IARP te gebruiken in een Frame-Relay verbinding.

IIR TECHNOLOGY

2.3 FR: ARP, RARP and IARP

- **Alleen in combinatie met een SNAP header**
- **ARP**
  - **End stations kennen eigen en bestemming hardware adres niet (incoming DLCI).**
  - **ARP request wordt gebroadcast door het netwerk over elke relevante PVC.**
  - **Bestemming beantwoord ARP , vult inkomende DLCI in als zijn hardware adres**
  - **Originating station ontvangt ARP reply en gebruikt het inkomende DLCI number als zijn eigen hardware adres.**



Het station dat berichten wil versturen, maar de ontvangers fysieke adres niet kent, zal een ARP request verzenden. Het enige dat het station kent, is zijn eigen IP adres en het IP adres van de ontvanger. Deze worden keurig ingevuld.

Het ARP request zal over een DLCI verzonden worden naar het netwerk en aankomen bij de ontvanger over een ander DLCI. Dat DLCI nummer (DLCI H_{src}) wordt door de ontvanger gezien als het hardware adres van de afzender.

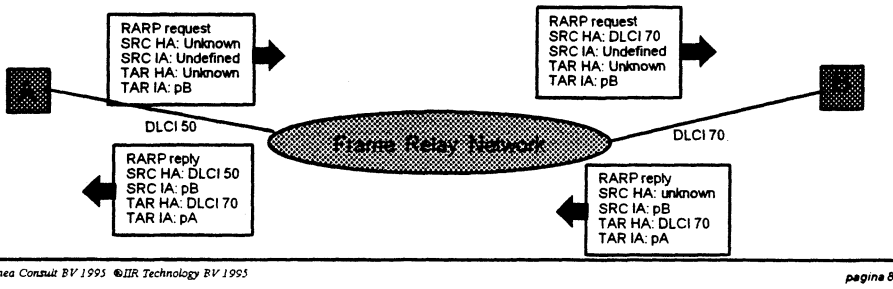
De ontvanger zal een ARP reply terugsturen, waarbij nu het hardware adres van de afzender (DLCI H_{src}) is ingevuld via een bepaald DLCI.

De ARP reply komt aan bij de afzender op een bepaald DLCI. Dit DLCI (DLCI H_{dest}) zal door de afzender nu gezien worden als hardware adres van de bestemming. Nu weet station A dus dat de berichten voor B verzonden moeten worden via DLCI H_{dest}, en B weet dat de berichten voor A verzonden moeten worden via DLCI H_{src}.

IIR TECHNOLOGY

2.3 Frame-Relay: RARP

- **Alleen in combinatie met een SNAP header**
- **RARP (reversed ARP)**
  - **End stations kennen eigen en bestemming hardware adres niet (incoming DLCI)**
  - **RARP request wordt gebroadcast door het netwerk naar RARP server**
  - **Bestemming beantwoord RARP , vult inkomende DLCI in als zijn hardware adres en vult het source IP address in**
  - **Originating station ontvangt RARP reply en gebruikt het inkomende DLCI number als zijn eigen hardware adres en krijgt zijn IP adres**



Het station dat berichten wil versturen, maar zijn eigen IP adres niet kent, zal een RARP request verzenden. Het enige dat het station kent, is het IP adres van de ontvanger. Dat wordt keurig ingevuld.

Het RARP request zal over een DLCI verzonden worden naar het netwerk, en aankomen bij de ontvanger over een ander DLCI. Dat DLCI nummer (DLCI H_{src}) wordt door de ontvanger gezien als het hardware adres van de afzender.

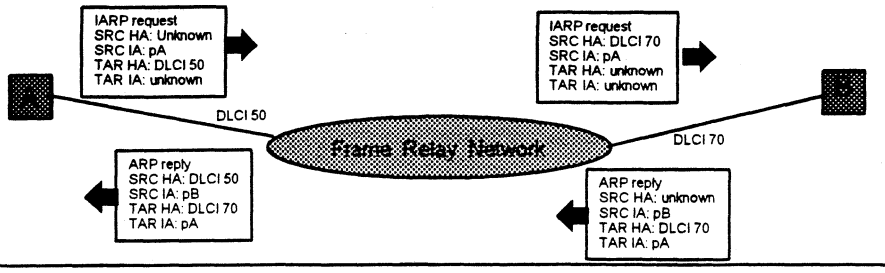
De ontvanger zal een RARP reply terugsturen, waarbij nu het hardware adres van de afzender (DLCI H_{src}) en het IP adres voor de afzender zijn ingevuld via een bepaald DLCI.

De RARP reply komt aan bij de afzender op een bepaald DLCI. Dit DLCI (DLCI H_{dest}) zal door de afzender nu gezien worden als hardware adres van de bestemming. Nu weet station A dus dat de berichten voor B verzonden moeten worden via DLCI H_{dest}, en B weet dat de berichten voor A verzonden moeten worden via DLCI H_{src}.

IIR TECHNOLOGY

2.3 Frame Relay: IARP

- **Alleen in combinatie met een SNAP header**
- **IARP (inversed ARP)**
  - **End stations kennen DLCI adressen, maar niet welk IP adres daar bij hoort**
  - **IARP request wordt ge-unicast door het netwerk over een bekende DLCI**
  - **Andere end station beantwoord IARP , vult het inkomende DLCI in als zijn hardware adres en vult source IP address in**
  - **Originating station ontvangt normale ARP reply en gebruikt het inkomende DLCI als eigen hardware adres and krijgt het IP adres van de DLCI**



Het station dat berichten wil versturen, maar de ontvangers IP adres niet kent, zal een IARP request verzenden. Het enige dat het station kent, is de gebruikte DLCI en zijn eigen IP adres . Deze worden keurig ingevuld.

Het IARP request zal over een DLCI verzonden worden naar het netwerk, en aankomen bij de ontvanger over een ander DLCI. Dat DLCI nummer (DLCI H_{src}) wordt door de ontvanger gezien als het hardware adres van de afzender.

De ontvanger zal een ARP reply terugsturen, waarbij nu het hardware adres van de afzender (DLCI H_{src}) is ingevuld en het IP adres van de ontvanger, via een bepaald DLCI.

De ARP reply komt aan bij de afzender op een bepaald DLCI. Dit DLCI (DLCI H_{dest}) zal door de afzender nu gezien worden als hardware adres van de bestemming. Nu weet station A dus dat de berichten voor B verzonden moeten worden via DLCI H_{dest}, en B weet dat de berichten voor A verzonden moeten worden via DLCI H_{src}.

IIR TECHNOLOGY

2.4 ICMP protocol

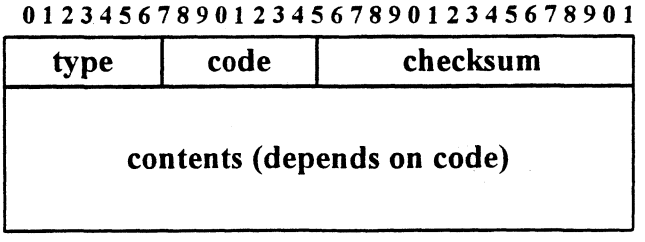
- **Apart protocol ontwikkeld om 'dienstberichten' uit te wisselen**
- **ICMP pakketjes, ingepakt in IP pakketjes**
- **Gestandaardiseerd in RFC792**
- **Verplicht onderdeel van TCP/IP implementaties (opgenomen in Host Requirements RFC)**

Zeer sterk gerelateerd aan het IP protocol, en in de praktijk ook bijna altijd ondergebracht in hetzelfde stuk coding, is het ICMP protocol. Dit protocol wordt gebruikt om 'dienstberichten' uit te wisselen tussen IP hosts. ICMP pakketten worden ingepakt in IP datagrammen. Er worden nooit ICMP berichten verzonden over verloren gegaan ICMP berichten!

IIR TECHNOLOGY

2.4 ICMP protocol

- **ICMP messages kunnen verschillende 'inhoud' hebben afhankelijk van type/code**
- **Per type zijn diverse codes mogelijk**
- **Sommige pakketjes hebben als 'inhoud' een deel van de oorspronkelijke IP header**



Onderstaand de tabel met gedefinieerde types/codes voor ICMP pakketten.

Type	Code	Description	query	error
0	0	echo reply (*PING*)	*	
3		destination unreachable		
	0	network unreachable	*	
	1	host unreachable		*
	2	protocol unreachable	*	
	3	port unreachable		*
	4	frag. needed, Don't fragment set	*	
	5	source route failed		*
	6	destination network unknown		*
	7	destination host unknown		*
	9	destination network adm. prohibitied		*
	10	destination host adm. prohibited		*
	11	network unreachable for TOS		*
	12	host unreachable for TOS		*
	13	communication prohibited by filter		*
	14	host precedence violation		*
	15	precedence cutoff in efect		*
4	0	source quench		*
5		redirect		
	0	redirect for network	*	
	1	redirect for host		*
8	0	echo request (*PING*)		
9	0	router advertisement	*	
10	0	router sollicitation	*	
11		time exceeded		
	0	Time = 0 during transit		*
	1	Time = 0 during reassambly		*
13	0	timestamp request	*	
14	0	timestamp reply	*	
17	0	address mask request	*	
18	0	address mask reply	*	

IIR TECHNOLOGY

2 Samenvatting

- IP is een connectionless best-effort delivery protocol
- Doet aan fragementatie en demultiplexing
- Kent een hiërarchische adresstructuur met daarin een aantal klassen netwerken (A, B, C)
- Geschikt om op LAN en WAN netwerken te gebruiken
- Hulp-protocollen:
  - ARP om fysieke adressen te kunnen vinden bij bekende logische adressen
  - ICMP om dienstberichten uit te wisselen

IIR TECHNOLOGY

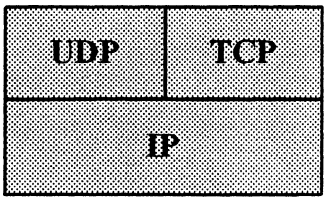
3. De transportlaag

3.1 Algemeen
3.2 UDP
3.3 TCP

IIR TECHNOLOGY

3.1 Algemeen

- **Binnen TCP/IP zijn twee implementaties van de transportlaag beschikbaar**
- **Connection-less, unreliable: UDP**
- **Connection-oriented, reliable:TCP**
- **Beide verplichte implementaties van TCP/IP**



TCP/IP heeft twee implementaties op de transportlaag, te weten UDP en TCP. Beide geven invulling aan een bepaalde behoefte: UDP levert een connection-less, unreliable transport service, TCP leverte een connection oriented, reliable transport service.

IP kan beide protocollen onderscheiden ('demultiplexing') aan de hand van het protocolnummer: TCP gebruikt nummer 6, UDP gebruikt nummer 17.

IIR TECHNOLOGY

3.2 UDP

- **UDP: User Datagram Protocol**
- **Laat verschillende programma's binnen één computer onafhankelijk datagrammen versturen**
- **Handig voor korte vraag-antwoord communicatie:**
  - **datum/tijd opvragen**
  - **bootstrap informatie (BOOTP)**
  - **SNMP**

Simple Network Management Protocol

Het User Datagram Protocol (UDP) is het eenvoudigste transportprotocol van TCP/IP. Het dient vooral voor korte vraag- en antwoord transacties, waarvoor het opzetten en weer afbreken van een volledige TCP-sessie niet de moeite waard is.

Binnen één computer kunnen er een heleboel programma's tegelijk gebruik maken van UDP (en TCP). Daarom gebruikt UDP het begrip *poort* om zijn verschillende klanten uit elkaar te houden.

Elke netwerkcommunicatie gaat tussen twee partijen, waarvan er één het initiatief tot communiceren neemt. Dat is per definitie de *client*. De andere partij noemen we de *server* en we spreken van een *client/server* paar. (NB. het begrip client/server heeft in een andere context een wat andere betekenis, al is de basisgedachte wel dezelfde.)

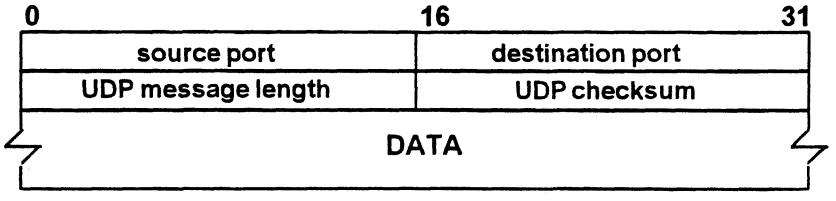
Zoals in de stad de winkels (servers) een duidelijk uithangbord hebben, terwijl de woonhuizen (de clients) hooguit een bescheiden naambordje hebben, zo moeten in een computernetwerk de servers door iedereen makkelijk gevonden kunnen worden, terwijl het adres van de clients er minder toe doet. Dat geldt ook voor poort-nummers.

Er zijn een aantal vast gedefinieerde *well-known port numbers*. Zo is in ieder computersysteem de Telnet server op poort 23 te vinden. Als een Telnet client (in opdracht van de gebruiker) contact zoekt met de Telnet server op een bepaalde computer, stuurt hij zijn bericht naar poort 23. De Telnet server stuurt het antwoord terug naar het poortnummer van de afzender, dat meestal willekeurig gekozen is.

IIR TECHNOLOGY

3.2 UDP

- **Source en destination port: *van* welke client, *naar* welke server en vice versa**
- **Length is aantal bytes, incl. data**
- **UDP checksum wordt berekend over de data èn de *pseudo header***



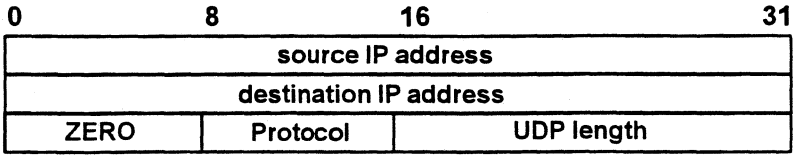
De UDP header is relatief eenvoudig. Als eerste de source en destination ports. Hiermee worden de applicaties geïdentificeerd die aan het communiceren zijn: dus *van* FTP client *naar* FTP server (en vice versa). De portnummers zijn het (de)multiplexing element in UDP (en ook in TCP, zoals uit de volgende paragraaf zal blijken).

Na de portnummers volgen UDP message length en UDP checksum. Dit zijn beide optionele velden en hoeven dus niet ingevuld te worden. 'Niet ingevuld' betekent in dit geval de waarde '0'.

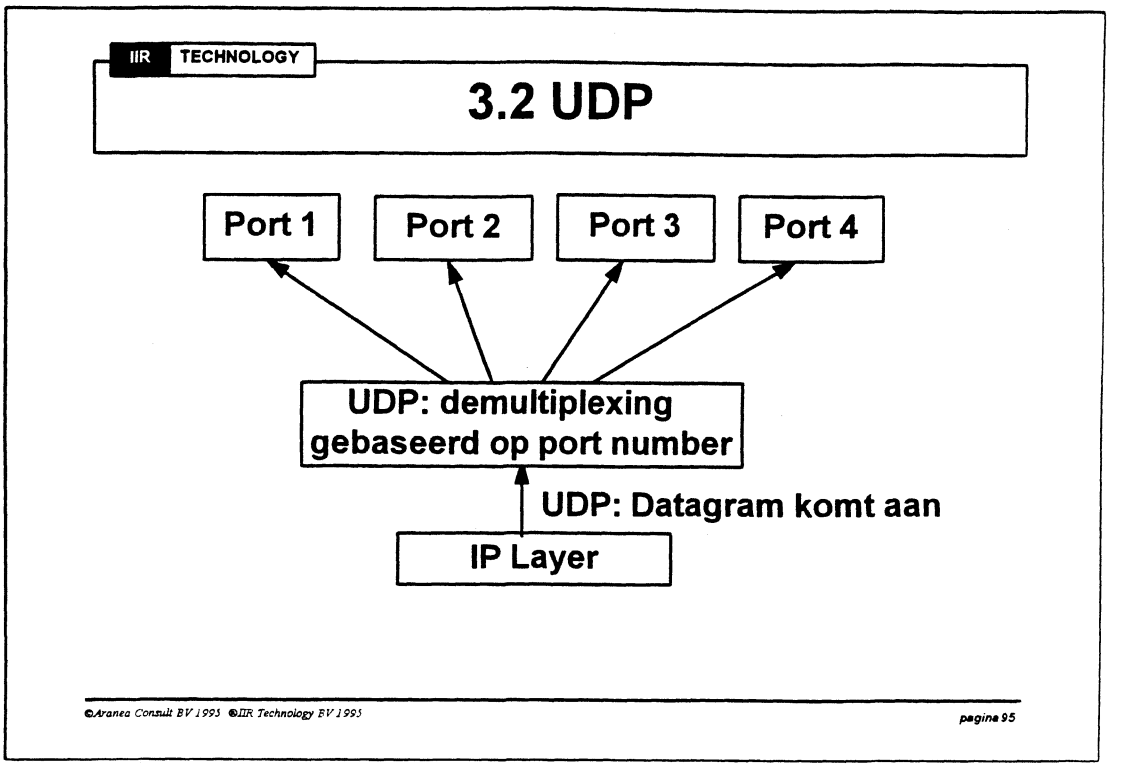
IIR TECHNOLOGY

3.2 UDP

- **UDP checksum over datagram èn pseudo header:**
  - **IP source en destination adressen**
  - **Protocol = IP protocol (17 voor UDP)**
  - **UDP length = de lengte UDP datagram**
- **UDP laag is dus *verweven* met IP laag**



De checksum bij UDP is optioneel, en hoeft dus niet berekend te worden. Als de checksum wèl berekend wordt, dient er gebruik te worden gemaakt van een zogenaamde *psuedo header*. Deze bestaat uit de velden source IP address, destination IP address, protocol number en UDP length. Op die manier wordt getracht te garanderen dat het ook 'terecht' is dat dit pakket op deze host is aangekomen en door de UDP laag wordt afgehandeld. Consequentie hiervan is dat UDP 'verweven' is met de IP laag; er moet immers met IP adressen gewerkt worden.



UDP doet aan (de)multiplexing op basis van zogenaamde well known port numbers. Voor deze nummers geldt dat de clients een random gekozen getal gebruiken, de servers één van de well known port numbers (uitzondering is de BOOTP client, die well known port 68 gebruikt). De port numbers 1-1024 zijn gereserveerd voor de, zeg maar, 'generieke' TCP/IP applicaties. Clients èn applicatiebouwers mogen hier *geen* gebruik van maken!
Onderstaand een lijst met well known ports die door UDP worden gebruikt.

Decimal	Keyword	Description
7	ECHO	Echo
9	DISCARD	Discard
11	USERS	Active Users
13	DAYTIME	Daytime
15	NETSTAT	Who is up or NETSTAT
17	QUOTE	Quote of the day
19	CHARGEN	Character generator
37	TIME	Time
42	NAMESERVER	Host Name Server
43	NICNAME	Who Is
53	DOMAIN	Domain Name Server
67	BOOTPS	Bootstrap Protocol Server
68	BOOTPC	Bootstrap Protocol Client
69	TFTP	Trivial File Transfer
79	FINGER	Finger
111	SUNRPC	SUN Remote Procedure Call
123	NTP	Network Time Protocol

IIR TECHNOLOGY

3.3 TCP

- **TCP: Transmission Control Protocol**
- **Transport protocol 'met alles er op en er aan':**
  - **sessie opbouwen en afbreken**
  - ***full-duplex***
  - **buffering**
  - ***error checking* en correctie**
  - **flow control**
  - **dynamische bepaling van *timeout* waarde**
- **overeenkomst met UDP: well known port numbers (demultiplexing element binnen TCP)**

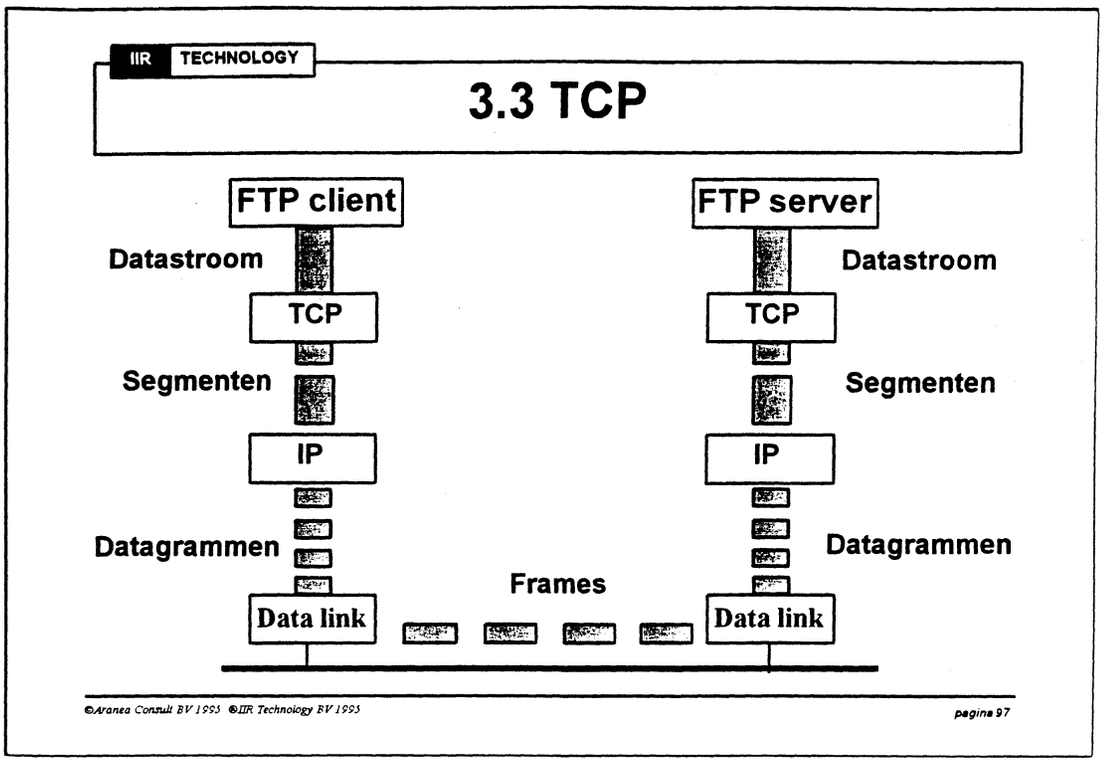
Transmission Control Protocol (TCP) is het 'grote' transport protocol van TCP/IP. Het gebruikt port nummers op dezelfde manier als UDP.

TCP biedt zijn klanten (bv. Telnet, FTP) een zeer complete service: een transparante 'pijp' waar zij een *stream* bytes doorheen kunnen sturen, met de garantie dat die ongeschonden aan de overkant komt. TCP zorgt zelf voor het formeren van *segmenten* (zo worden de berichten die TCP - via IP - naar zijn peer-TCP in de andere machine stuurt, genoemd). Een TCP verbinding is full-duplex, kan dus tegelijkertijd in beide richtingen worden gebruikt.

Als de ontvanger achterop loopt met het verwerken van de verstuurde data, zorgt TCP automatisch dat er niet te veel data wordt verstuurd: flow control is ingebouwd.

Als IP een frame 'verliest', merkt TCP dat, omdat hij niet op tijd een ontvangstbevestiging van zijn partner krijgt. De ontbrekende data worden dan opnieuw verstuurd. Maar wat is 'op tijd'? In een LAN komt de ontvangstbevestiging (*acknowledgement* of kortweg *ack*) binnen enkele milliseconden, bij een satellietverbinding kan het een volle seconde duren!

TCP meet voortdurend hoe lang het duurt voor de ack's binnen komen en stelt daar zijn *timeout* op in. Zelfs bij sterk wisselende netwerkbelasting (en dus sterk wisselende netwerkvertraging) zorgt dit mechanisme voor een optimale timeout waarde: niet te kort (dat leidt tot vals alarm), niet te lang (dan duurt het te lang voor een verloren gegaan frame wordt nagezonden).



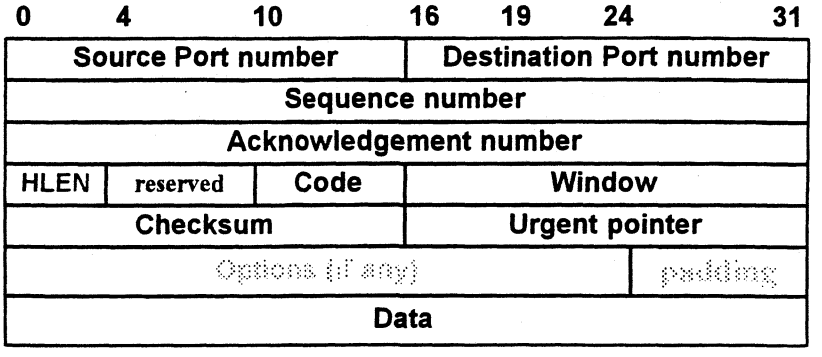
Ook TCP maakt gebruik van port nummers waardoor clients en servers met elkaar kunnen communiceren. Onderstaand een lijstje van de well known ports die TCP gebruikt. Clients en applicatiebouwers mogen géén gebruik maken van deze port nummers.

Decimal	Keyword	Description
5	RJE	Remote Job Entry
7	ECHO	Echo
9	DISCARD	Discard
11	USERS	Active Users
13	DAYTIME	Daytime
15	NETSTAT	Who is up or NETSTAT
17	QUOTE	Quote of the day
19	CHARGEN	Character generator
20	FTP-DATA	File Transfer (Data)
21	FTP	File Transfer (Control)
23	TELNET	Telnet
25	SMTP	Simple Mail Transfer
37	TIME	Time
39	RLP	Resource Location Protocol
42	NAMESERVER	Host Name Server
43	NICNAME	Who Is
53	DOMAIN	Domain Name Server
79	FINGER	Finger
109	POP-2	Post Office Protocol version 2
111	SUNRPC	SUN Remote Procedure Call
115	SFTP	Simple File Transfer Protocol
119	UNTP	USENET News Transfer Protocol
160-223	Reserved	

IIR TECHNOLOGY

3.3 TCP

- **PDU beschrijving van TCP:**



Code bits: URG, ACK, PSH, RST, SYN, FIN

Source port number - port nummer van de applicatie die het TCP segment verstuurd heeft.

Destination port number - port nummer van de applicatie waar dit segment voor bestemd is.

Sequence number - 32 bits integer die aangeeft met het hoeveelste byte, uit de stream van bytes die verzonden worden, dit segment begint.

Acknowledge number - getal dat aangeeft welk byte het volgende byte is dat verwacht wordt (meegeven ACK bij *zenden* van data: *piggy backing*).

HLEN - lengte van de header.

code bits - aantal bits waarmee middels 0 of 1 kan worden aangegeven dat bepaalde velden valide zijn cq. er iets verwacht wordt van de ontvanger:

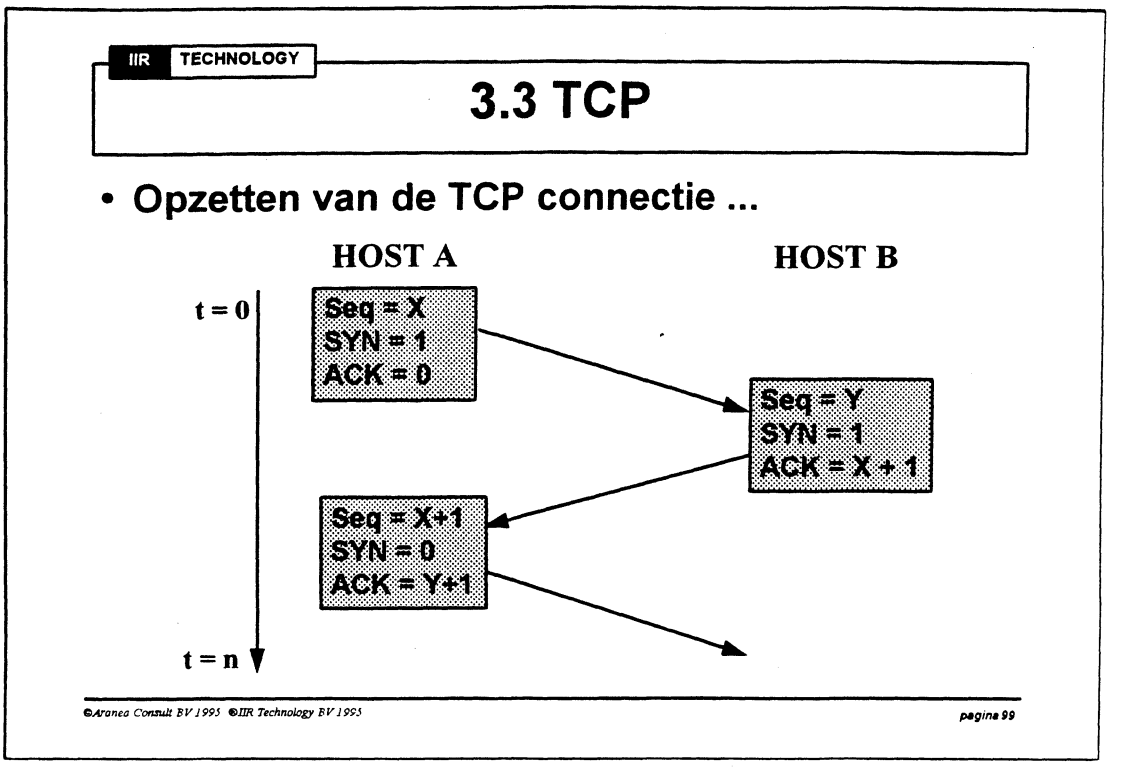
- URG- urgent pointer valid
- ACK- acknowledgement value valid
- PSH - dit segment svp als eerste afhandelen, voor andere processen
- RST - reset van connection (bijvoorbeeld bij aanvraag niet bestaande port)
- SYN - synchronize sequence nummers: afspreken initiële sequence nr's.
- FIN - final bits, afsluiten connectie

Window size - hoeveel bytes kunnen verwerkt worden alvorens een ACK uitgestuurd moet worden (*window advertisement*). Geeft bufferruimte aan!

Checksum - TCP checksum, inclusief pseudo header (zie beschrijving van UDP voor een verklaring van het begrip 'pseudo header').

Urgent pointer - een gedeelte uit de stream voorrang verlenen aan de ontvangende kant.

Options - optionele parameters. Vrijwel altijd gebruikt tijdens opzetten connectie: MSS (Maximum Segment Size) geeft aan hoe groot de ontvangende buffer kan zijn. Is enigzins gerelateerd aan MTU: bij opzetten van connectie over routers heen, is default MSS van 536 en default MTU dus 576!

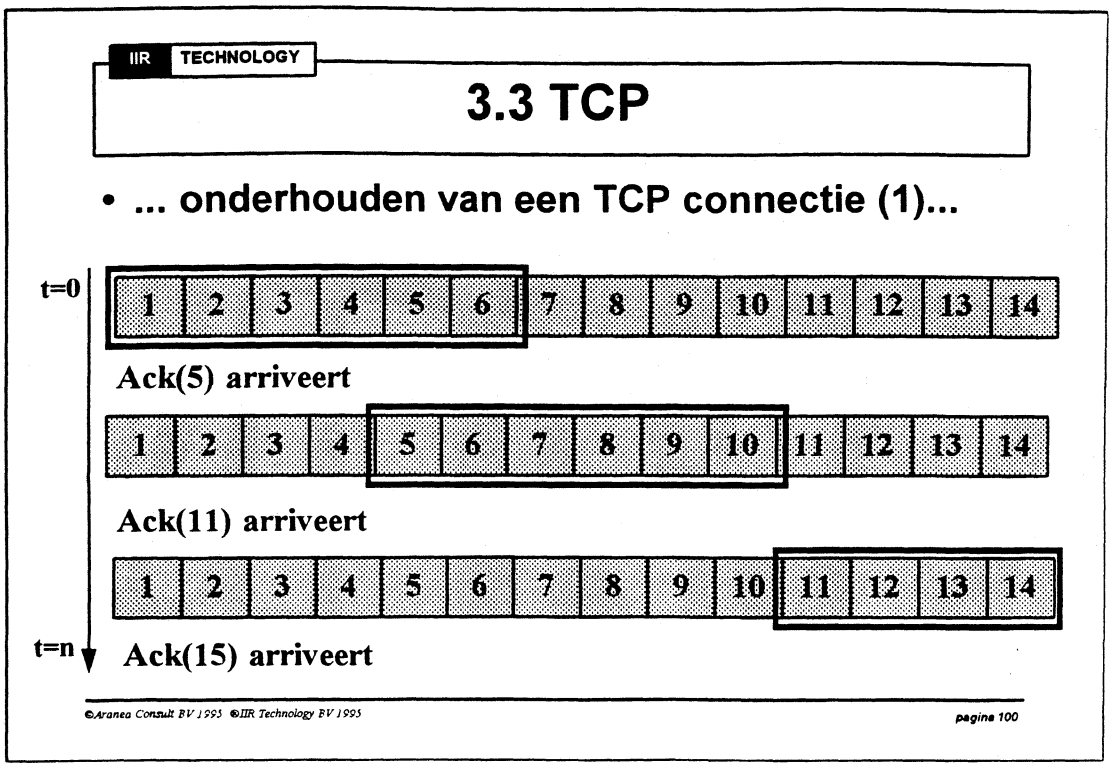


Het opzetten van een TCP connectie gebeurt middels de zogenaamde 'three way handshake'. Op de slide is te zien dat er inderdaad drie pakketten nodig zijn om een TCP connectie te openen.

Host A, in dit geval de initiator van de TCP connectie, stuurt een (leeg) TCP segment naar host B. Dit segment bevat het (initiële) sequence nummer X, het SYN bit dat op 1 staat en het ACK bit dat op 0 staat (daarmee aangevende dat er met dit segment niets bevestigd wordt!). Alleen bij SYN pakketjes kan een MSS optie meegestuurd worden.

Host B ontvangt dit segment, is bereid een connectie op te zetten met host A en stuurt dus een segment terug. Dit segment bevat het initiële sequence nummer dat host B van plan is te gebruiken, alsmede een SYN bit met waarde 1, *èn* een acknowledge op het X+1 byte van host A. Hiermee geeft host B te kennen dat het volgende byte dat verwacht wordt van host A, het byte X+1 is, gerekend vanaf het initiële sequence nummer X. Merk op dat bij het opzetten van de verbinding er vanuit wordt gegaan dat, ook al worden er eigenlijk niet èchte data-bytes verstuurd, het SYN bit één byte aan data representeert!

Host A ontvangt dit segment en dient vervolgens het SYN bit van host B te bevestigen. Host A stuurt derhalve een acknowledgement terug, met acknowledgement nummer Y+1 (immers, het SYN bit van B heeft, virtueel, één byte data opgeleverd, dus het Y+1 byte is het volgende byte dat A verwacht), het ACK bit zal op 1 staan.



Een tweetal parameters is van belang bij het uitwisselen van bytes in een TCP connectie. Enerzijds is dat de window size, anderzijds het ACK nummer.

Als twee hosts middels TCP gaan communiceren, wordt daarbij een window size afgesproken. Dit is een variabele die gedurende de sessie aan verandering onderhevig kan zijn. De window size geeft aan, hoeveel bytes een zendende machine 'uit heeft staan'; het mogen er nooit meer zijn, want de andere kant moet dan in de gelegenheid worden gesteld een ACK te sturen op die uitstaande bytes!

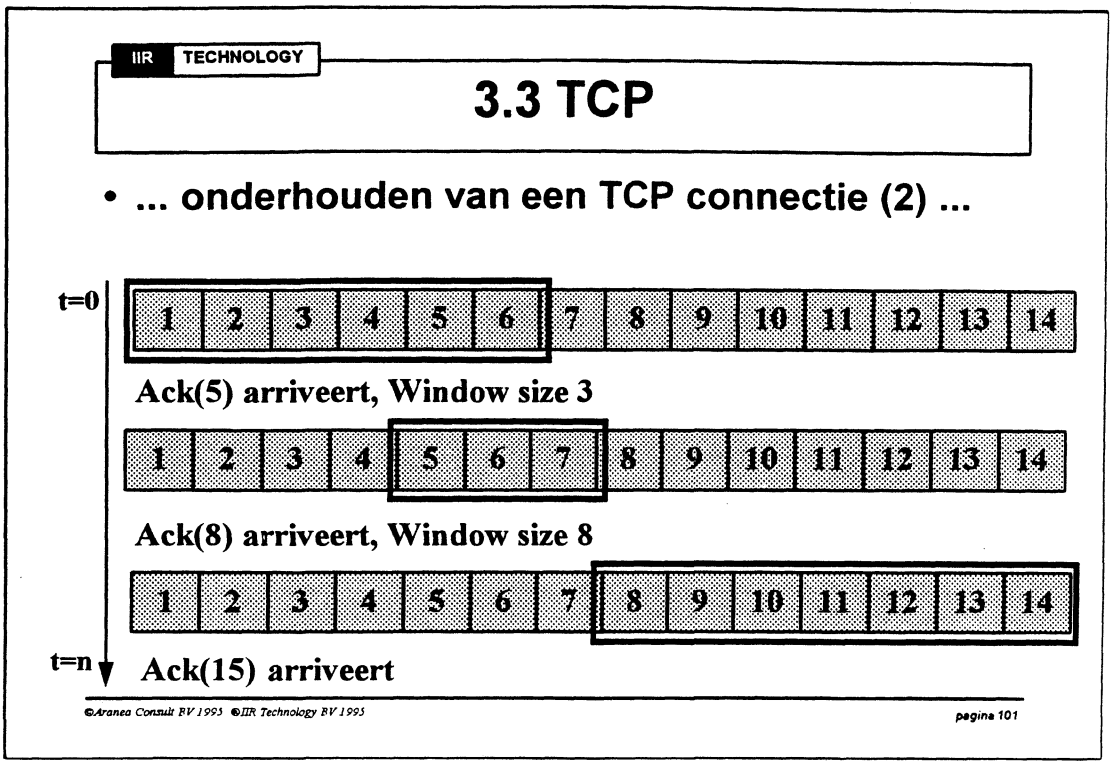
In het voorbeeld op de slide is de window size 6, en zal deze tijdens de connectie ook niet veranderen. De zendende machine heeft 14 bytes klaar staan om te verzenden, en zal beginnen met het verzenden van de eerste 6 bytes. Vervolgens zal de zender moeten wachten op een ACK.

Die ACK komt, met waarde 5. Er zijn diverse schema's voor het bevestigen cq. ontkennen van de ontvangst van data. Het schema dat TCP gebruikt, is dat er bevestigd wordt welke het volgende te ontvangen byte dient te zijn. Met andere woorden: een ACK 5 betekent dat het volgende verwachte byte, het 5e byte is. Daarmee wordt *expliciet* de goede ontvangst van bytes 1-4 bevestigd! Blijkbaar zijn byte 5 en byte 6 echter niet goed aangekomen (TCP draait boven op het onbetrouwbare IP: wellicht zijn bytes 5 en 6 in een apart IP datagram verstuurd en is dit datagram onderweg verloren gegaan!).

Omdat er een ACK op byte 5 is gekomen, kan het window van de zendende host opschuiven (vandaar vaak de term *sliding window* protocol) en kunnen bytes 5 tot en met 10 uitgezet worden. Vervolgens zal de zender weer moeten wachten op een ACK op deze uitstaande bytes.

Na enige tijd arriveert er een ACK op byte 11. Blijkbaar heeft de ontvanger bytes 1-10 inmiddels goed ontvangen en is het volgende verwachte byte, byte nummer 11.

De zender zal nu de laatste bytes van de set klaarstaande bytes verzenden.



Op deze slide een voorbeeld hoe met behulp van de window size aan flow control kan worden gedaan.

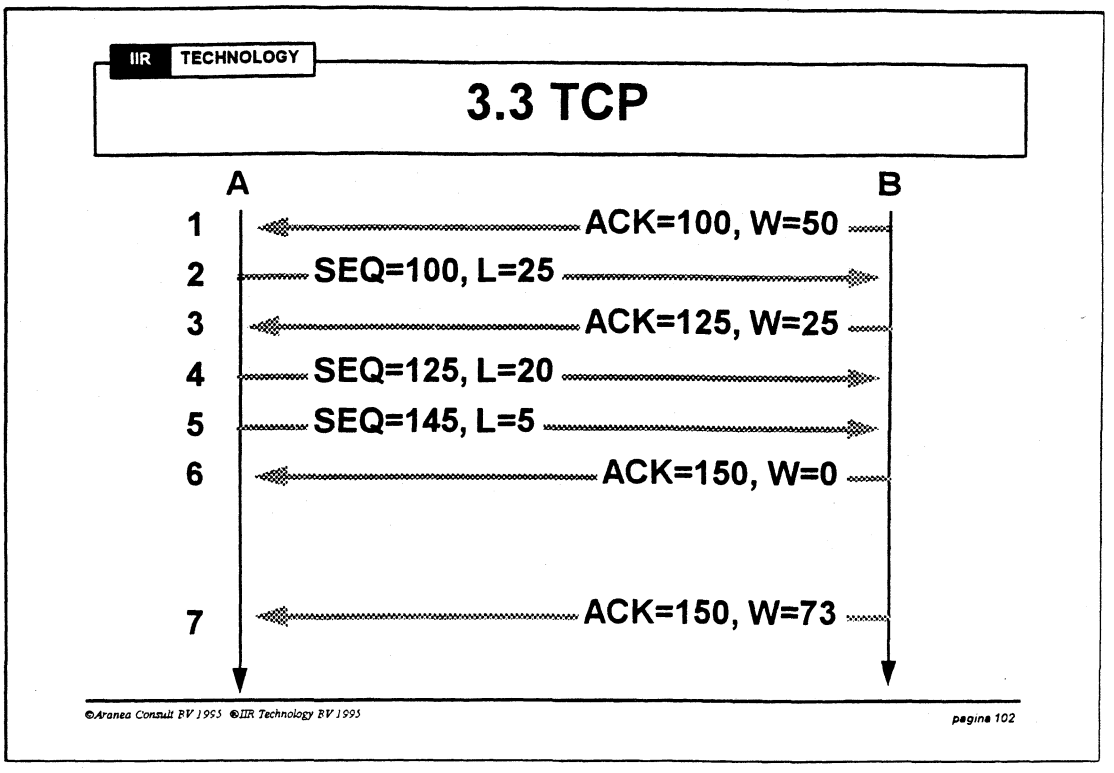
Bij het ACK van het eerste setje bytes dat verstuurd is, geeft de ontvanger ook een nieuwe window size door. Blijkbaar is de vorige window size van 6 naar beneden toe aangepast. De zender mag nu nog maar 3 bytes hebben uitstaan. Dus zullen de bij de tweede zending bytes 5, 6 en 7 verstuurd worden.

Als de ACK op deze bytes binnenkomt (ACK(8), dus blijkbaar zijn ze alledrie binnengekomen), wordt wéér een nieuwe window size geadverteerd, namelijk window size 8. De zender kan nu alle resterende bytes in één keer 'uitzetten', waarop ook inderdaad een ACK(15) volgt.

Mocht overigens een ACK niet binnen een bepaalde time out periode aankomen op de zendende machine (omdat de ACK, vervoerd over het onbetrouwbare IP, verloren is gegaan), dan zal de zendende TCP het hele window nóg een keer zenden.

Merk op dat bij de ACK's wordt aangegeven wat het volgende verwachte byte is. Hiermee wordt aangegeven dat alle daarvóór liggende bytes goed zijn ontvangen. Als er echter ook maar één byte gemist wordt, en alle daaropvolgende bytes wèl zijn aangekomen, zal dit toch een hertransmissie van vele, reeds aangekomen bytes, tot gevolg *kunnen* hebben! ('Intelligentere' implementaties van TCP zullen middels buffers de reeds goed ontvangen data proberen te bewaren).

TCP maakt gebruik van diverse timers. Ook deze timers zijn, evenals de Window size, dynamische parameters. Zo houdt TCP bijvoorbeeld rekening met de RTT (Round-Trip Time). Als deze erg hoog is, dan zal de zendende TCP meer tijd nemen om op ACK's te wachten. Een aantal wiskundige algorithmes (Nagle, Karn) wordt gebruikt om optimale time outs te bepalen.



In de figuur is het effect van het windowing gezien in de tijd, nog maar eens weergegeven. Windowing, puur bedoeld als flow control mechanisme.

Op tijdstip 1 wordt er een ACK ontvangen op pakketje 100, met een window size van 50. Met andere woorden, de ontvangende kant kan nog 50 pakketjes afwerken. De zender verstuurt vervolgens op tijdstip 2 een pakketje van 25 bytes. De terugmelding op tijdstip 3 geeft aan dat deze zijn aangekomen en er een window size van 25 overblijft. Op tijdstip 4 wordt een pakketje van 20 bytes verstuurd, op tijdstip 5 een pakketje van 5 bytes (immers, de ontvanger had nog ruimte voor 25 bytes). Dit resulteert in een ACK van deze beide pakketjes en een geadverteerde window size van 0. Immers, de ontvanger heeft geen ruimte meer om nog meer data te ontvangen. Na verloop van tijd, op tijdstip 7, zal de ontvanger eenzelfde ACK sturen, nu echter met een nieuwe window size van 73.

Ontvanger verwerkt niet direct:
kan aan bijv. opstarten MS-Word liggen (WIN
niet echt multitasking) en DLL's staan geen
interrupts toe. Bijv. DLL's zijn VxD's aan
te roepen (32 bit; interrupts toegestaan).

IIR TECHNOLOGY

3.3 TCP

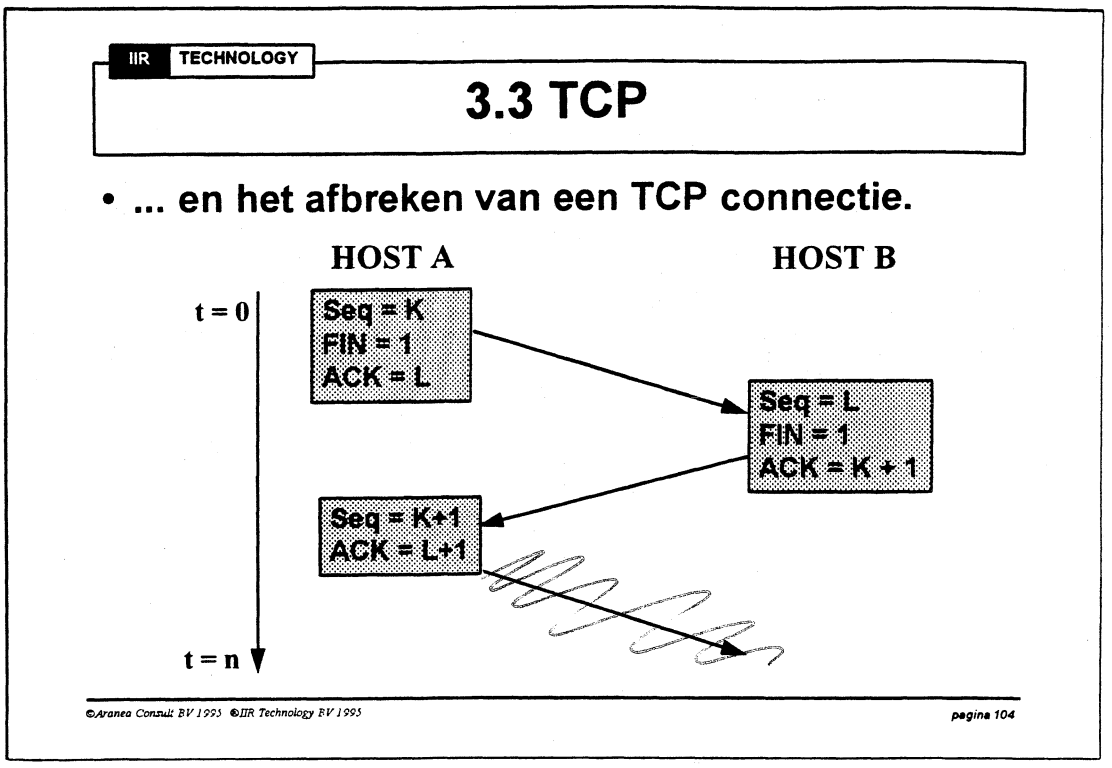
- **Slow-start mechanisme**
  - **Hiermee wordt voorkomen dat er veel data verloren gaat.**
  - **Maakt gebruik van algoritmen om de RTT te bepalen (time-out waarden), Kahn en Nagel.**
  - **Werkt met de opgegeven MSS.**
  - **Apart window bij zender om bij te houden wat er verzonden mag worden.**

Voor het slow-start mechanisme heeft de zender de beschikking over een zogeheten 'congestion-window (cwnd)'. Initieel is de cwnd zo groot als een enkel segment. Dit geeft het aantal bytes aan dat verzonden mag worden.

Voor iedere ACK die op tijd (afhankelijk van de RTT) terugkomt, wordt cwnd verhoogd met de segment grootte. Zodra er twee segmenten verstuurd zijn en beiden worden op tijd bevestigd, wordt cwnd voor iedere ontvangen ACK verhoogd met de segment grootte en dus vier. Het aantal in een window te verzenden segmenten groeit als alles goed gaat dus exponentieel. Uiteraard kan cwnd niet groter worden dan de geadverteerde window grootte van de ontvanger.

Zodra ACKs niet meer op tijd binnenkomen of dubbele ACKs arriveren, is er congestie in het netwerk wordt er dus te veel verkeer aangeboden. Met andere woorden: cwnd is te groot geworden. De cwnd wordt in dat geval op 1 gesteld en er vindt een slow-start plaats. In plaats van exponentieel te stijgen zal cwnd echter gedeeltelijk exponentieel groeien tot het aantal segmenten in cwnd ongeveer de helft is van het aantal segmenten waarbij congestie optrad. Vanaf dat punt zal cwnd lineair stijgen (congestion-avoidance).

Als er drie of meer ACKs van hetzelfde segment terugkomen, kan aangenomen worden dat het segment verloren is gegaan. Er is dan geen reden om een slow-start uit te voeren en te wachten op het aflopen van de time-out periode. Het segment kan direct opnieuw verzonden worden. Dit wordt fast-retransmit genoemd.



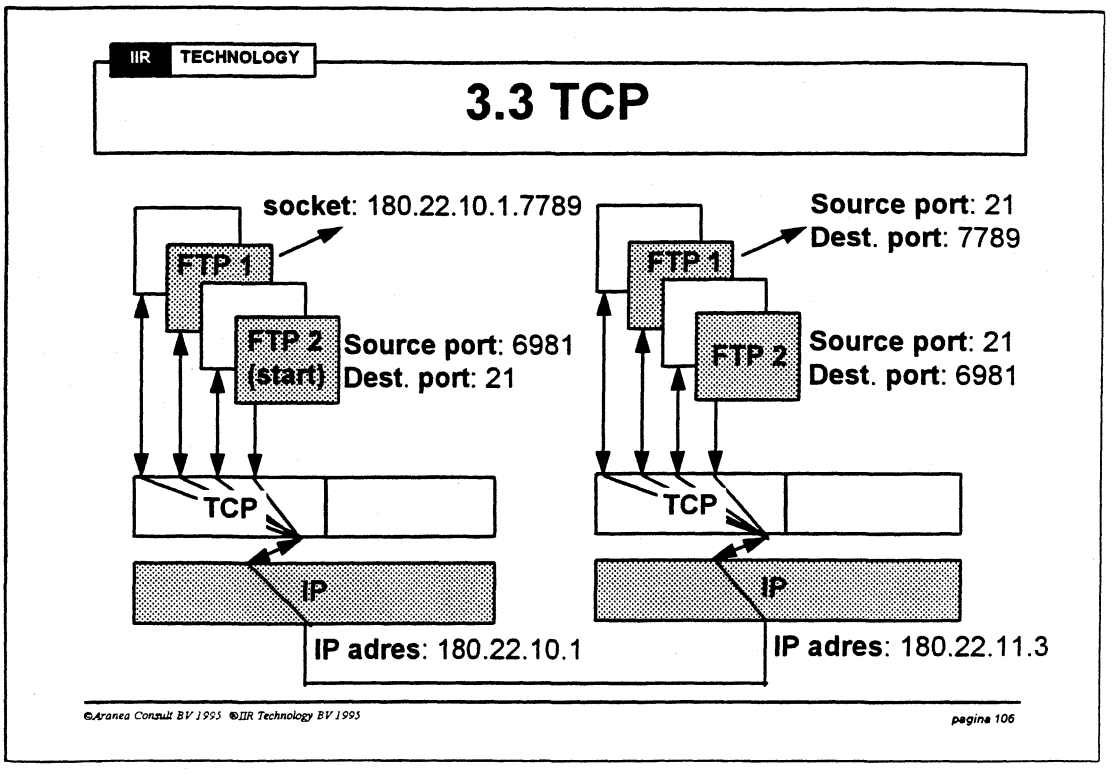
TCP connecties moeten uiteraard ook weer afgesloten worden. Daarvoor wordt gebruik gemaakt van het FIN bit. Op de slide is als voorbeeld gegeven dat beide hosts gelijktijdig hun connectie met elkaar afbreken. Dit hoeft echter niet! Daar waar op de slide host B een ACK terugstuurd op het FIN bit van host A, en daarmee ook maar meteen een FIN bit meestuurt, had host B er ook voor kunnen kiezen dit FIN *later* te sturen. In dat geval had het afsluiten van de TCP connectie uit twee stappen bestaan, van ieder twee segmenten: een FIN bit van A naar B met daaropvolgend een ACK (stap 1), en een FIN bit van B naar A met daaropvolgend een ACK (stap 2).

IIR TECHNOLOGY

3.3 TCP

- **Connecties tussen applicaties op hosts worden eenduidig bepaald door *socket* nummers**
- **Socket nr = IP nummer + port nummer**
- **De combinatie van de vier getallen (IP-port-IP-port) is ten alle tijde uniek identificerend!**

Om connecties tussen machines uniek te identificeren, wordt er gebruik gemaakt van zogenaamde socket nummers. Een socket nummer is de combinatie van een IP nummer/ portnummer op een machine. Door nu beide socket nummers van twee communicerende processen/machines achter elkaar te plaatsen, is daarmee uniek een connectie tussen twee communicerende programma's geïdentificeerd.



Op de slide een voorbeeld van IP nummers, port nummers en socket nummers. Kennelijk zijn er hier twee FTP sessies tussen twee dezelfde hosts. Het mag duidelijk zijn dat er uniek identificerende kenmerken nodig zijn om onderscheid te kunnen maken tussen beide sessies. Dit wordt dus gedaan met behulp van de socket nummers. De combinatie van de socket nummers maakt de connecties uniek!

Op machine A zijn twee sockets te zien, te weten 180.22.10.1.6981 en 180.22.10.1.7789.

Op machine B is slechts één socket nummer aanwezig, te weten 180.22.11.3.21 (de control port van de FTP applicatie: hier wordt nog op teruggekomen!).

Door nu de sockets van de twee machines achter elkaar te plaatsen, worden de beide connecties tóch uniek geïdentificeerd:

180.22.10.1.6981.180.22.11.3.21

180.22.10.1.7789.180.22.11.3.21

Diverse programma's zijn in staat om op basis van socket nummers allerlei statistische informatie bij te houden over TCP connecties. Het pakket LANWatch van FTP Software bijvoorbeeld heeft hier een aantal mogelijkheden voor.

IIR TECHNOLOGY

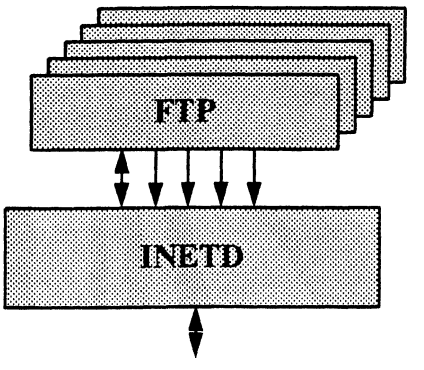
4. De applicatielaag

4.1 Algemeen
4.2 Telnet
4.3 FTP
4.4 SMTP
4.5 SNMP
4.6 NFS
4.7 X/Windows

IIR TECHNOLOGY

4.1 Algemeen

- **Passive open vs. active open**
- **INETD op Unix hosts om applicaties te starten**



Applicaties op het TCP/IP platform maken onderscheid tussen zogenaamde *passive opens* en *active opens*. Zoals besproken bij de transportprotocollen wordt er gebruik gemaakt van well known ports. Applicaties bevinden zich 'achter' dergelijke ports. Als een TCP/IP host gestart wordt, zullen de applicaties *passive opens* doen van deze well known ports. Met andere woorden, de Telnet applicatie (zie later) zal poort 23 openen, en gaat 'luisteren' of er aanvragen binnenkomen op deze port. Het zijn de clients die zogenaamde *active opens* doen. De clients proberen immers een Telnet sessie op te zetten, en dus een connectie te krijgen met port 23.

Als een TCP/IP host opstart, zullen er diverse TCP/IP applicaties een passive open gaan doen van hun well known port. Dit kan enige tijd in beslag nemen. Daarnaast betekent dit, dat er onmiddelijk een aantal system resources wordt gealloceerd, terwijl deze nog niet (echt) gebruikt worden (denk aan memory). Om dit te voorkomen, kent de Unix omgeving, maar inmiddels ook andere omgevingen, de Internet Deamon (INETD). Dit is de *enige* applicatie die bij system boot gestart wordt. Deze applicatie zal vervolgens op basis van de active opens van de clients de bijbehorende TCP/IP applicaties starten, en de client request doorsluizen naar de betreffende well known port. INETD haalt de informatie van de te starten host uit de configuratie file INETD.CONF.

IIR TECHNOLOGY

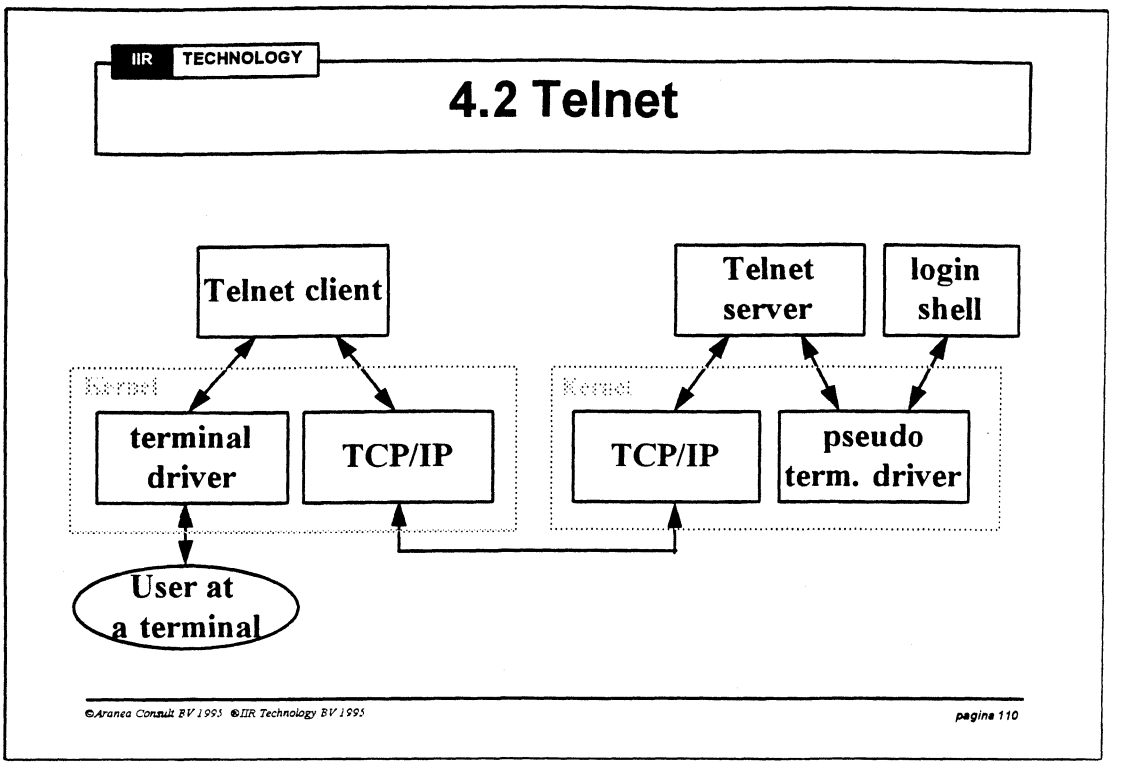
4.2 Telnet

- **Telnet: Telecommunications network protocol**
- **Terminal *protocol***
- **Terminal verkeer tussen hosts**
- **Network Virtual Terminal (NVT):**
  denkbeeldige terminal, toetsenbord en printer
  TELNET onderhandelt over NVT opties
- **Maakt gebruik van TCP well known port 23**
- **'Out of band' signaal met Urgent pointer**

Telnet is één van de twee virtual terminal protocols die beschikbaar zijn binnen TCP/IP. Het tweede protocol is Rlogin:

- Telnet is de standaard applicatie die in vrijwel iedere TCP/IP implementatie aanwezig is. Het kan gebruikt worden tussen hosts waarop verschillende operating systemen draaien. Met behulp van NVT negotiation (zie later) wordt er onderhandeld over de te gebruiken features.
- Rlogin is oorspronkelijk een Berkeley Unix applicatie die enkel bedoeld was om tussen Unix omgevingen te werken (maar inmiddels ook naar andere omgevingen geporteerd is).

Telnet is het oudste TCP/IP protocol en werd al gebruikt in het ARPAnet van 1969.



Uit bovenstaande figuur is een aantal interessante karakteristieken in verband met het Telnet protocol te herleiden:

- De Telnet client heeft zowel met een gebruiker achter zijn terminal als de TCP/IP protocollen te maken. Alles wat gebruikers intypen wordt via de TCP/IP connectie naar de andere kant verzonden en vice versa.
- De Telnet server heeft te maken met wat genoemd wordt een *pseudo terminal driver*. Hierdoor lijkt het voor het login programma, en alle andere programma's die onder de login shell draaien, alsof de client rechtstreeks met zijn terminal aan de betreffende host is gekoppeld.
- Er wordt gebruik gemaakt van slechts één TCP/IP connectie waarover zowel Telnet data (bijvoorbeeld het invoeren van een database) als Telnet controll informatie (wisselen van VT220 naar VT340 terminal emulator) wordt gestuurd. Er moet dus een manier zijn om hiertussen onderscheid te kunnen maken.
- In de figuur zijn licht-grijze kaders gebruikt om aan te geven dat de (pseudo) terminal drivers vaak geïntegreerd zijn in de TCP/IP kernel. De Telnet client en server zijn vaak separate programma's.
- Het mag duidelijk zijn dat Telnet naar een machine slechts dán zin heeft als er ook een login ID voor die machine beschikbaar is, aangezien alle Telnet verkeer door de login shell moet.

IIR TECHNOLOGY

4.2 Telnet

- **Telnet gebruikt NVT, de 'lowest common denominator' terminal**
- **NVT ASCII: 7-bit U.S. variant van de ASCII character set**
- **leder 7-bit character als 8-bit character verstuurd (aangevuld met '0').**
- **commando's worden voorafgegaan door 0xff : IAC (Interpret As Command)**

Om het Telnet commando zo algemeen mogelijk te houden, is gekozen voor de NVT ASCII terminal character set. Op deze manier is het mogelijk om Telnet te gebruiken op vrijwel ieder denkbaar platform.

Binnen Telnet wordt er onderscheid gemaakt tussen data die verstuurd wordt (en vrijwel rechtstreeks kan worden doorgegeven aan de login shell) en aan control data die de Telnet server moet interpreteren. Dit gebeurt middels het zogenaamde IAC byte (Interpret As Command). Na het IAC byte, dat wordt weergegeven door 0xff, volgt een command code. De volgende command codes zijn gedefinieerd:

Name	Code	Description
EOF	236	end-of-file
SUSP	237	suspend current process (job control)
ABORT	238	abort process
EOR	239	end of record
SE	240	sub option end
NOP	241	no operation
DM	242	data mark
BRK	243	break
IP	244	interrupt process
AO	245	abort interrupt
AYT	246	are you there?
EC	247	escape character
EL	248	erase line
GA	249	go ahead
SB	250	suboption begin
WILL	251	option negotiation
WONT	252	option negotiation
DO	253	option negotiation
DONT	254	option negotiation
IAC	255	data byte IAC

Als in de te verzenden data het IAC byte voorkomt, wordt dit onmiddelijk gevolgd door nóg een IAC byte (escape character).

IIR TECHNOLOGY

4.2 Telnet

- **Option negotiation gebruikt om afspraken te maken**
  - **WILL: de zender wil zelf een optie enablen**
  - **DO: de zender wil dat de ontvanger de optie enabled**
  - **WONT: de zender wil een optie zelf disablen**
  - **DONT: de zender wil dat de ontvanger disabled**
- **Option negotiation heeft 3 bytes nodig**
  - **IAC byte (commando byte)**
  - **één van de 4 negotiation bytes**
  - **ID byte van de betreffende optie**

Alvorens twee hosts met elkaar kunnen communiceren, dienen afspraken gemaakt te worden over wie, wat doet. Dit gebeurt op basis van de vier negotiation commando's WILL, DO, WONT, DONT. Belangrijk daarbij is de afspraak over hoe er met het verzenden van gegevens wordt omgegaan. Daarbij is één van de belangrijkste parameters wel het verschil tussen '*character at a time*' en '*line mode*'. Bij 'character at a time' wordt ieder door de user ingevoerd character door de client naar de server gestuurd. De server stuurt de Echo terug, die door de client wordt weergegeven op het beeldscherm van de user. Dit betekent per ingetypt character een pakketje van één byte met een totale header van 54 bytes op Ethernet! Dit is helaas de default van vrij veel implementaties (bijvoorbeeld de Solaris 2.2, SunOS4.1.3 en AIX 3.2.2 werken op deze manier). De elegantere oplossing is 'line mode'. Bij 'line mode' worden ingevoerde characters door de client weergegeven en wordt de complete line na een cr-lf naar de server gestuurd. Dit vermindert header overhead op het netwerk en bevordert de performance voor de gebruiker aan de client kant. Onderstaande tabel geeft een overzicht van een aantal codes waarover client en server kunnen onderhandelen alvorens de daadwerkelijke Telnet sessie kan beginnen (*niet* volledig!!!).

Name	Code	RFC	Betekenis
Transmit Binary	0	856	Change transmission to 8 bit binary (niet besproken in cursus)
Echo	1	857	Allow one side to echo data it receives
Suppress-GA	3	858	Suppress (no longer send Go-Ahead signal after data)
Status	5	859	Request for status of a TELNEToption from remote side
Terminal-type	24	884	Exchange information about the make and model of a terminal being used
End-of-Record	25	885	Terminate data sent with EOR code
Linemode	34	1116	Use local editing and send complete lines instead of individual characters

IIR TECHNOLOGY

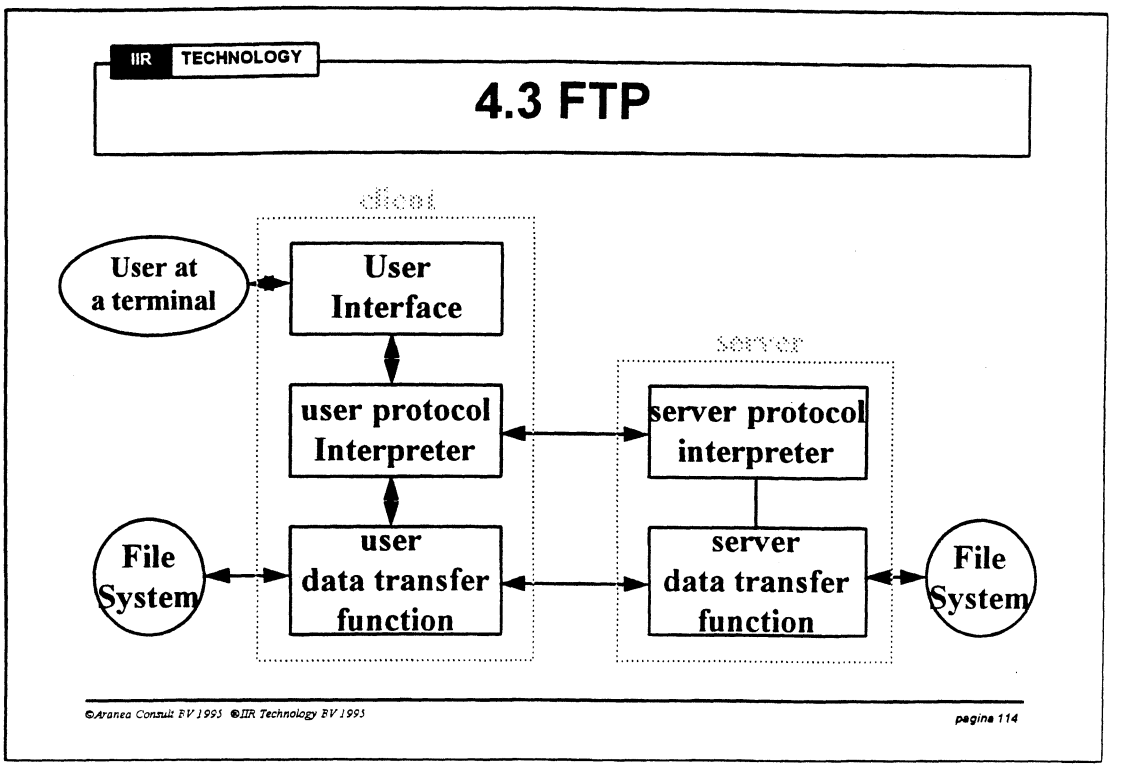
4.3 FTP

- **FTP: File Transfer Protocol**
- **Bestanden: ASCII óf binair**
- **Interaktief in gebruik**
- **Beveiliging: aanloggen met UserID/PassWord**
- **Uitzondering: anonymous FTP**
- **Gebruikt TCP**
- **Gebruikt *twee* well known ports: 20 en 21**

Het tweede applicatie protocol dat wijd verbreid gebruikt wordt, is het File Transfer Protocol. FTP kan gebruikt worden om hetzij ASCII bestanden, hetzij binaire bestanden te verzenden (dit dient door de gebruiker overigens expliciet aangegeven te worden!). FTP draait uiteraard op TCP: bij het overhalen van een file uit de Verenigde Staten wil de gebruiker natuurlijk kunnen uitgaan van een betrouwbaar transportprotocol.

Om toegang te krijgen tot een host om daar met FTP files vandaan te halen cq. naartoe te kopiëren, is op die host een UserID/PassWord vereist. Uitzondering hierop vormen de zogenaamde 'anonymous FTP sites'. Dit zijn servers waar gebruikers op kunnen aanloggen met het UserID 'anonymous'. Vervolgens krijgen de gebruikers een beperkte set van bestanden cq. bestandsgebieden ter beschikking. Anonymous FTP begint steeds meer in trek te komen, met name leveranciers die op deze manier patches en nieuwsbrieven de wereld in krijgen!

Een verschil tussen FTP en de andere applicatie protocollen is, dat FTP gebruik maakt van twee well-known ports.



Het FTP protocol wijkt enigzins af van datgene dat we tot dusver gezien hebben: het FTP protocol gebruikt twee connecties, een control connectie en een dataconnectie. De control connectie wordt gebruikt voor het versturen van allerlei control informatie van client naar server en vice versa. De dataconnectie wordt gebruikt voor:

- het zenden van een file van client naar server
- het zenden van een file van server naar client
- het zenden van een listing van files of directories van server naar client

De dataconnectie blijft niet constant open. Enkel dán wanneer één van bovenstaande datastromen verstuurd dient te worden. Het is de client die vervolgens een active open zal moeten doen van de betreffende well known port (20) op de server, waarbij hij zijn eigen port nummer zal moeten mededelen. Dit gaat met behulp van het PORT commando (zie ook verderop).

IIR TECHNOLOGY

4.3 FTP

- **FTP commando's in NVT ASCII**
- ***Alle* FTP commando's als 3/4 bytes uppercase ASCII character verstuurd**
- **Er zijn ongeveer 30 commando's gedefinieerd over de *control link* (well known port 21)**
- **Antwoorden van de server in de vorm van 3 digit cijfers, optioneel gevolgd door tekst**

Als je met een analyser op een netwerk zou kijken, zou je 'leesbare' commando's over de lijn zien gaan: de FTP commando's zijn allen gedefinieerd als redelijk begrijpelijke woorden van 3 of 4 letters. Vaak worden deze commando's één op één door gebruikers ingevoerd en doorgesluisd door de client applicatie naar de server applicatie. Het kan voorkomen dat één commando, door de gebruiker ingevoerd, leidt tot meerdere FTP commando's over de control link.

De antwoorden die terugkomen van de server bestaan uit codes, al dan niet aangevuld met verklarende teksten. De codes moeten overigens voldoende zijn voor de client om precies te weten wat de server bedoeld. Ook het electronic mail protocol uit de TCP/IP familie maakt gebruik van deze codes.

Onderstaande enkele voorbeelden van commando's en antwoorden:

Command	Description
ABOR	Abort previous FTP command and any data transfer
LIST filelist	list files or directories
PASS password	Password on server
PORT n1...n6	Client IP addres (n1...n4) and port (256*n5+n6)
QUIT	Logoff from server
RETR filename	Retrieve file ('get')
STOR filename	Store file ('put')
TYPE typespecify the file type: A(scii) or B(inairy)	
USER username	Username on server

Code	Description
125	Data connection already open; transfer starting
200	Command OK
214	Help message (for human reader)
425	Error writing file
500	Syntax error (unrecognised command)
501	Syntax error (invalid arguments)

IIR TECHNOLOGY

4.3 FTP

- **TFTP: *Trivial* File Transfer Protocol**
- **'klein broertje' van FTP**
- **Eenvoudig file transfer protocol**
- **Klein -> diskless workstations**
- **Gebruikt UDP**
- **Elk pakket wordt bevestigd met ACK**
- **Vaste blokgrootte: 512 bytes**
- **Geen beveiliging**

Het FTP protocol heeft een klein broertje, het TFTP protocol. Met name in die situaties waarin een kleine, snelle implementatie van een file transfer protocol benodigd is, wordt al snel gebruik gemaakt van het eenvoudige TFTP en niet van FTP.

TFTP draait boven op UDP (dat ook klein en snel is, en dus eenvoudig in een klein stukje code te implementeren is). Dit betekent dat er binnen TFTP zèlf enige error recovery en flow control aanwezig moet zijn. Dit is bewerkstelligd door middel van het versturen van een ACK(nowledgement) op ieder binnengekomen pakket. De gebruikte pakketgrootte is 512 bytes.

Hèt praktijkvoorbeeld van het gebruik van TFTP is in combinatie met het nog te bespreken BootP. BootP is een bootstrap protocol dat ingebakken kan zitten op (bijvoorbeeld) Ethernet kaarten. Met BootP kunnen opstartende machines 'vragen' aan daarvoor bestemde server hoe de bootimage heet. Vervolgens kan dit bootimage, met het kleine, snelle TFTP worden opgehaald.

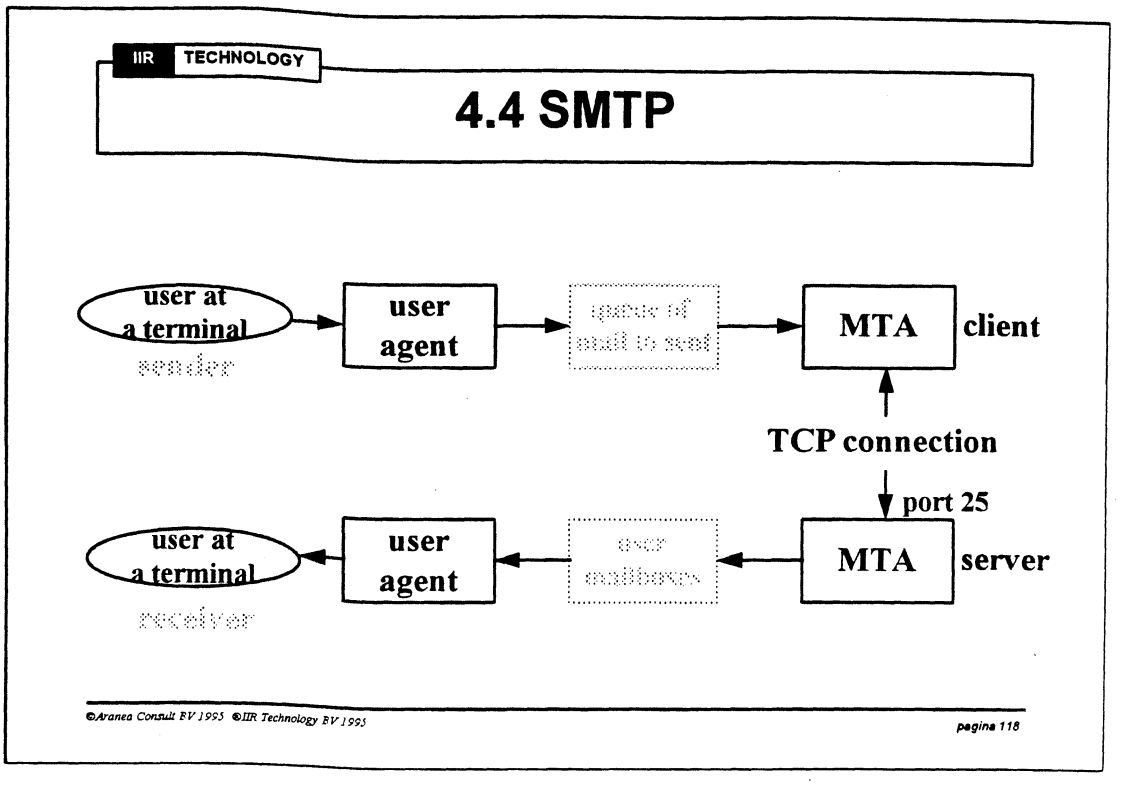
IIR TECHNOLOGY

4.4 SMTP

- **SMTP: Simple Mail Transfer Protocol**
- **Eenvoudig electronic mail protocol voor het verzenden van eenvoudige ASCII berichten**
- **Geen geavanceerde features als BCC of RR**
- **'Beroemde' RFC's 821/822**
- **De helft van alle TCP connecties over het Internet betreffen SMTP!**

Naast Telnet en FTP, het derde 'oude' TCP/IP protocol. Het is met recht een 'simple' electronic mail protocol te noemen. De gebruiker kan enkel een geadresseerde, een carbon copy houder, een onderwerp en een attachment (een mee te zenden file) definiëren, om vervolgens een 'plain text' bericht in te voeren en te versturen.

RFC821 beschrijft het SMTP protocol. Dat is het protocol dat twee MTA's gebruikt om via het Internet, over TCP met elkaar te communiceren. RFC822 beschrijft hoe SMTP berichten er mogen uitzien. Zie ook volgende slides.



Op de slide het electronic mail model dat binnen de TCP/IP protocol suite gebruikt wordt. Eindgebruikers hebben te maken met de user agents (front ends). Deze front ends, waarvan elm en Pine twee voorbeelden zijn, zorgen ervoor dat de te verzenden mailtjes in een aparte directory komen te staan (queue). De MTA (Message Transfer Agent) zorgt er vervolgens voor dat het mail bericht, over het Internet, verzonden wordt naar de MTA van de ontvangende partij. De Unix MTA heet Sendmail. Gebruikers hebben dus niets met een MTA te maken, die valt onder verantwoording van de systeem beheerder. Gebruikers hebben wellicht wel iets te zeggen over de user agents, aangezien het niet ongebruikelijk is dat er meerdere front end programma's beschikbaar zijn op een host.
De MTA's gebruiken NVT ASCII commando's onderling. Dit is een beperkte set van zo'n 10 commando's. Een aantal voorbeelden:

Mail commando	**Betekenis**
HELO	Testen of op de doel-machine een MTA draait (volledige domain name)
MAIL From: <...>	Afzender aangeven
RCPT To: <...>	Geadresseerde aangeven
DATA	Het daadwerkelijk te versturen bericht. Het einde van het bericht wordt aangegeven door een '.' op een aparte regel.
QUIT	Afsluiten van TCP connectie tussen MTA's
TURN	In dezelfde TCP sessie de zendende en ontvangende MTA van rol laten verwisselen
RSET	Mail uitwisseling onmiddelijk afbreken
VRFY	Ontvangende kant laten controleren of het ontvangst adres bestaat

Omdat er gebruikt wordt gemaakt van 3/4 byte NVT ASCII commando's, is het mogelijk om met behulp van een Telnet sessie naar een MTA op een host (telnet naar host èn well known port 25 wordt door veel implementaties ondersteund) mail te sturen zonder dat er een front end of een MTA op je eigen host draait!

IIR TECHNOLOGY

4.4 SMTP

- **Ontwikkelingen binnen SMTP**
  ESMTP: Extended SMTP (RFC1425)
  MIME: Multi-purpose Internet Mail Extensions (RFC1521)
- **Met name MIME begint vlucht te nemen op Internet**
- **Voorbeeld van MIME implementatie: *Pine, cc:Mail, Eudora, Lotus Notes***

Een aantal ontwikkelingen is zichtbaar binnen de Internet electronic mail wereld. Ten eerste het ontstaan van ESMTP, waardoor er ook niet NVT ASCII characters kunnen worden gebruikt in de header, en er een aantal extra opties gedefinieerd zijn. Eén van de opties is het EHLO commando, de opvolger van het HELO commando. Met het EHLO commando kunnen twee hosts bepalen of er al dan niet ESMTP wordt gebruikt. Het gebruik van non-ASCII characters in de header wordt mogelijk gemaakt door escape characters mee te sturen.

MIME werpt een geheel nieuw licht op Internet electronic mail. Met behulp van MIME is het namelijk mogelijk om ook non-ASCII parts mee te sturen in de body. Hiertoe heeft MIME een vijftal nieuwe header velden gedefinieerd:

MIME-Version:
Content-Type:
Content-Transfer-Encoding:
Content-ID:
Content-Description:

De huidige versie van MIME is 1.0. Het Content-Type TEXT/PLAIN specificeert het default Internet mail type. Momenteel zijn er 7 MIME content types gedefinieerd, ieder onderverdeeld in sub-types. Een aantal voorbeelden:

Content	Sub-type	Description
text	plain	unformatted text
	richtext	with simple formatting like **bold** and underline
multipart	mixed	multiple bodyparts, to be processed sequentially
	parallel	multiple bodyparts, to be processed parallel
image	jpeg	ISO 10918 format
	gif	CompuServe's Graphic Interchange Format
audio	basic	Encoded using 8-bit ISDN micro-law format
video	mpeg	ISO 11172 format

Er is inmiddels een aantal (client)implementaties beschikbaar, onder andere op het Sun platform.

IIR TECHNOLOGY

4.5 SNMP

- **SNMP: Simple Network Management Protocol**
- **De facto standaard voor management**
- **Ontstaan uit de behoefte om Internet routers te managen**
- **Initieel gezien als *voorloper* voor CMIP**
- **Ook geschikt om bovenop andere (dan UDP) transport protocollen te draaien**

Eén van de applicatie protocollen die inmiddels breed geaccepteerd zijn, is het SNMP protocol. Het wordt gebruikt om devices in netwerken te managen. In eerste instantie was het enkel bedoeld om de routers, in het steeds groter wordende Internet, te beheren. Later is het ook doorgetrokken naar andere kritische devices (bridges, hubs, systemen).

In eerste instantie is het protocol ontwikkeld om als tijdelijke oplossing te dienen. Binnen de ISO gemeenschap was men immers bezig met de ontwikkeling van het CMIP (Common Management Information Protocol) protocol, dat uiteindelijk *het* management protocol zou worden. Inmiddels is wel duidelijk geworden dat CMIP *niet* de standaard zal gaan worden waar het in eerste instantie op leek. Door de enorme penetratiegraad van SNMP is het ondenkbaar dat dit protocol zal migreren naar CMIP. Zeker met de komst van SNMPv2, de opvolger van SNMP die een aantal tekortkomingen zal opheffen, is er eigenlijk geen reden meer om CMIP te gaan gebruiken (behalve wellicht een aantal redenen die in een academische discussie genoemd zouden kunnen worden).

SNMP is afkomstig uit de TCP/IP wereld en is een applicatie die boven op UDP draait. De keuze voor het *on*betrouwbare UDP, en niet het betrouwbare TCP, is een bewuste keuze. Het SNMP protocol zal immers zèlf willen ervaren of er calamiteiten in het netwerk zijn, en niet een betrouwbare TCP laag, die met error recovery en flow control werkt, de mogelijke ellende laten afschermen!

IIR TECHNOLOGY

4.5 SNMP

- **Twee begrippen van belang binnen SNMP**
  - **SMI: Structure of Management Information**
  - **MIB: Management Information Base**
- **5 basale commando's:**
  - **GET**
  - **GET-NEXT**
  - **RESPONSE**
  - **SET**
  - **TRAP**
- **SNMPv2 kent 'extra' commando's**
  - **GET-BULK**
  - ~~INFORM~~

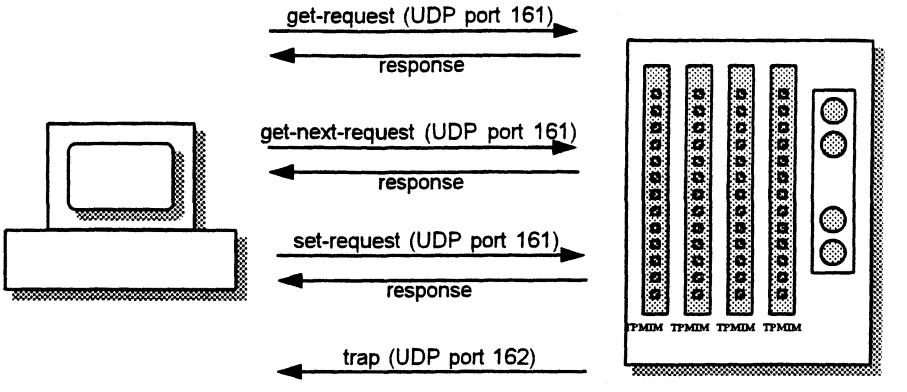
De 5 basale SNMP commando's worden in twee verschillende PDU's vervoerd. De GET, GET-NEXT, SET en RESPONSE worden in een PDU vervoerd met respectievelijke type aanduidingen 0, 1, 2 en 3. De TRAP's worden vervoerd in een aparte PDU met type 4.

Een GET wordt gebruikt om informatie op te vragen over een bepaalde variabele. Met GET-NEXT is het mogelijk om, beginnend vanaf een bepaald punt, een hele serie variabelen op te vragen aan een bepaald device. De RESPONSE is uiteraard het antwoord van het device waar de vraag aan gesteld. is. Met behulp van de SET parameter kan informatie worden weggeschreven in een bepaald device. Zo kan bijvoorbeeld een poort van een hub worden afgeschakeld (bijvoorbeeld met zoiets als '*SET PORT=0*').

IIR TECHNOLOGY

4.5 SNMP

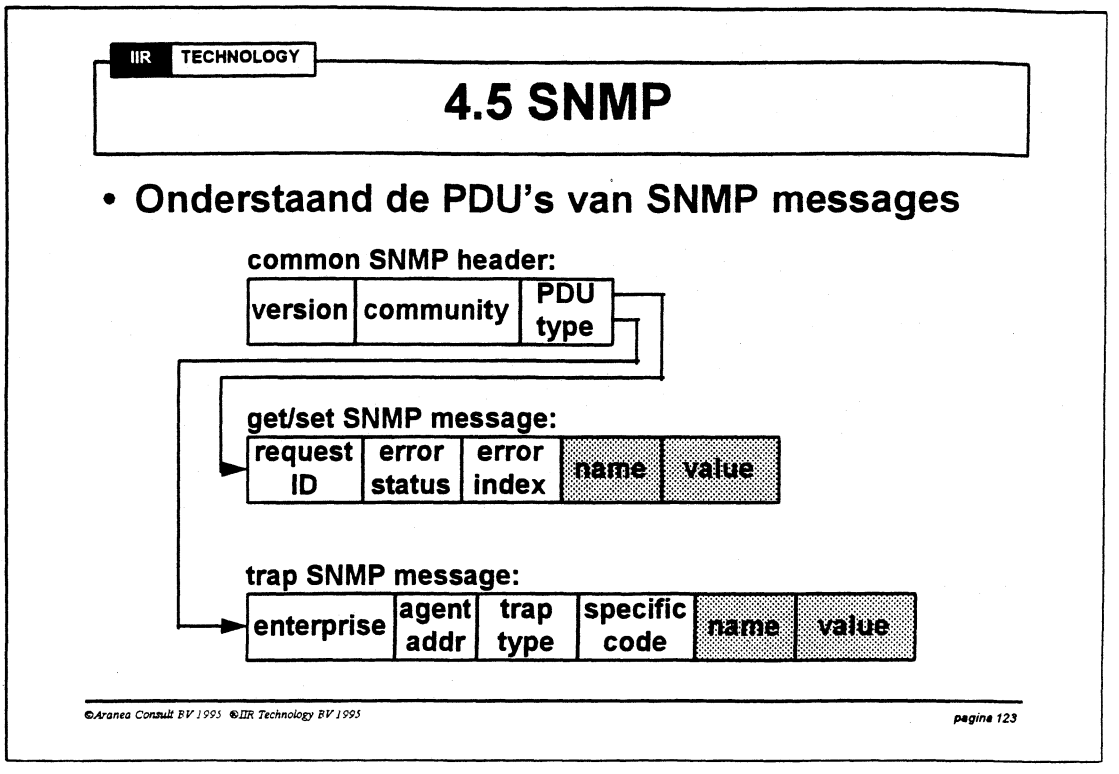
- **Onderstaand de uitwisseling van informatie:**



Bovenstaand een typische SNMP configuratie. Een SNMP management station wisselt informatie uit met een zogenaamde *SNMP agent*. In het voorbeeld is de SNMP agent een hub. Om precies te zijn, is de SNMP agent een stukje software (een apart board) dat op de hub draait.

Op de get-request van de SNMP manager zal de agent antwoorden met een response. Op de get-next-request (wat 'get-next' precies betekent wordt duidelijk als de MIB besproken wordt) zal de hub eveneens antwoorden met een response. En ook op een set-request zal de hub antwoorden met een response; op die manier heeft de SNMP manager inzicht in het resultaat van zijn set opdracht!

Er is één aparte PDU gedefinieerd, te weten de trap. Deze wordt 'spontaan' door de SNMP agent, in het voorbeeld de hub, verzonden naar de SNMP manager bij het 'afgaan' van een bepaalde threshold. Op die manier wordt de 'problematiek' een beetje verlegd van SNMP management station naar SNMP agent. Immers, de SNMP agent zal zèlf bepaalde waarden moeten gaan bijhouden en bij het overschrijden van die waarden een alarmmelding moeten genereren.



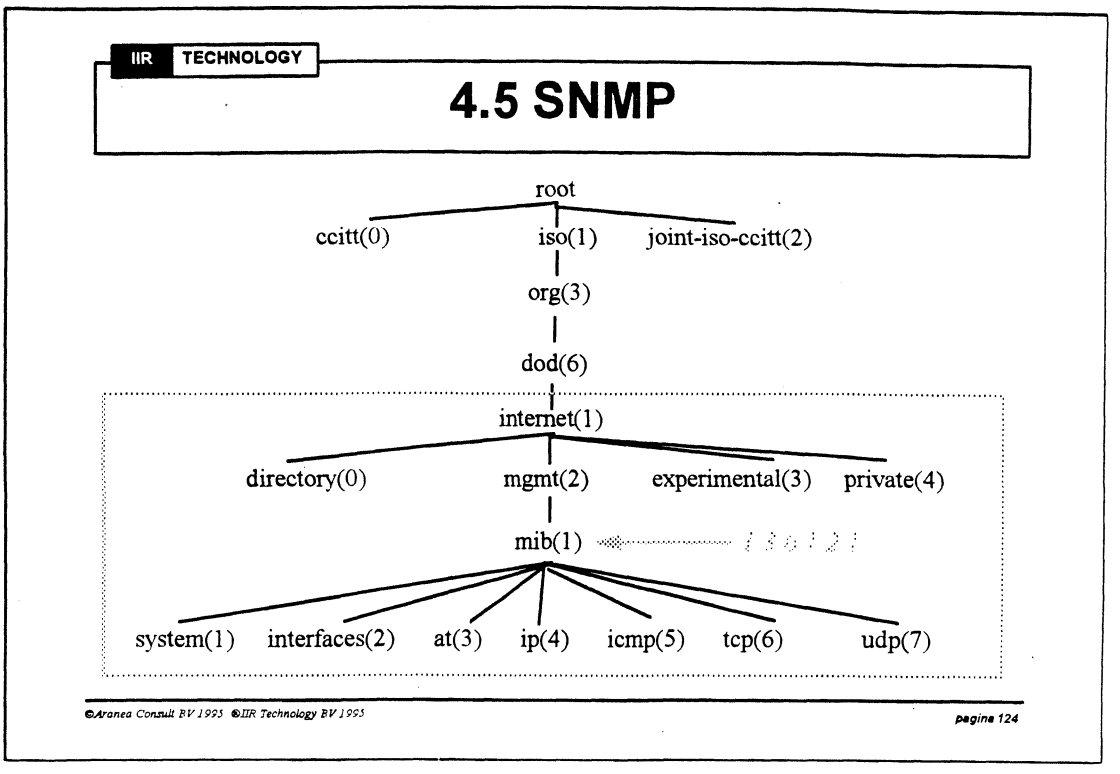
version - versie nummer van SNMP. Er wordt '0' gebruikt voor versie '1'!
community - het 'wachtwoord' dat manager en agent moeten gebruiken. Per community name kunnen rechten (RO of RW) worden toegekend.
PDU type - de 5 PDU types worden als volgt onderscheiden: 0 is get, 1 is get next, 2 is set, 3 is response en 4 is trap.
request ID - een random geselecteerd getal dat in de get, get-next en set wordt gebruikt door de manager, en door de agent in de response gebruikt wordt om vraag en antwoord te laten matchen.
error-status - wordt gebruikt in reply van SNMP agent. Zie tabel.
error-index - geeft aan op welke variabele de error betrekking heeft.
name/value - de eigenlijke SNMP variabelen en hun waarde (zie ook MIB).

error-status codes:

code	name	description
0	noError	All is ok
1	tooBig	agent could not fit reply into a single SNMP msg
2	noSuchName	operation specified a nonexistent variable
3	badValue	a set specified an invalid value or syntax
4	readOnly	manager tried to modify a Read-Only variable
5	genErr	some other error

trap types:

trap type	name	description
0	coldStart	Agent is initializing itself
1	warmStart	Agent is reinitializing itself
2	linkDown	An interface has changed from up to down
3	linkUp	An interface has changed from down to up
4	authentication Failure	A message was received from an SNMP manager with an invalid community
5	egpNeighborloss	An EGP peer has changed to the down state
6	enterpriseSpecific	look in the *specific code* field for information



Object-identifiers worden gebruikt om bladeren van de boom aan te duiden. Om de waarde van een blad van een boom op te halen, dient de Object-identifier te worden afgesloten met .0. Zo is de volgende object identifier de object-identifier voor de sysObjectId:

1.3.6.1.2.1.1.1

Om de waarde van deze system object identifier te weten, dient de manager dus de volgende vraag aan de client te stellen:

get 1.3.6.1.2.1.1.1.0

Sommige bladeren in de MIB zijn te bestempelen als tabellen. Zo is het bijvoorbeeld mogelijk dat in een bepaald device meerdere *interfaces* aanwezig zijn. Op dat moment moet de object-identifier niet worden afgesloten met .0, maar met het nummer van het betreffende interface. De volgende variabele duidt op de algemene variabele die het aantal inkomende bytes telt:

IfInOctets

Om nu te weten hoeveel bytes er zijn binnengekomen op een bepaald interface, zal dus het interface nummer moeten worden gespecificeerd:

IfInOctets.3

Als dit een eind-blad zou zijn geweest, had de variabele als volgt gebruikt moeten worden:

IfInOctets.0

De get-next operator is uitermate nuttig onder deze omstandigheden. Immers, het kan voorkomen dat je niet precies weet hoeveel interfaces er in een device aanwezig zijn. Door nu een aantal keren get-next te doen, krijg je alle interface informatie te pakken, en kom je vanzelf in een andere tak van de MIB tree terecht!

De private MIB's hebben gezorgd voor de enorme populariteit van SNMP. Immers, alle leveranciersdozen waren nu op afstand te managen!

IIR TECHNOLOGY

4.5 SNMP

- **'Open' SNMP managers**
  - **IBM NetView/6000**
  - **HP OpenView**
  - **NMC Vision**
  - (SNMPc)
- **'Gesloten' SNMP managers**
  - **CiscoWorks**
  - **Cabletron Remote LanView**

Er zijn zeer vele op SNMP gebaseerde netwerkmanagement stations beschikbaar momenteel. Deze variëren van eenvoudige MS DOS gebaseerde applicaties tot high end multi user multi tasking platformen (veelal op Unix).

Bedenk dat ***al*** deze applicaties gebruik maken van de 5 eenvoudige basis commando's die gedefinieerd zijn in SNMP! Onderscheid ligt natuurlijk in allerlei randverschijnselen, zoals wel of geen grafische presentatie, wel of niet importeren van private MIB's, wel of niet bijhouden van databases met historische gegevens, wel of niet aansturen van semafoon na afgaan van alarmen, etcetera.

Op de slide wordt een onderscheid gemaakt tussen wat genoemd wordt 'open' SNMP managers en 'gesloten' SNMP managers. Open SNMP managers kunnen betiteld worden als *platformen*, die basale SNMP functionaliteit bieden. Hier bovenop kunnen vendors add ons bieden waardoor hun devices beheerd kunnen worden.

De gesloten SNMP managers zijn min of meer stand alone applicaties, geheel geoptimaliseerd (met name in termen van grafische ondersteuning) voor één specifiek device.

IIR TECHNOLOGY

4.6 NFS

- **NFS: Network File System**
- **Oorspronkelijk van Sun Microsystems**
- **In '94 is versie 3 beschikbaar gekomen**
- **Gebaseerd op Sun RPC (remote procedure call, SunRPC gedefinieerd in RFC1057)**
- **Client krijgt, via RPC calls naar server, access op remote files**
- **Om transparantheid te garanderen tussen diverse OS'en wordt XDR gebruikt**

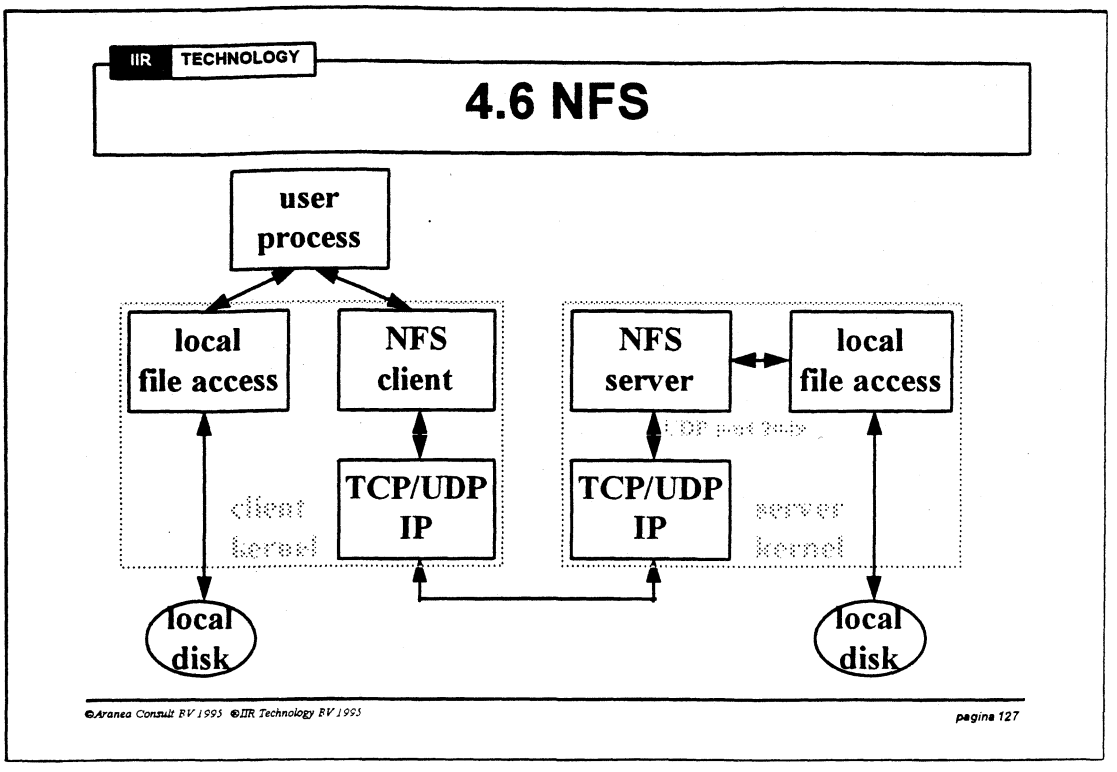
Geruime tijd heeft Sun haar NFS produkten proprietary gehouden, maar uiteindelijk zijn ze toch aangeboden en geaccepteerd als 'standard'. Eén van de meest bekende implementaties is nog steeds PC-NFS, het produkt van Sun, dat er voor kan zorgen dat MS-DOS gebaseerde PC's zich als NFS client kunnen gedragen (zie verder voor het verschil tussen client en servers).

NFS maakt gebruik van zogenaamde RPC's: remote procedure call's. Dit wijkt enigzins af van de NVT ASCII-achtige commando structuren die we tot dusver gezien hebben bij protocollen als FTP, Telnet en SMTP.

Er zijn twee versies van RPC. De eerste versie is gebouwd om bovenop TCP en/of UDP te draaien, en maakt gebruik van de sockets API. De tweede is gebouwd om bovenop de TLI API te draaien, het zogenaamde '*Transport Layer Interface*'. Dit is een kleine laag, die tussen de transport laag en de applicatielaag zit en het voor de applicatie laag transparant maakt wat er nu eigenlijk als transportprotocol gebruikt wordt.

Omdat NFS bedoeld is als protocol dat op diverse platformen moet kunnen draaien, zijn er speciale features ontwikkeld om dit te kunnen bewerkstelligen. Alle RPC call's worden altijd vertaald naar XDR (eXternal Data Representation) alvorens ze verzonden worden. XDR specificeert een soort 'esperanto' om RPC call's uit te wisselen. De ontvangende kant vertaalt het XDR formaat naar het native formaat, om het vervolgens uit te voeren.

NFS gaat uit van ephemeral ports (de tegenhanger van de well known ports). Om onderscheid te kunnen maken tussen de diverse client call's, wordt er gebruik gemaakt van een zogenaamde *port mapper*. De port mapper, die beschouwd kan worden als een RPC server programma, heeft zelf wèl een well known port nummer, te weten UDP port 111 en TCP port 111 (SunRPC is oorspronkelijk geschreven voor UDP, maar kan ook over TCP).



Bovenstaand een typische NFS configuratie. De volgende 6 stappen beschrijven de interaktie tussen de verschillende componenten:

- Voor de client is het transparant of er een local file wordt benaderd of een file op afstand. De kernel op de client houdt in de gaten of een file op de local disk is geopend, of remote. Vervolgens zal de kernel de requests doorgeven aan het *local file access* subsystem of aan de *NFS client*.
- De NFS client stuurt RPC requests naar de NFS server. Oorspronkelijk gebeurde dit louter over UDP, tegenwoordige implementaties kunnen ook over TCP werken.
- De server ontvangt de aanvragen op well known port 2049. Merk op dat dit in principe iedere andere willekeurige port had kunnen zijn (het komt immers van een leverancier af, alvorens het werd gestandaardiseerd!), echter juist *omdat* het van die leverancier komt, die oorspronkelijk van port 2049 uitging, zullen veel implementaties 2049 gebruiken.
- Omdat het kan voorkomen dat het benaderen van de betreffende file op disk enige tijd in beslag neemt, zijn de meeste NFS servers als multithreaded applicatie geïmplementeerd. Op die manier worden de aanvragen van meerdere clients 'gelijktijdig' niet in de wacht gezet.
- Ook voor de client (op Unix implementaties) geldt dat er meerdere instances van de betreffende programmatuur draaien om de doorvoer te versnellen.

NFS bestaat uit meer dan het NFS protocol zèlf, namelijk uit verschillende RPC programma's; *port mapper, NFS, mount, lock manager* en *status monitor*.

IIR TECHNOLOGY

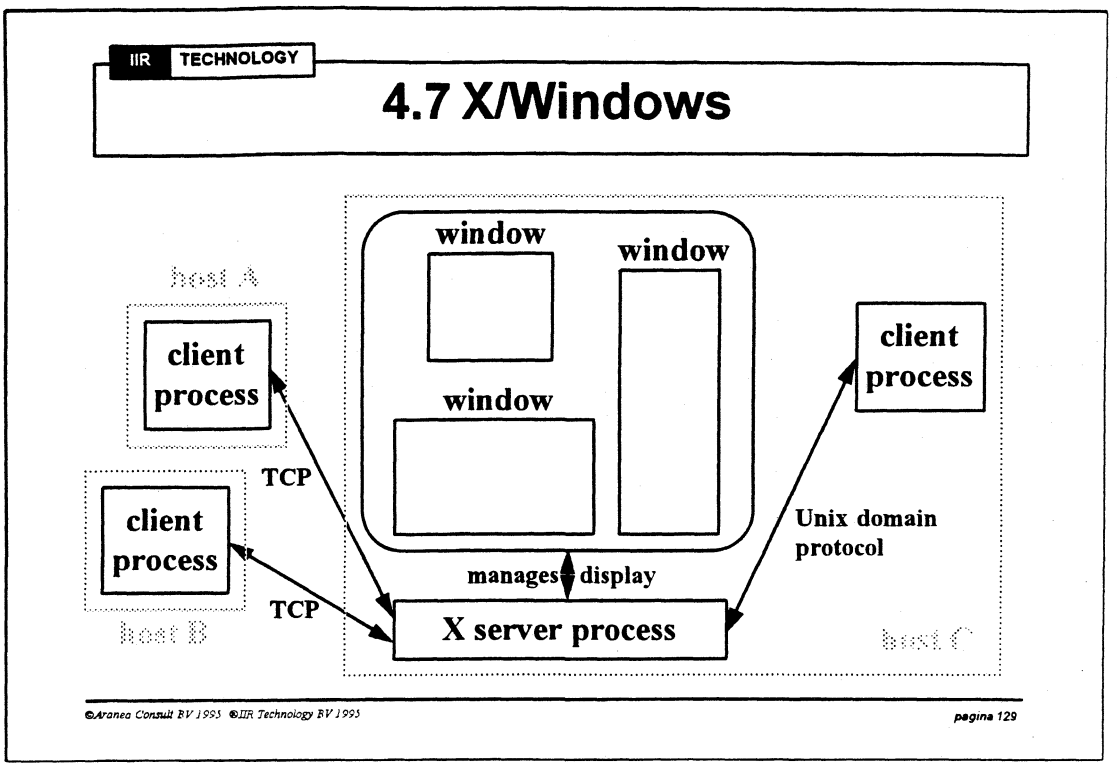
4.7 X/Windows

- **X/Windows als GUI**
- **Gestandaardiseerd in X/Open commissie**
- **Te beschouwen als *'display server'***
- **Client/server protocol, waarbij client en server zijn omgedraaid**

Telnet is het terminal protocol dat reeds besproken is. Met Telnet krijgt de gebruiker een character oriented user interface te zien, vaak als VT220 gedefinieerd. Er is echter ook een grafisch user interface beschikbaar. Dit is het zogenaamde X/Windows interface (of kort gezegd 'X'). X is een typische Unix applicatie en heeft derhalve ook 'iets minder' met TCP/IP te maken.

Ook X is een client/server applicatie, waarbij het server proces als 'display server' omschreven zou kunnen worden. Zo'n display server, die op een lokaal, intelligent werkstation draait, wordt 'aangestuurd' door een client, zijnde één of andere applicatie op een ander werkstation of een minicomputer. De applicatie beschrijft dan precies hoe het display opgebouwd moet worden, de X server op het lokale station zal het display verzorgen.

In totaal zijn er zo'n 150 commando's binnen X gedefinieerd.



Op de slide een voorbeeld van een omgeving waarin X gebruikt wordt. Op het (lokale, intelligente) werkstation draait een X server. Deze X server krijgt zijn input van lokale clients, waarschijnlijk lokale applicaties die hun output naar de X server schrijven, en van remote clients, waarschijnlijk applicaties op een remote host die hun output naar de lokale X server schrijven.

Als de lokale client naar het display schrijft, zal er gebruik worden gemaakt van Unix domain commando's ('transport'). Als de remote client naar het display schrijft, zal er gebruik worden gemaakt van TCP/IP.

Het is voor te stellen dat schermoutput informatie nogal grote netwerkbelasting kan veroorzaken. Er wordt dan ook onderzocht of het haalbaar is een 'kleiner' broertje van X te ontwerpen, of toch in ieder geval een versie die minder bandbreedte opeist. Dit wordt LBX genoemd, Low Bandwidth X.

IIR TECHNOLOGY

4.8 Diverse tools

- **Veelheid aan TCP/IP gebaseerde utilities:**

  NSLOOKUP [IP_address | host_name]
  PING <options> [IP_address | host_name]
  FINGER [username]@host_name
  TRACEROUTE <options> [IP_address | host_name]
  WHOIS | NICKNAME

IIR TECHNOLOGY

5. Routering

5.1 Algemeen
5.2 Direct routing
5.3 Indirect routing
5.4 Routing protocollen

IIR TECHNOLOGY

5.1 Algemeen

- **Diverse 'type' devices betrokken bij koppelen van netwerken**
- **Generieke term: 'internetworking'**
- **Onderscheid tussen *repeaters*, *bridges* en *routers***
- **Routers worden in de Internet wereld *Gateways* genoemd!** (in de cursus door elkaar gebruikt!)

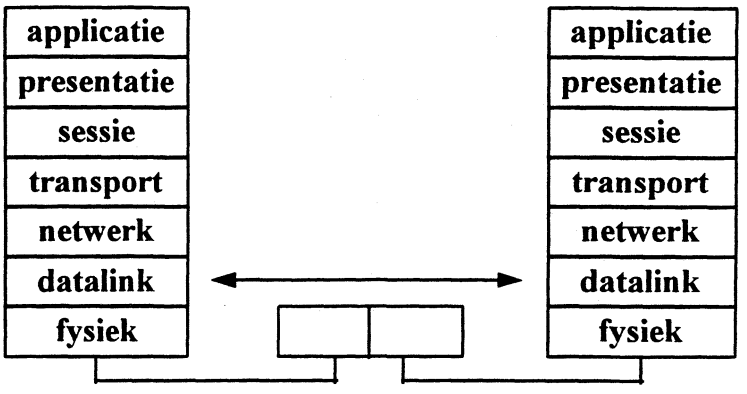
Het aan elkaar koppelen van netwerken kan op diverse lagen, gezien vanuit het OSI model, gebeuren. De generieke term voor de devices die gebruikt worden om netwerken aan elkaar te koppelen, is 'internetworking devices'.

Op laag 1, de fysieke laag, wordt gesproken over 'repeaters'. Op laag 2, de datalink laag, worden deze devices 'bridges' genoemd. Op laag 3, de netwerklaag, zitten de routers. Deze zullen de meeste aandacht krijgen in dit hoofdstuk, aangezien dit typische devices zijn waar het Internet vandaag de dag niet meer zonder kan! In het 'oude' Internet, worden de routers (min of meer abusievelijk) *gateways* genoemd. Houdt hier rekening mee bij het lezen van Internet literatuur!

IIR TECHNOLOGY

5.1 Algemeen - repeaters

- 'Repeaters' zijn eigenlijk 'regeneratoren'
- Low delay, eenvoudig te installeren



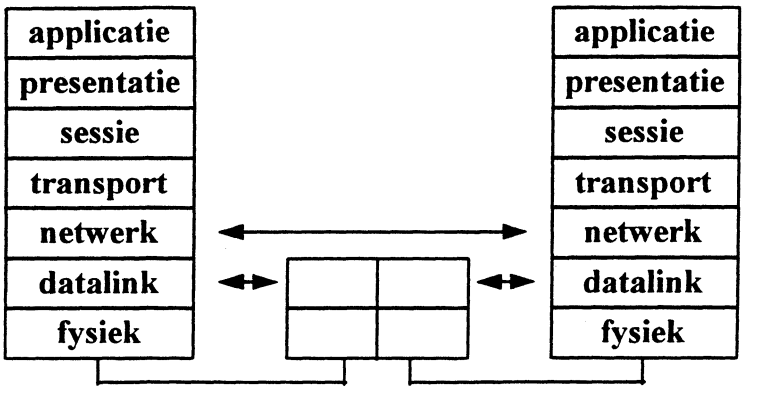
Op het laagste niveau, het niveau van de fysieke laag, hebben we te maken met repeaters. 'Repeater' is eigenlijk een verkeerde benaming. Het is namelijk juist niet de bedoeling dat er 'herhaald' wordt, maar dat er geregenereerd wordt. Een signaal dat langs een kabel verstuurd wordt, zal na verloop van tijd niet meer dat perfecte signaal van '0' en '1' symbolen zijn (-12V en +12V), maar enigzins vervormd zijn. Om de verzwakte/versterkte signalen weer netjes op niveau te krijgen, zijn regeneratoren nodig.

Regeneratoren zijn 'onzichtbaar' voor de datalink laag. Deze kijkt als het ware over de repeaters 'heen'. Het maakt een repeater ook niet uit wat er nu precies geregenereerd moet worden.

IIR TECHNOLOGY

5.1 Algemeen - bridges

- **Afhankelijk van de MAC laag (TRN of ETH)**
- **Als dedicated box of opgewaardeerde PC**



Bridges zitten een niveau hoger in het OSI model, namelijk op de datalink laag. Dat betekent dat er speciale Ethernet bridges en speciale Token-Ring bridges zijn. Immers, de datalink laag 'kijkt tegen de bridges aan', en heeft hier dus mee te maken!

Zowel de Ethernet wereld als de Token-Ring wereld kennen hun eigen specifieke bridges. In de Ethernet wereld wordt gebruik gemaakt van zogenaamde Spanning Tree bridges, ook wel Transparant bridges genoemd (werkend volgens het Spanning Tree Algorithm, gespecificeerd in IEEE 802.1d). De bridges zullen onderling ervoor zorgen dat er te allen tijde één, en precies één pad beschikbaar is van een willekeurige Ethernet machine A naar een willekeurige Ethernet machine B. Om dit te bewerkstelligen wordt gebruik gemaakt van het STA algorithme. Beheerders hoeven hier (amper) op in te grijpen. Bridges kunnen dit vrij zelfstandig uitvoeren. Een op Ethernet aangesloten host hoeft zich geen zorgen te maken over de (fysieke) topologie van het Ethernet netwerk. Ethernet hosts blijven gewoon hun pakketjes versturen en als er al bridges voorkomen in het netwerk, zullen die zèlf wel bepalen of ze iets met de betreffende pakketjes moeten doen.

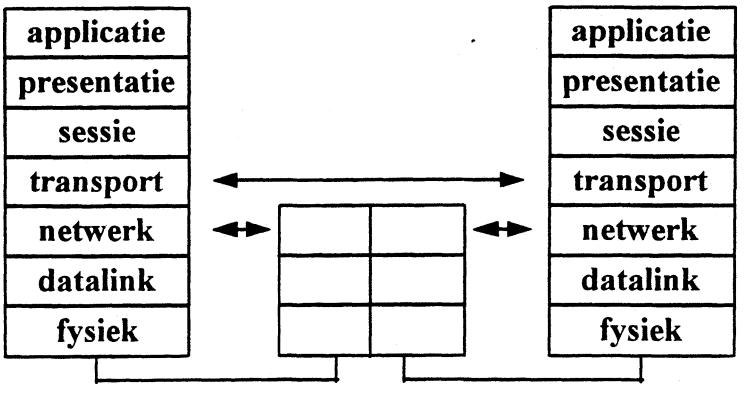
In de Token-Ring omgeving wordt gebruik gemaakt van Source Routing bridges. Hier wordt gebruik gemaakt van een ander principe dan bij Ethernet. Bij Source Routing wordt er vanuit gegaan dat eind stations zèlf moeten bepalen of een frame verstuurd moet worden met gebruikmaking van de eventueel aanwezige bridges.

Er is maar één produkt beschikbaar om Ethernet en Token-Ring op basis van bridging aan elkaar te koppelen, de IBM 8209. Merk op dat een dergelijk device nogal wat kunstgrepen zal moeten uithalen, aangezien de framestructuur en werking van het Ethernet protocol en Token-Ring protocol totaal verschillend is.

IIR TECHNOLOGY

5.1 Algemeen - routers

- **Router nodig *per* protocol (bijna altijd: MPR)**
- **Moeilijk te configureren**



Routers zitten nóg een stapje hoger in het OSI model, namelijk op de netwerklaag. Dit impliceert dat routers verstand hebben van netwerklaag protocollen. Immers, netwerklagen op hosts kijken tegen routers aan!

Wat betekent routering eigenlijk? Routering betekent dat het netwerklaag protocol een hiërarchische adresseringsstructuur gebruikt. Met andere woorden, adressen in de vorm van netwerknummer, hostnummer. In hoofdstuk 2 is al besproken dat dergelijke functionaliteit in het IP protocol aanwezig is. Voorbeelden van andere protocollen die hieraan voldoen zijn DECnet (waar overigens gesproken wordt over areanummer, nodenummer, IPX/SPX en Banyan VINES. Een voorbeeld van een protocol dat géén gebruik maakt van een hiërachische namenstructuur is NetBIOS. NetBIOS zèlf is dus ook *niet* routeerbaar.

Routers, dat wil zeggen routing functionaliteit, kan in verschillende hoedanigheden gebruikt worden. Zo kunnen Unix hosts perfect als router fungeren, aangezien de *routed* deamon van de Unix host een volwaardige router maakt. Veel vaker wordt er echter gebruik gemaakt van dedicated routers van leveranciers als Cisco, Wellfleet, Crosscom en 3Com. Het configureren van dergelijke routers is vaak geen eenvoudig karwei en vereist nogal wat kennis van zaken.

IIR TECHNOLOGY

5.1 Algemeen

- **Het IP protocol heeft routing 'in zich' (hiërachische adres structuur)**
- **Routers betrokken bij het routing proces**
- **Onderscheid tussen IGP's en EGP's**
- **IGP's binnen '*autonomous systems*', EGP's tussen '*autonomous systems*'**
- **Rest van dit hoofdstuk: IGP's**

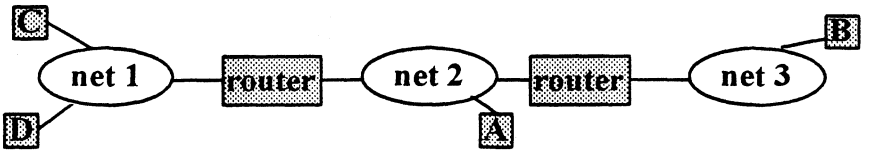
Een voorwaarde om aan routering te kunnen doen is een hiërarchische naamstructuur. Zoals besproken in hoofdstuk 3, voorziet IP hier in: IP kent een 'netwerkgedeelte' van het adres en een 'host gedeelte' van het adres. Een ander protocol dat dit ondersteunt is bijvoorbeeld het DECnet protocol. Bij DECnet wordt er onderscheid gemaakt tussen een 'area gedeelte' van het adres en een 'node gedeelte' van het adres.

Binnen de Internet wereld wordt er onderscheid gemaakt tussen IGP's en EGP's. IGP is de verzamelnaam voor de zogenaamde Interior Gateway Protocols. Dit zijn de protocollen die binnen één zogenaamd 'autonomous system' gebruikt worden. Een autonomous system kan beschouwd worden als een zelfstandige eenheid die aan het Internet gekoppeld is. Dus het bedrijfsnetwerk van bedrijf X, verspreid over verschillende lokaties die onderling met routers gekoppeld zijn, is te beschouwen als een autonomous system. De koppeling met het Internet gebeurt wellicht met één specifieke router. Deze router moet meedoen in het 'routing proces op het Internet', en gebruikt hiervoor een EGP: een Exterior Gateway Protocol.

IIR TECHNOLOGY

5.2 Direct routing

- **Routing beslissing op basis van IP adres**
- **Splitsen in (Sub)netwerk part en Host part**
- **Als destination host op zelfde (sub)netwerk: rechtstreeks afleveren (*direct routing*)**
- **Als destination host op ander (sub)netwerk: afleveren bij GW (*indirect routing*)**

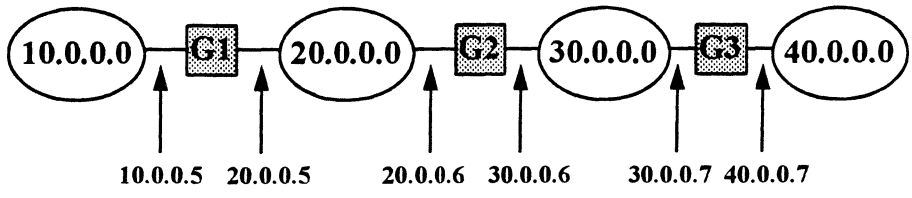


Hosts die iets te verzenden hebben, kunnen zelf op basis van het destination IP address bepalen of er al dan niet gerouteerd moet gaan worden. Daartoe zullen ze het destination IP address opsplitsen in een (sub)netwerk part en een host part. Als de destination zich op hetzelfde (sub)netwerk bevindt als de host zelf, dan kan de host het IP datagram zelf afleveren (middels ARP en dergelijke: zie hoofdstuk 3). Als het destination network afwijkt van het eigen network, dan zal de host het pakketje moeten afleveren bij 'een' router. Dit wordt indirect routing genoemd.

IIR TECHNOLOGY

5.3 Indirect routing

- **Indirect routing: host niet op zelfde netwerk**



To reach:	20.0.0.0	Route to:	deliver direct
	30.0.0.0		deliver direct
	10.0.0.0		20.0.0.5
	40.0.0.0		30.0.0.7

Indirect routing door hosts gebeurt op basis van tabellen, vandaar ook de benaming table-driven routing. Wellicht ten overvloede: ook een router is een host, weliswaar een host met een speciale 'missie', maar nog steeds een host! Op basis van het destination network address en de tabel die de host aan boord heeft, kan worden besloten waar het betreffende IP datagram naar toe gestuurd moet worden.

Als de host 'toevallig' niet een router is, maar bijvoorbeeld een Unix host of een MS DOS PC, dan wordt er vrijwel altijd gebruik gemaakt van wat genoemd wordt de '*default gateway*'. De betreffende host hoeft dan maar één regeltje te onthouden:

'als het niet direct afgeleverd kan worden (direct routing), dan moet het afgeleverd worden bij router X'.

In feite is dit ook een vorm van table-driven routing, maar dan met een table met maar één entry.

IIR TECHNOLOGY

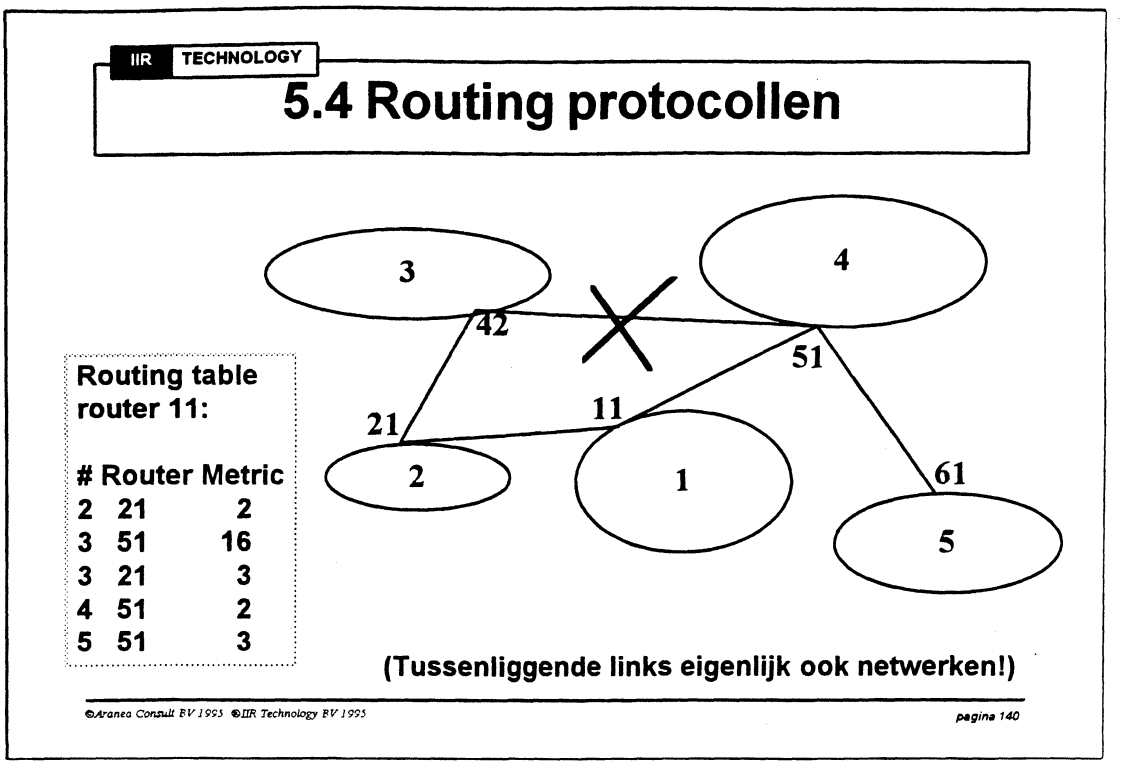
5.4 Routing protocollen

- **RIP: Routing Information Protocol**
- **Beschikbaar in twee versies, RIP en RIP-2**
- **Niet *alleen* voor TCP/IP, ook voor bijvoorbeeld IPX/SPX**
- **RIP is een '*distance vector*' protocol**
- **Standaard in Unix beschikbaar als *routed***
- **Beschreven in RFC1058: *jaren* nadat het protocol in gebruik was!**
- **RIP draait op UDP, well known port 520**

In het voorgaande is besproken dat de hosts de IP datagrammen óf zelf afleveren bij de destination (direct delivery), of naar een router sturen (default gateway). En routers waren in feite ook IP hosts die de IP datagrammen weer forwarden op basis van tabellen (table-driven routing). De tabellen die de routers gebruiken zijn uiteraard geen statische tabellen. Het mag duidelijk zijn dat het wellicht nog mogelijk is om binnen één autonomous system de tabellen van de routers te vullen, over bedrijfsgrenzen heen is dit niet meer te doen. Dit probleem wordt opgelost door het gebruik van routing protocollen, onder te verdelen in IGP's en EGP's. De rest van dit hoofdstuk concentreert zich op de bekendste IGP's.

Hèt bekendste IGP is het Routing Information Protocol. Het protocol was al wijd verbreid in gebruik alvorens het gestandaardiseerd werd in een RFC. RIP is een applicatieprotocol dat boven op UDP draait, maar is ook beschikbaar boven op diverse andere transportprotocollen (onder andere IPX/SPX).

Het RIP protocol wordt gebruikt door de routers om informatie over de bereikbaarheid van netwerken uit te wisselen. Deze bereikbaarheid wordt uitgedrukt in het aantal hops dat een netwerk van de betreffende router verwijderd is. Vandaar dat RIP een zogenaamd 'distance vector' protocol is. Afstanden tot destination hosts (lees: netwerken) worden enkel uitgedrukt in aantallen te passeren routers, en houden geen rekening met andere kwalificaties van de tussenliggende links (zoals MTU's en speed).

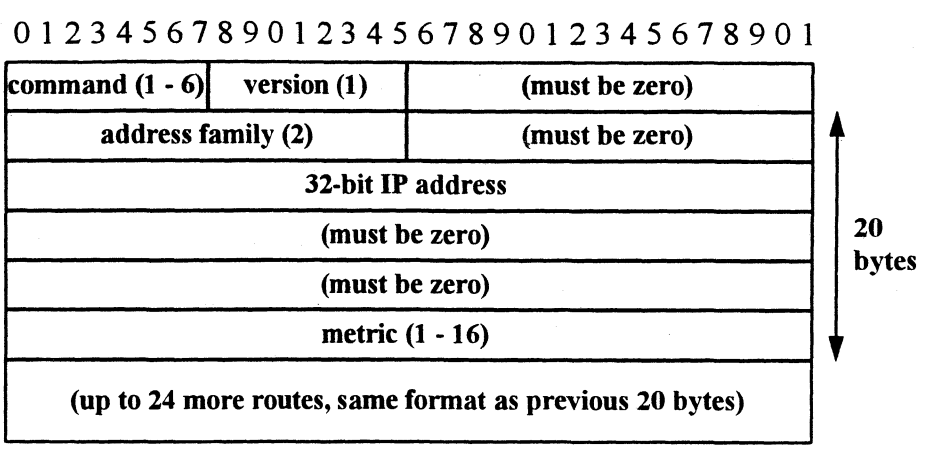


Bovenstaand een topologie met de routing tabel van router 11. De routing tabel van de router bevat informatie over de destination networks, het aantal hops dat het betreffende netwerk verwijderd is, en de router waar het IP datagram moet worden afgegeven.

De 'metrics' geven dus het aantal hops aan dat het netwerk verwijderd is. In de topologie is de link tussen router 51 en router 42 'down'. Dat betekent dat netwerk 3 niet meer via router 51 te bereiken is. Binnen het RIP protocol wordt dit aangegeven met behulp van de waarde 'infinite': 16. Dit betekent dat een netwerk voor het RIP protocol slechts 16 netwerken (0-15) 'verderop' kan liggen!

IIR TECHNOLOGY

5.4 Routing protocollen



Bovenstaand het formaat van een RIP bericht. In dit voorbeeld een RIP bericht dat gebruik wordt voor *IP adressen*. Zoals vermeld, RIP wordt ook door andere protocollen gebruikt, zoals IPX/SPX.

command - 1 is een *request*, 2 is een *reply*.

version - '1' als het versie 1 betreft, '2' voor RIP-2.

address family - voor IP adressen is dit '2'.

32-bit IP address - een adres dat door de betreffende router bereikt kan worden. Dit kan een netwerkadres zijn (netwerk 145.46.0.0) of een host nummer (145.46.203.254).

metric - Dit geeft het aantal hops aan dat de betreffende router verwijderd is van het netwerk/de host. Maximum waarde is 16!

Het RIP protocol kan als volgt beschreven worden.

Initialization. Als *routed* start, zal over ieder interface een request worden uitgestuurd. Aan iedere router wordt gevraagd de volledige routing tabel te sturen. Op point-to-point links wordt dit naar 'het andere eind' gestuurd, op een LAN wordt dit middels een broadcast verstuurd.

Request received. Als het request het 'speciale' request is uit de initialize state (commando code 1, address family 0, metric is 16), dan zal de gehele tabel teruggestuurd worden. Op een request voor een netwerk dat niet bekend is bij de betreffende router wordt metric '16' ('infinite') teruggegeven.

Response received. Responses worden gevalideerd en kunnen updates (wijzigen of verwijderen) van de eigen routing table tot gevolg hebben.

Regular routing updates. Routers zullen *iedere* 30 seconden (een gedeelte van) routing tabellen naar iedere buur-router versturen.

Triggered updates. Als de metric van een route wijzigt, moet de router dit aan de andere routers kenbaar maken. Hiervoor hoeft niet de gehele routing table verstuurd te worden.

Routes hebben time-outs van 3 minuten: als binnen 3 minuten geen update komt, zal die route gemarkeerd worden met metric '16' (infinite).

IIR TECHNOLOGY

5.4 Routing protocollen

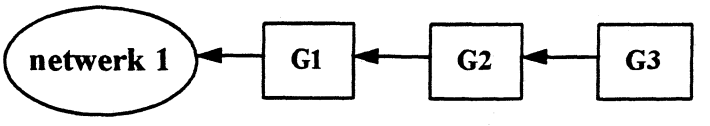
- **Nadelen van RIP:**
  - **RIP maakt geen onderscheid tussen IP adressen van (sub)netwerken en/of hosts**
  - **RIP heeft tijd nodig om te 'stabiliseren' nadat een link is weggevallen (minuten)**
  - **Tijdens dit stabiliseren kunnen loops ontstaan ('slow convergence')**
  - **Het werken met hop counts negeert andere eigenschappen die aan een link gekoppeld kunnen worden (MTU, speed)**
  - **Beperkte netwerkomvang**

RIP heeft duidelijk een aantal nadelen. Zo kan RIP aan een destination address nooit zien of het een netwerkadres, subnetwerkadres of hostadres betreft. RIP heeft immers geen kennis over subnetwerk maskers en kan dus nooit aan een adres als 145.46.0.0 zien of dit een klasse B netwerk adres is. Verder heeft RIP tijd nodig om te stabiliseren. Als diverse timers enigzins 'ongunstig' staan, dan is de kans groot dat er routing loops in het netwerk ontstaan (zie volgende slide). Een algemeen nadeel van distance vector protocollen is het feit dat er enkel wordt gemeten in afstanden. Daarbij worden andere eigenschappen van een link, zoals MTU en speed, genegeerd. Het mag duidelijk zijn dat een netwerk dat volgens router A 3 netwerken verderop ligt, en volgens router B 1 netwerk verderop ligt, via A tóch sneller te bereiken is als het 3 Ethernetten betreft ten opzichte van 1 2400 Baud dial-up link! En verder heeft RIP natuurlijk het nadeel van de bijzonder beperkte netwerkomvang.

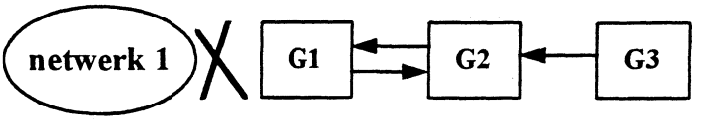
IIR TECHNOLOGY

5.4 Routing protocollen

- **Alle routers hebben een route naar netwerk 1**



- **G1 verliest zijn link naar netwerk 1, G2 vertelt dat er nog een alternatieve route is...**



Bovenstaand het probleem van 'slow convergence' in een RIP netwerk. Alle routers weten hoe ze network 1 kunnen bereiken. Router 3 (in de figuur G3 genoemd) kan netwerk 1 bereiken via G2, met een metric van 3. Router 2 (in de figuur G2 genoemd) kan netwerk 1 bereiken via router G1 met een metric van 2. Router G1 kan netwerk 1 zelf bereiken met een metric van 1.

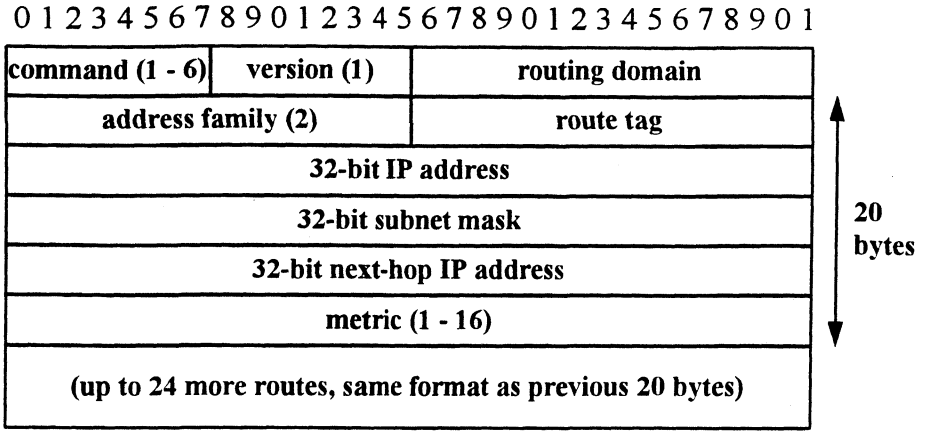
Op het moment dat router G1 de link naar netwerk 1 verliest, heeft hij een probleem. Maar gelukkig krijgt hij een update van router G2, die aangeeft dat hij wel weet hoe hij netwerk 1 kan bereiken, via een metric van 2. Router G1 zal zijn eigen routing table hierop aanpassen: netwerk 1 is niet meer met een metric van 1 te bereiken (rechtstreeks gekoppeld), maar G2 heeft nog een route met een metric van 2. Een metric van 2, daarbij opgeteld zijn eigen metric van 1, dus via router G2 en een metric van 3. Router G1 zal dit doorgeven, en zo krijgt router G2 te horen dat de route naar netwerk 1, via G1, nu ineens niet meer een metric van 1 heeft, maar een metric van 3. Dus zal de router zijn eigen routing table aanpassen, en een metric van 4 opnemen voor netwerk 1 (metric 3 van router G1 en één hop van hemzelf). Dit gaat natuurlijk weer via een broadcast naar router G1 toe, enzovoorts...

Er is een aantal voorzorgsmaatregelen getroffen om dit soort calamiteiten in de kiem te smoren. In Comer (zie literatuurlijst) worden de volgende 4 mogelijke algorithmen genoemd:

- 'split horizon update': geen informatie verzenden over een netwerk over het interface waar je zelf de informatie over binnen hebt gekregen
- 'good news travels quickly, bad news travels slowly'
- 'hold down': informatie over het niet bereikbaar zijn van een netwerk moet enige tijd worden vastgehouden zodat alle routers het slechte nieuws kunnen vernemen
- 'Poisson reverse': zend een aantal malen een metric 'infinite'

IIR TECHNOLOGY

5.4 Routing protocollen



Bovenstaand het RIP-2 formaat. RIP-2 past het eigenlijke routing protocol niet aan, maar gebruikt de door RIP *niet-gebruikte* velden ('must be zero') om additionele informatie mee te sturen:

version - RIP-2 heeft version 2.

routing domain - een identifier van de routing deamon van wie het pakket afkomstig is. Dit wordt gebruikt om onderscheid te kunnen maken tussen verschillende routing deamons op één en dezelfde router.

route tag - hierin zit het autonomous system nummer voor EGP en BGP.

subnet mask - subnet masker van (ieder) 32-bit IP address.

next-hop IP address - het IP address van de volgende hop waar het betreffende pakket naartoe gestuurd moet worden. Als de volgende hop de router betreft waar dit pakket vandaan komt, staat in dit veld 0x00 ingevuld.

Er is een eenvoudig authentificatie schema opgenomen in RIP-2. Als de eerste 20 byte entry in een RIP pakket als address family de waarde 0xffff heeft en als route tage de waarde '2' is opgenomen, dan zijn de overige 16 bytes van het pakket een ASCII password.

Als laatste toevoeging heeft RIP-2 nog de mogelijkheid om multicasting te ondersteunen.

IIR TECHNOLOGY

5.4 Routing protocollen

- **OSPF: Open Shortest Path First**
- **Geen vector distance, maar *link state* protocol**
- **Per definitie convergeren link state protocollen *sneller* dan vector distance protocollen**
- **Draait *rechtstreeks* op IP (eigen protocol nummer)**
- **Gestandaardiseerd in RFC1583**
- **Verschillende routes voor elke 'TOS'**
- **Authentificeerde uitwisseling van route informatie**

OSPF biedt een aantal voordelen, als link state protocol, ten opzichte van RIP:

- OSPF kan verschillende routes bepalen naar een eindbestemming afhankelijk van het IP veld TOS (Type-of-service).
- Aan een interface wordt een dimensie-loze kostenfactor gekoppeld. Deze kostenfactor wordt bepaald aan de hand van RTT, reliability, throughput, of enige andere parameter.
- Als er verschillende, gelijk kostende routes, naar een eindbestemming zijn, zal OSPF aan *load balancing* doen.
- OSPF ondersteunt subnet masks (evenals RIP-2, overigens). Routes naar een expliciete host worden aangegeven met subnet masker 0xffffffff.
- Point-to-point links hebben géén aparte IP nummers nodig. Dit wordt *unnumbered networks* genoemd.
- Er is een eenvoudig authentificatie schema aanwezig (vgl. RIP-2).
- OSPF kan multi-casting in plaats van broadcasting gebruiken (vgl. RIP-2).

IIR TECHNOLOGY

5.4 OSPF - networks

- **Stub network**
  - **Geen transit network, slechts verbonden met een router**
  - **Een host route is ook een stub network**
- **Multi-access network**
  - **Transit network, tenminste met twee routers verbonden**
  - **Twee typen:**
    - **Broadcast networks, bv. Ethernet of TRN**
    - **Non-broadcast networks, bv. X.25 en FR**
- **Routers**
  - **Transit node met minimaal twee interfaces**

OSPF maakt onderscheid in verschillende soorten routers, afhankelijk van de functie die ze hebben. Een router die een enkel netwerk verbindt met de rest van de wereld, hoeft bijvoorbeeld slechts als een bridge te fungeren: is het bestemmingadres niet het eigen netwerk, dan moet het datagram overgezet worden. Zo'n netwerk wordt een stub-network genoemd, de router een stub-router.

Een netwerk kan ook door twee of meer routers verbonden zijn met de rest van de wereld. In dat geval is het mogelijk, dat verkeer verzonden wordt over het netwerk dat als bestemming niet het netwerk heeft. In dat geval is er sprake van een transit functie, en wordt zo'n netwerk een transit netwerk genoemd. De erop aangesloten routers zijn dan eveneens transit routers.

Er zijn twee soorten transit netwerken: broadcast en non-broadcast. In de eerste categorie vallen de LAN's zoals Ethernet en Token-Ring, in de tweede categorie de meeste WAN's zoals X.25 en Frame-Relay.

IIR TECHNOLOGY

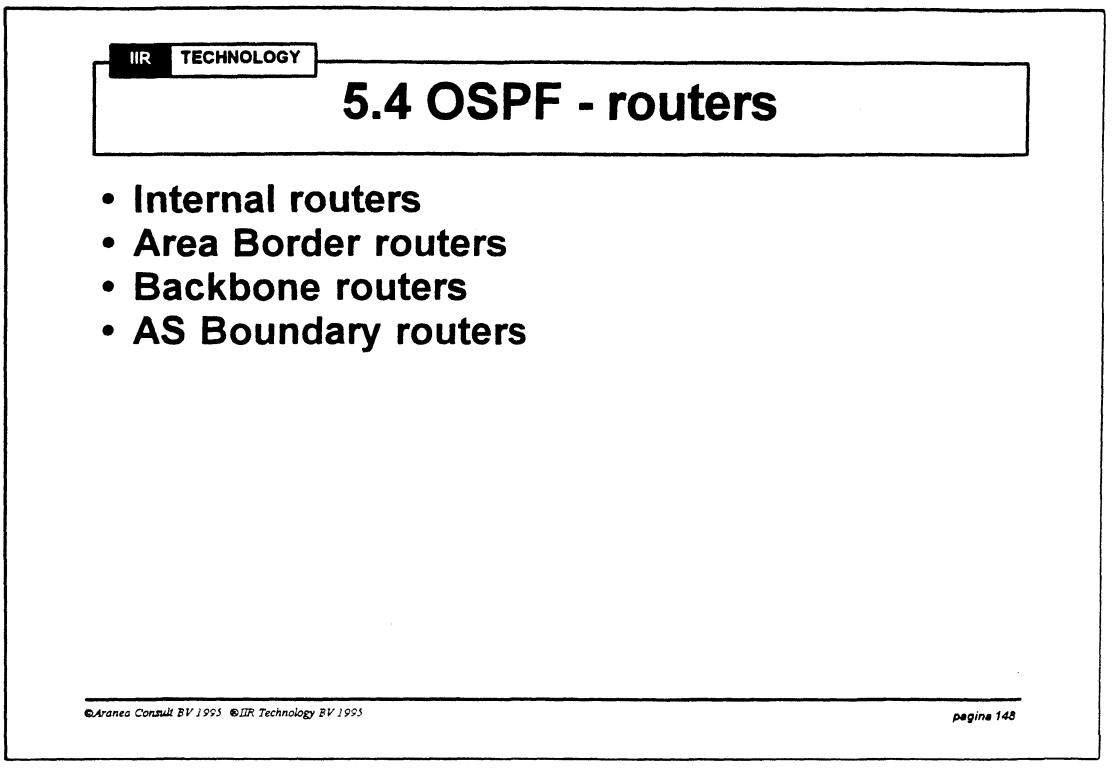
5.4 OSPF - networks

- **Backbone network**
  Bestaat uit netwerken die niet tot een area behoren, de erop aangesloten routers, en routers die bij meerdere area's horen
- **Area**
  Groep van aangesloten subnetted netwerken of hosts
- **Boundary network**
  Netwerk dat niet tot het autonomous system behoort

Meerdere netwerken die logisch tot hetzelfde domein behoren, worden bij OSPF gegroepeerd tot areas. Routers binnen een area hebben alleen informatie over de netwerken binnen de area en het adres van de router die de area met de rest van de wereld verbindt.

Tussen de areas moet een backbone zijn die de areas onderling verbindt. De routers in dat netwerk worden backbone routers genoemd en hebben informatie over hoe de verschillende areas te bereiken. Ze hebben geen informatie over de netwerken binnen de areas. Een backbone moet een aaneengesloten geheel zijn.

Het laatste soort netwerken zijn de boundary networks. Dit zijn de netwerken die de autonome systemen verbinden, eigenlijk een backbone voor het koppelen van de (autonome) backbone netwerken. In deze netwerken worden EGP's gebruikt, die buiten deze cursus vallen.



Internal Routers

- Routers in een area die route gegevens bewaren naar netwerken en hosts binnen de area , plus een pad naar een area border router.

Area Border routers

- Routers die areas verbinden met het backbone network. Ze bewaren route gegevens naar verbindingen binnen de area, naar andere area border routers en naar AS boundary routers in de backbone.

Backbone routers

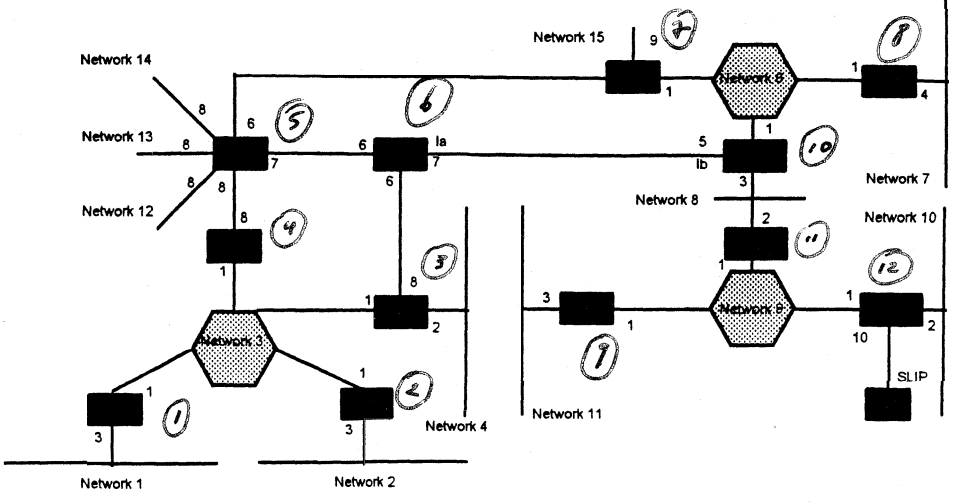
- Routers die niet tot een area behoren. Ze bewaren route gegevens naar andere backbone routers en naar area border routers.

AS Boundary routers

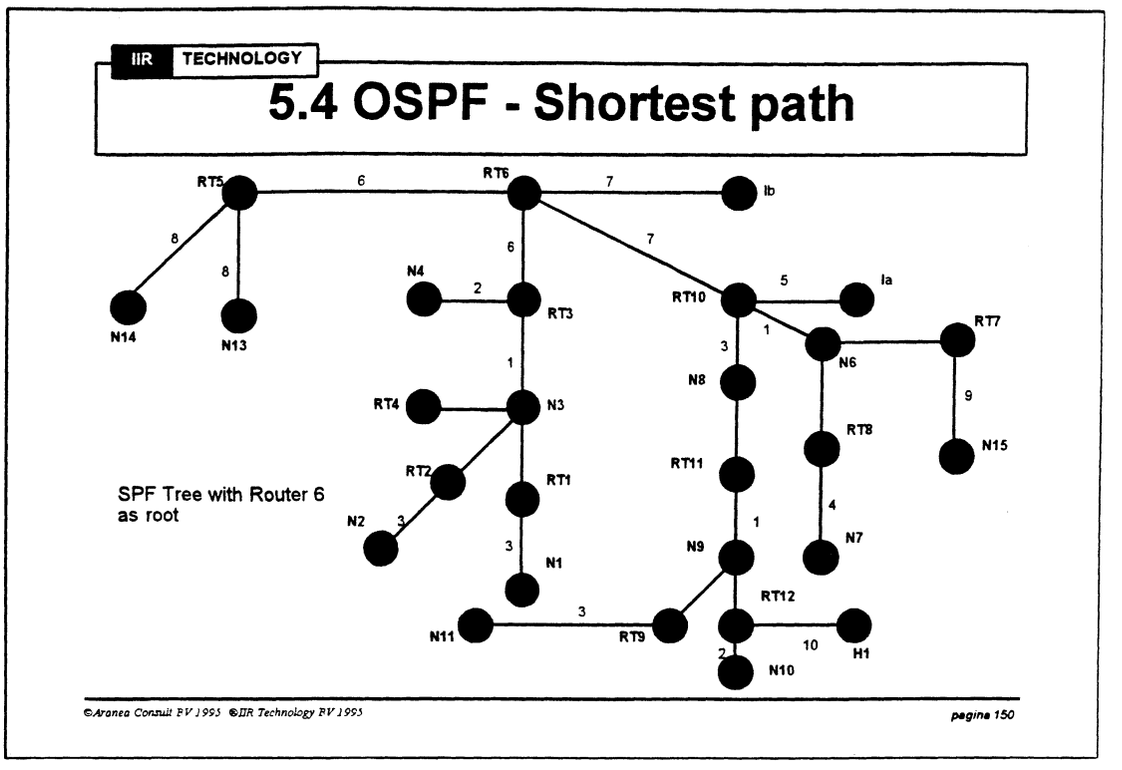
- Routers die gegevens uitwisselen met routers van andere autonomous systems.

IIR TECHNOLOGY

5.4 OSPF - Een voorbeeld AS



Dit is een voorbeeld van een autonoom systeem. In dit voorbeeld zijn nog geen area's onderscheiden. Zoals duidelijk wordt, zijn alle netwerken genummerd. Iedere router weet aan welke netwerken hij gekoppeld is, en wat de afstand is tot de volgende hop (de path-cost). Transiet netwerken worden door de routers als hop gezien, waarbij de afstand tot de hop altijd 1 is (de zeshoeken in het voorbeeld)



Op basis van de path-cost die in elke router aanwezig is, kunnen de routers ieder een kortste route berekenen naar ieder gegeven bestemmingsnetwerk. In het voorbeeld is router 6 als uitgangspunt genomen. Omdat iedere router zijn path-costs doorgeeft, kan iedere router een volledige afbeelding van de topologie van het netwerk maken.

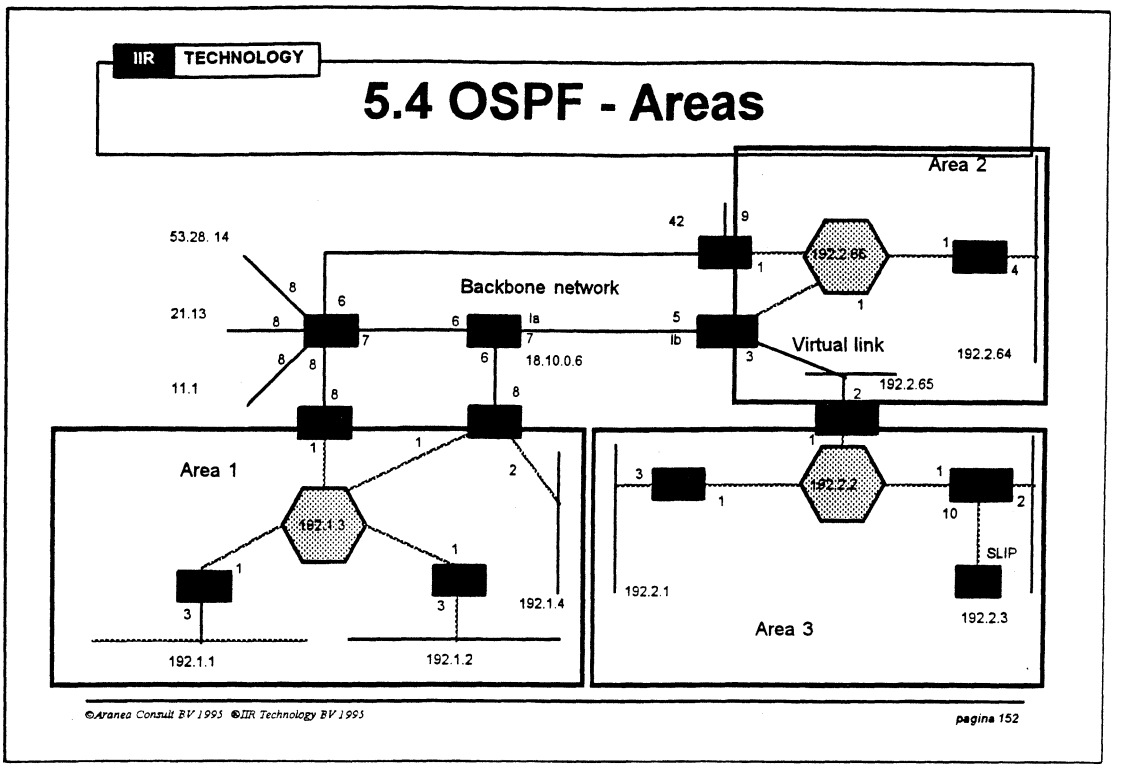
IIR TECHNOLOGY

5.4 OSPF - Routing tables (R6)

Type	Dest	Area	Path type	Cost	Next hop	Adv. router
N	N1	0	intra-area	10	RT3	
N	N2	0	intra-area	10	RT3	
N	N3	0	intra-area	7	RT3	
N	N4	0	intra-area	8	RT3	
N	Ib	0	intra-area	7	*	
N	Ia	0	intra-area	12	RT10	
N	N6	0	intra-area	8	RT10	
N	N7	0	intra-area	12	RT10	
N	N8	0	intra-area	10	RT10	
N	N9	0	intra-area	11	RT10	
N	N10	0	intra-area	13	RT10	
N	N11	0	intra-area	14	RT10	
N	H1	0	intra-area	21	RT10	
ASBR	RT5	0	intra-area	6	RT5	
ASBR	RT7	0	intra-area	8	RT10	
N	N13	*	type 1 ext	14	RT5	
N	N14	*	type 1 ext	14	RT5	
N	N15	*	type 1 ext	17	RT10	

Dit is een voorbeeld van een routing-tabel zoals die in het voorbeeld er uit zal zien. In de type kolom wordt aangegeven of de bestemming een netwerk of een router is (N=netwerk, ASBR=Autonomous System Boundary Router).

Ook wordt het type routering aangegeven: intra-area voor routering binnen een area, inter-area voor routering tussen areas en external voor routering via een niet autonoom systeem.

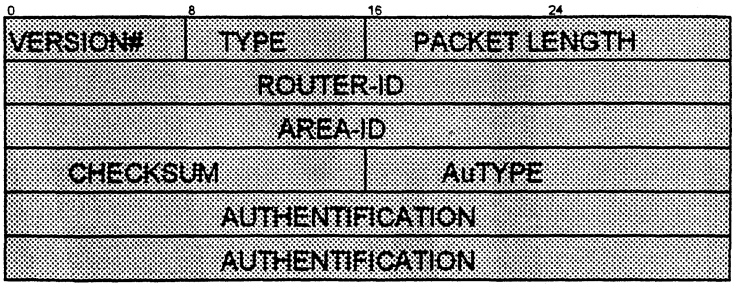


In dit voorbeeld wordt gebruik gemaakt van areas. De topologie is nu opgedeeld in drie verschillende areas. Om een backbone tot stand te kunnen brengen die de areas met elkaar verbindt, is het in dit geval nodig om een virtuele backbone verbinding te definiëren door een area. Was deze link niet gedefinieerd, dan was area 3 onbereikbaar geworden voor hosts uit area 1 en vanuit de backbone.

Het doel van areas is om routingtabellen zo klein mogelijk te maken. Daarom moeten binnen een area de gebruikte netwerk nummers aaneengesloten zijn, zodat met een enkele verwijzing in een routingtabel alle netwerken binnen een area aangeduid kunnen worden.

IIR TECHNOLOGY

5.4 OSPF Packets: Header formaat



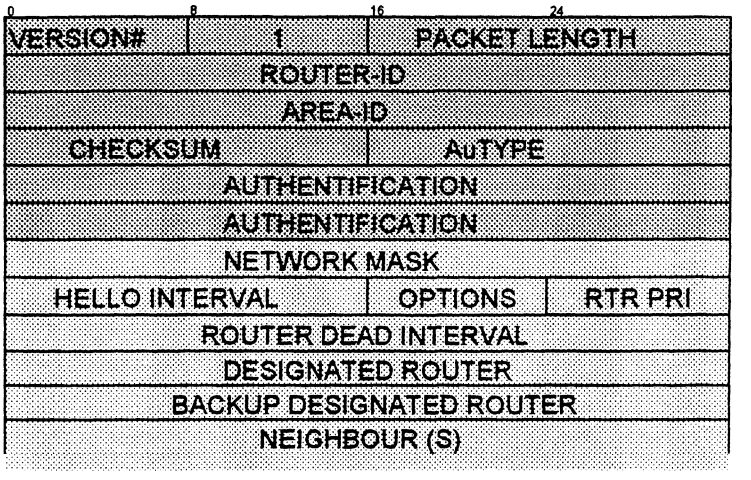
Current version: 2
Protocol type:
1 Hello
2 Database description
3 Link State request
4 Link State update
5 Link State acknowledgement

Ieder OSPF packet heeft dezelfde 24 byte grote header. Deze header bevat alle nodige gegevens om te bepalen of het pakket voor verdere verwerking gebruikt moet worden.

Version #	Het OSPF versie nummer. Dit is nu versie 2.
Type	De volgende OSPF typen zijn mogelijk: • 1 Hello • 2 Database Description • 3 Link State Request • 4 Link State Update • 5 Link State Acknowledgment
Packet length	De lengte van het packet in bytes, inclusief de standaard OSPF header.
Router ID	De Router ID van de bron. In OSPF zijn de bron en bestemming van een routing pakket de uiteinden van een (potentiële) aangrenzing.
Area ID	Een 32 bit nummer dat het area identificeert waar het pakket toe behoort. Alle OSPF pakketten worden geassocieerd met een enkele area. Pakketten die over een virtuele link gaan, worden gelabeled met backbone Area ID 0.0.0.0.
Checksum	De standaard IP checksum van de volledige inhoud van het pakket, behalve het 64-bit authenticatie veld.
AuType	Geeft het authenticatie schema dat gebruikt wordt voor dit pakket.
Authentication	Een 64-bit veld, gebruikt voor het authenticatie schema.

IIR TECHNOLOGY

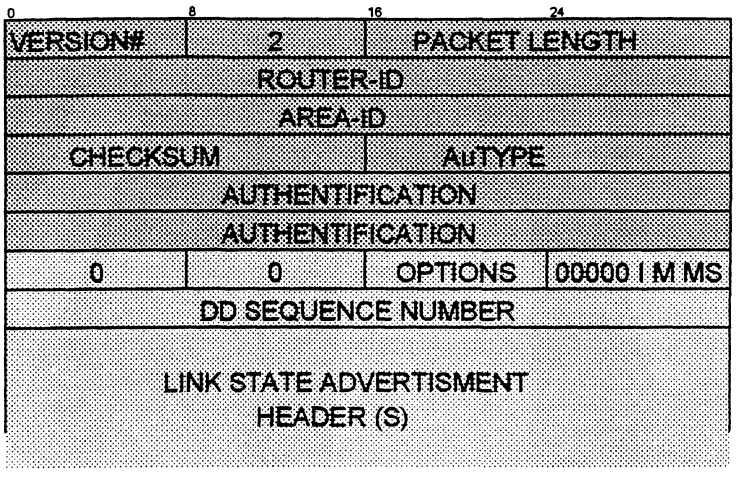
5.4 OSPF - Hello packet



Network mask	Het network mask dat bij deze interface hoort. Bv., als de interface een klasse B netwerk is, met het derde byte als subnet, dan is het masker 0xffffff00.
Options	De optionele mogelijkheden van deze router.
HelloInterval	Aantal seconden tussen de Hello packets van deze router.
Rtr Pri	De prioriteit van deze router. Alleen gebruikt voor het kiezen van de (Backup) Designated Router.
RouterDeadInterval	Aantal seconden voordat een stille router dood wordt verklaard.
Designated Router	De Designated Router voor dit netwerk. De Designated Router wordt aangegeven door zijn IP interface adres in het netwerk. Is 0.0.0.0 als er geen Designated Router is.
Backup Designated Router	De Backup Designated Router voor dit netwerk.
Neighbour	De Router IDs van iedere router waarvan geldige Hello packets recentelijk gezien zijn in het netwerk. Recentelijk betekent: gedurende de laatste RouterDeadInterval seconden.

IIR TECHNOLOGY

5.4 OSPF DB Description pakket



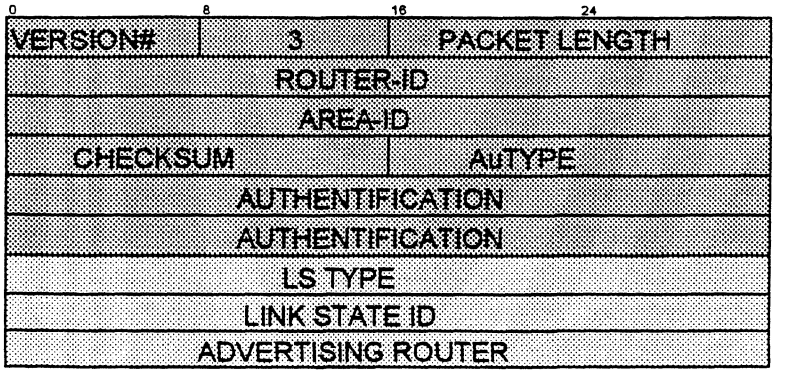
Het fromaat van het Database Description pakket is overeenkomsting de Link State Request en Link State Acknowledgment pakketten.
Het belangrijkste van alle drie is een lijst met items, die ieder een stuk van de topologische database beschrijven.

0	Gereserveerd.
Options	Optionele mogelijkheden van de router.
I-bit	Het Init bit. Indien 1 is dit pakket het eerste in de opeenvolging van Database Description Packets.
M-bit	Het More bit. Indien 1 geeft het aan dat er meer Database Description Pakketten komen.
MS-bit	Het Master/Slave bit. Indien 1 geeft het aan dat de router is de master gedurende het Database Exchange proces. Anders is de router slave.
DD sequence number	Wordt gebruikt voor de volgordelijkheid van de Database Description Pakketten. De initiële waarde (aangeven doordat het Init bit gezet is) moet uniek zijn.

De rest van het pakket bevat een gehele of gedeeltelijke lijst van de onderdelen uit de topologische database. Iedere link state advertisement in the database wordt beschreven door zijn link state advertisement header. De link state advertisement header bevat alle gegevens die nodig zijn om zowel de advertisement als de status van de huidige advertisement uniek te identificeren.

IIR TECHNOLOGY

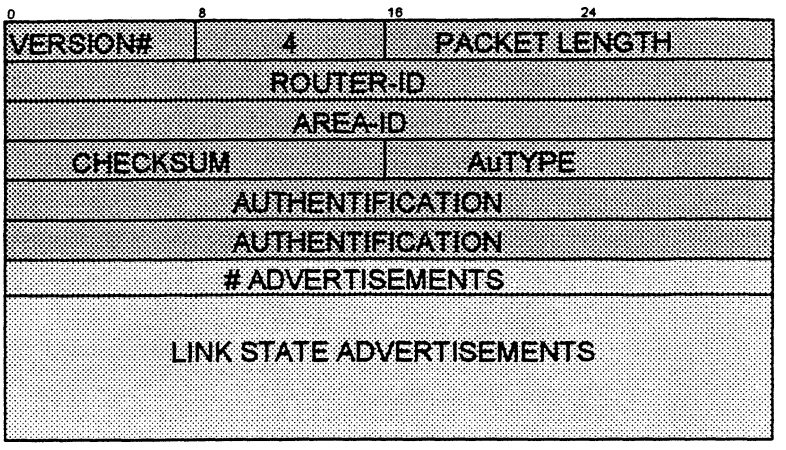
5.4 OSPF - Link State Request



Elke advertisement waarom verzocht wordt, wordt aangeduid met zijn LS type, Link State ID en Advertising Router. Dit duidt het advertisement uniek aan, maar niet de versie. Er wordt vanuitgegaan dat Link State Request pakketjes verzoeken zijn voor de meest recente versie (wat dat ook zijn mag).

IIR TECHNOLOGY

5.4 OSPF Link State Update packets



Link State Update pakketjes zijn OSPF packet type 4. Met deze pakketjes wordt het verspreiden van de link state advertisements geïmplementeerd. Ieder Link State Update pakket bevat een collectie van link state advertisements; een hop verder van zijn oorsprong. Meerdere link state advertisements kunnen in een enkel pakket opgenomen worden.

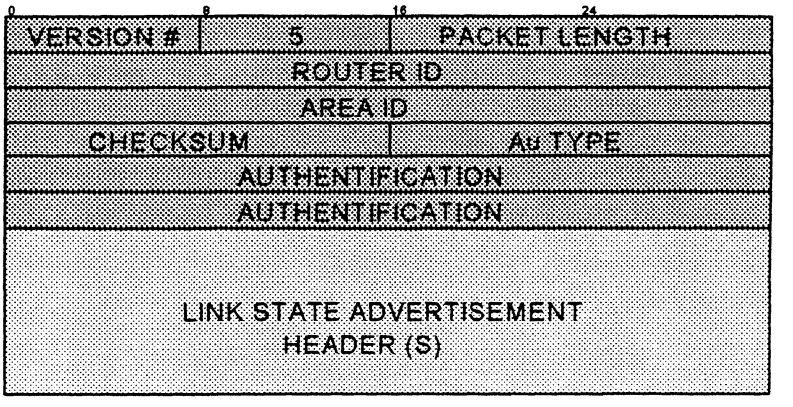
Link State Update pakketjes zijn multicast op de fysieke netwerken die dat ondersteunen. Om de flooding procedure betrouwbaar te maken, worden flooded advertisements bevestigd met Link State Acknowledgment pakketjes. Als hertransmissie van een bepaalde advertisements nodig is, worden de opnieuw verzonden advertisements altijd vervoerd in unicast Link State Update pakketjes.

advertisements Aantal link state advertisements in deze update.

De body van een Link State Update pakket bestaat uit een lijst van link state advertisements. Iedere advertisement begint met een gemeenschappelijke 20 byte header; de link state advertisement header. Verder is het formaat van ieder van de vijf mogelijke link state advertisements types verschillend.

IIR TECHNOLOGY

5.4 OSPF Link State Acknowledgement



Link State Acknowledgment Packets zijn OSPF packet type 5. Om het verspreiden van link state advertisements betrouwbaar te maken, worden verspreide advertisements expliciet bevestigd. Deze bevestiging wordt bewerkstelligd door het zenden en ontvangen van Link State Acknowledgment packets. Meerdere link state advertisements kunnen bevestigd worden in een enkel Link State Acknowledgment pakket.

Afhankelijk van de status van de zendende interface en de bron van de advertisements die bevestigd worden, wordt een Link State Acknowledgment of naar het multicast adres AllSPFRouters, naar het multicast adres AllDRouters, of als unicast verzonden.

Het formaat van dit pakket is gelijk aan dat van het Data Description packet. De body van beide pakketjes is een lijst van link state advertisement headers.

Iedere bevestigde link state advertisement wordt beschreven door zijn link state advertisement header. De link state advertisement header bevat alle gegevens die nodig zijn om zowel de advertisement, als de advertisement's huidige versie uniek te identificeren.

IIR TECHNOLOGY

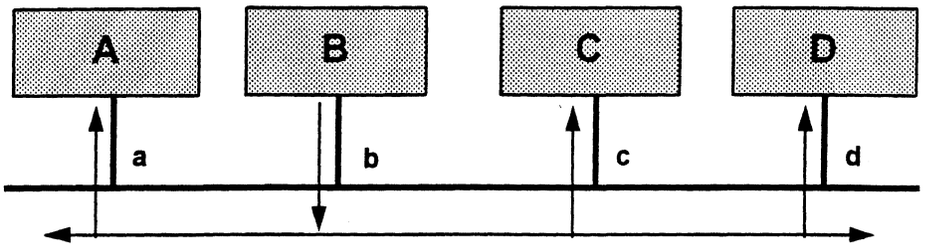
6. Adressenbeheer

6.1 RARP
6.2 BootP
6.3 DHCP
6.4 DNS

IIR TECHNOLOGY

6.1 RARP

- **Reverse Address Resolution Protocol (RARP): de vraag**



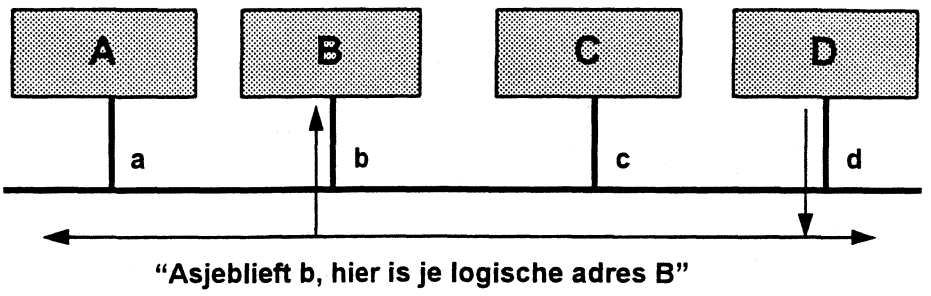
"Ik ben fysieke b en ik zoek mijn logische adres, kent iemand dat?"

In bepaalde gevallen kan het voorkomen dat hosts bij het opstarten niet hun eigen IP adres in een configuratie tabel hebben staan. Er bestaat een mogelijkheid om dit logische IP adres 'op te halen' bij een daartoe bestemde server in het netwerk. Deze functionaliteit wordt geboden door zogenaamde RARP servers, waarbij RARP staat voor 'Reverse Address Resolution Protocol'. Op basis van het eigen Ethernetadres, vraagt de host naar zijn eigen IP adres. RARP pakketten zijn ARP pakketten met operation code 3 (voor de request) of 4 (voor de reply). Ook nu weer wordt er van Ethernet broadcast adres ff ff ff ff ff ff gebruik gemaakt.

IIR TECHNOLOGY

6.1 RARP

- **Reverse Address Resolution Protocol (RARP): het antwoord**



"Asjeblieft b, hier is je logische adres B"

De Ethernet broadcast komt aan bij de RARP server, in dit geval server D. Deze zal in zijn tabel bij het binnengekomen Ethernetadres het bijbehorende IP adres opzoeken en dit terugsturen naar b/B. Het mag duidelijk zijn, dat er goed beheer nodig is voor wat betreft de RARP tabel. Immers, vervangen van Ethernetkaarten betekent muteren in de tabel.

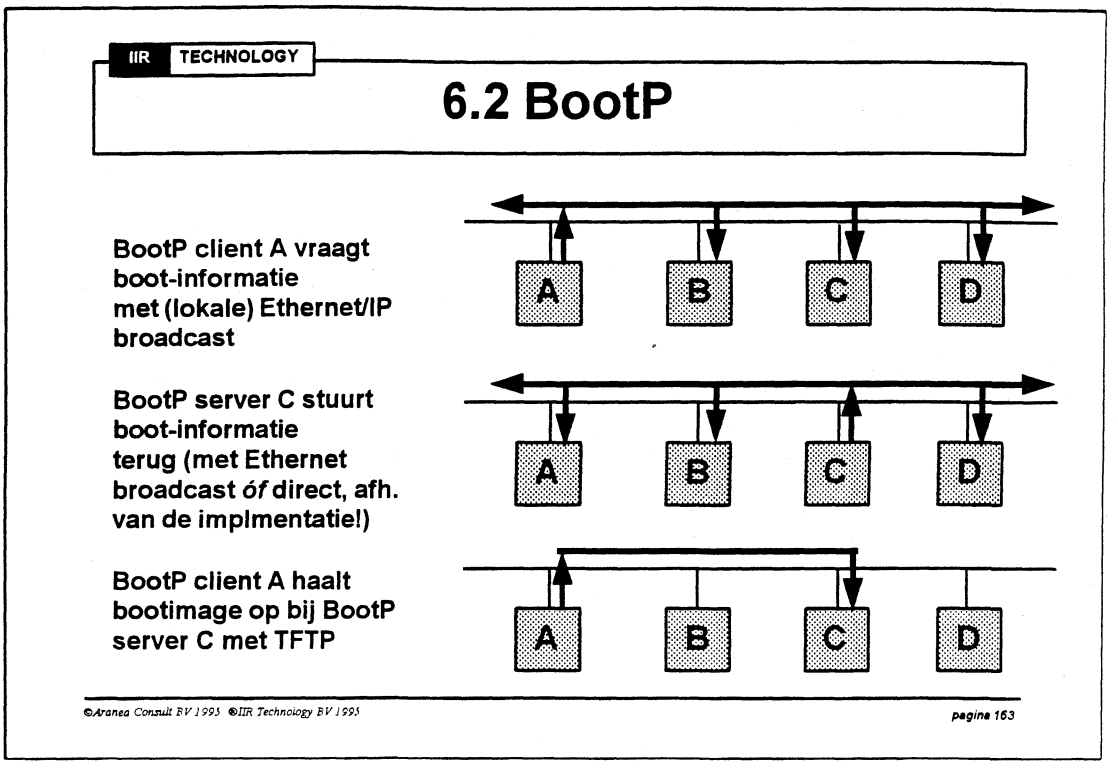
IIR TECHNOLOGY

6.2 BootP

- **BootP: Bootstrap Protocol**
- **Gebruikt, in eerste instantie, door *diskless* workstations om bootinformatie te krijgen**
- **Client/server proces, gebruikmakend van well known UDP ports 68 (client) en 67 (server)**
- **Door gebruik van '*BootP relay agents*' niet noodzakelijk om ieder segement met een BootP server uit te rusten**
- **Wordt vaak gebruikt in combinatie met TFTP om het OS te downloaden**

BootP is een applicatie protocol dat bovenop UDP draait. Het is bedoeld als protocol om diskless workstations te voorzien van hun boot-image. Dit betekent dat de code van het BootP protocol vaak in ROM chips op de Ethernet adapter aanwezig is, tezamen met het (kleine, snelle) UDP protocol. Daarnaast zal vaak het TFTP protocol aanwezig zijn om het daadwerkelijke operating system te downloaden. De reden voor het gebruik van UDP en TFTP mag duidelijk zijn. Kleine, snelle protocollen die optimaal geschikt zijn voor het downloaden van de operating system informatie in de diskless werkstations. Als dit op basis van TCP en FTP zou moeten gebeuren, zou er veel meer coding in de ROM's op de Ethernet adapters gestopt moeten worden.

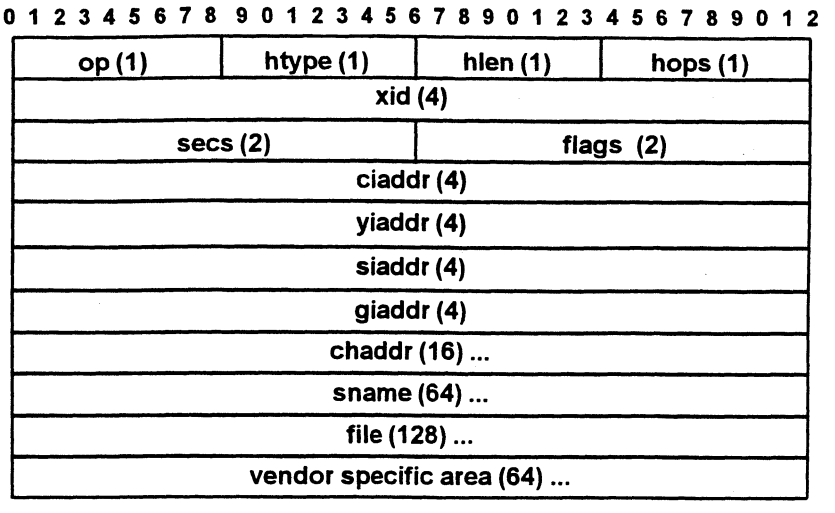
De reden waarom bij het protocol BootP ervoor gekozen is om zowel de server als de client een ***vast*** port number te geven, komt voort uit het feit dat servers de mogelijkheid hebben om hun informatie terug te sturen met behulp van een *broadcast* (zowel op Ethernet als IP niveau, zie volgende slides). Als de server inderdaad zijn antwoord zou terugsturen met behulp van een broadcast, en er op dat moment machines in het netwerk zijn die toevallig een client port number hebben gedefinieerd dat overeenkomt met het client BootP port number, dan zullen deze machines verkeerde informatie op die ports binnenkrijgen. Immers, ze verwachten op die port bijvoorbeeld Telnet informatie en krijgen plotsklaps, middels de broadcast, BootP informatie. Om dit te voorkomen, is afgesproken dat BootP clients zich altijd achter well known port number 68 bevinden.



De client in bovenstaand voorbeeld gaat tijdens het booten een BootP request uitsturen. Dit is een broadcast (op zowel Ethernet als IP niveau) en wordt dus door alle machines ontvangen. Die machines die als BootP server zijn geïnitialiseerd zullen reageren op deze aanvraag middels een BootP reply. Dit kan zijn met behulp van een broadcast (immers, de server kan niet met ARP het Ethernet adres achterhalen) of rechtstreeks (als de server de ARP cache mag 'patchen'). De BootP reply bevat de benodigde informatie voor de client, waaronder de *server* waar de client zijn boot informatie kan gaan ophalen. Dit hoeft overigens absoluut ***niet*** dezelfde server te zijn als degenen die het reply geeft! BootP servers worden vaak ook gebruikt om alleen maar IP adressen uit te delen. Op die manier kan het beheer van de IP adressen centraal gebeuren. De BOOTPTAB.TXT file bevat dan (minstens) een overzicht van de hardware adressen en de bijbehorende IP adressen. Ook andere informatie kan in een dergelijke file worden opgenomen. Er is nog een ander protocol dat gebruikt kan worden om op basis van het Ethernet adres het bijbehorende IP adres te vinden. Dit is het RARP protocol. Het nadeel van dit protocol is echter dat het een datalinklaag protocol is: RARP pakketten worden rechtstreeks ingepakt in Ethernet frames en zijn derhalve ***niet*** routeerbaar. BootP daarentegen, is een applicatielaag protocol dat bovenop IP draait, en dus routeerbaar. Dat betekent dat niet ieder segment (subnet) over een BootP server hoeft te beschikken, maar er gebruik kan worden gemaakt van zogenaamde **BootP relay agents**. Dit zijn niks anders dan routers die bepaalde functionaliteit ondersteunen, namelijk die van 'doorgeef-BootP-server'. De router gaat namelijk 'luisteren' op well-known port 67 naar BootP requests. Als een dergelijk BootP request binnenkomt, dan zal de router het **giaddr** veld (zie PDU beschrijving) invullen en doorsturen naar de èchte BootP server (die zich best drie netwerken verderop mag bevinden!). Deze ziet het **giadrr** ingevuld staan en weet dat een router bij het BootP proces betrokken is, en zal derhalve de reply naar de router terugsturen, die het vervolgens doorstuurt naar de client.

IIR TECHNOLOGY

6.2 BootP

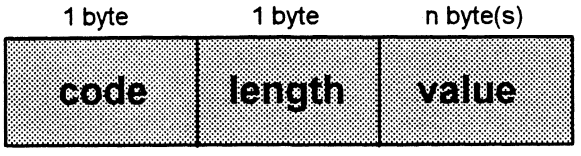


op	**Message operation code**. Code 1 wordt gebruikt voor de BOOTREQUEST, code 2 wordt gebruikt voor de BOOTREPLY.
htype	**Hardware address type**. Hiervoor worden dezelfde codes gebruikt als bij het ARP protocol. '1' is 10 Mb Ethernet.
hlen	**Hardware address length**. In het geval van 10Mb Ethernet: 6.
hops	**Hops.** Door een client op '0' gezet. Wordt gebruikt door relay-agents om op te hogen.
xid	**Transaction ID**. Random nummer dat door de client gezet wordt en door client en server tijdens communicatie gebruikt wordt.
secs	**Seconds.** Geeft aan hoelang een client al aan het booten is. Dit kan voor een secundairy BootP server reden zijn om te gaan reageren, aangezien het booten al een bepaalde tijd duurt en dus de primairy BootP server wel eens down zou kunnen zijn.
flags	**Flags**. Eerste bit is BROADCAST bit, de rest is reserved (unused). Sommige clients kunnen geen unicast terugontvangen van de BootP server, omdat ze immers nog niet (volledig) geïnitialiseerd zijn. Door het broadcast bit aan te zetten, geven ze de server te kennen deze dient te antwoorden middels een broadcast.
ciaddr	**Client IP address**. Door client ingevuld indien dit al bekend is.
yiaddr	**Your IP address**. Door server ingevuld om de client te vertellen wat diens IP adres is.
siaddr	**Server IP address**. Het IP adres van de (volgende te gebruiken) server.
giaddr	**Relay IP address**. Adres van de relay agent die betrokken is bij de boot.
chadrr	**Client hardware address**.
sname	**Server host name**.
file	**Boot file name**. Indien nodig, voorafgegaan door een volledig pad.
vendor	**Vendor specific items**. Dit zijn optionele parameters die meegegeven kunnen worden, waarvan een aantal vast ligt en een aantal vrij te kiezen is. Ze beginnen altijd met een 4 octetten lang 'magic cookie', decimaal gezien 99, 130, 83, 99.

IIR TECHNOLOGY

6.2 BootP

- **In de 'vendor specific area' kunnen zogenaamde 'vendor extensions' worden opgenomen**
- **Vendor extensions zijn (deels) gedefinieerd in RFC 1534 (opvolger van 1048, 1497), en hebben de vorm:**

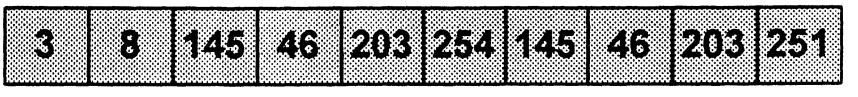


BootP pakketten bevatten een vendor specific information veld. Dit veld kan gebruikt worden om allerlei 'extra' informatie te verschaffen, waaronder de mogelijk van belang zijnde default gateway en subnet mask. De vendor specific information is 64 bytes lang en kan ook informatie bevatten die proprietary is voor een bepaalde leverancier of een bepaalde site.

IIR TECHNOLOGY

6.2 BootP

- **Vendor extensions van vaste lengte, bijvoorbeeld *Subnet Mask* (code 1)**
- **Vendor extensions van variabele lengte, bijvoorbeeld *Gateways* (code 3)**
- **Vendor extensions 128-256 zijn gereserveerd voor 'site specific use' (behalve 255!)**
- **speciale codes '0' en '255'**



Een aantal vendor specific items is gedefinieerd in RFC's en maakt gebruik van vaste codes. Daarnaast is er een range gedefinieerd die gebruikers en leveranciers naar eigen goeddunken kunnen gebruiken. Ervaringen uit de praktijk hebben geleerd dat het belangrijk is om clients en servers bij eenzelfde leverancier te betrekken. Vanwege een aantal vrijheden in de protocol definitie in de RFC's, zijn niet alle clients en servers even compatible met elkaar.

Op de slide een voorbeeld van een gedeelte van de vendor specific area. Code 3 geeft IP adressen van routers aan. De code wordt gevolgd door het length veld; dit staat op 8, met andere woorden er zijn twee IP adressen, en dus 2 routers, gedefinieerd.

De code 255 is een speciale code en geeft het einde van de vendor specific area aan. De code 0 is een speciale code en wordt gebruikt als 'padding' character, met name bij DHCP (zie volgende paragraaf) om velden van vaste lengte op te vullen.

IIR TECHNOLOGY

6.3 DHCP

- DHCP: Dynamic Host Configuration Protocol
- DHCP is een uitbreiding op BootP
- Ook DHCP maakt gebruik van UDP well known ports 67 (server) en 68 (client)
- Beschreven in RFC's 1534 (options) en 1541 (protocollen)
- DHCP servers ondersteunen BootP clients
- DHCP servers ondersteunen BootP relay agents

DHCP is *géén* nieuw protocol, maar een uitbreiding op het bestaande BootP protocol. Het 'enige' dat er veranderd is, is dat er een aantal extra codes zijn gedefinieerd die nu gebruikt kunnen worden in het vendor specific information veld. Het vendor specific information veld heet nu ook anders (zie volgende slide) en is groter geworden.

IIR TECHNOLOGY

6.3 DHCP

- **Wat biedt DHCP meer ten opzichte van BootP:**
  1. ***Reusable IP adresses***
     Met behulp van DHCP is het mogelijk om tijdelijk IP adressen uit te delen, die later weer 'hergebruikt' kunnen worden
  2. ***Meer configurerende parameters***
     DHCP kan *alle* parameters leveren die nodig zijn voor een computer om als IP host te fungeren
  3. ***Andersoortige client-identifier mogelijk***
     BootP clients kunnen alleen maar geïdentificeerd worden met behulp van een HW-adres; DHCP ondersteunt ook andere identifiers

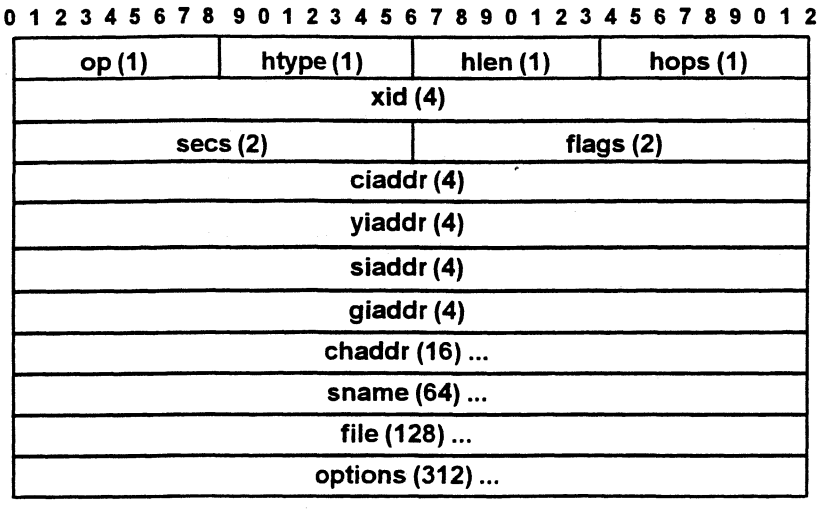
Eén van de voordelen van DHCP ten opzichte van BootP, is dat er *dynamisch* IP adressen kunnen worden uitgedeeld. De DHCP server onderhoudt een pool van adressen en kan uit deze pool adressen verstrekken op het moment dat dat nodig is. Dat betekent dat als een site 10 machines heeft, die nooit alle 10 tegelijk aan zullen staan, die site kan volstaan met 7 IP adressen, die uitgedeeld (en beheerd!) kunnen worden door de DHCP server.

Tevens kan DHCP meer configurerende parameters uitdelen dan BootP, deels omdat er meer codes gedefinieerd zijn voor DHCP, en deels omdat een DHCP pakket groter kan zijn dan een BootP pakket (zie PDU beschrijvingen).

Overigens kan een DHCP server nog steeds als een BootP server geconfigureerd worden en op basis van Ethernet adressen, IP adressen uitdelen. Voordeel van DHCP boven BootP is echter, dat DHCP dit niet alleen kan op basis van Ethernet (meer algemeen: datalinklaag) adressen, maar ook op 'andersoortige' identifiers (bijvoorbeeld 'computernaam', een zelf te definiëren string op een computer in één of andere .INI file).

IIR TECHNOLOGY

6.3 DHCP



op	**Message operation code**. code 1 wordt gebruikt voor de BOOTREQUEST, code 2 wordt gebruikt voor de BOOTREPLY
htype	**Hardware address type**. Hiervoor worden dezelfde codes gebruikt als bij het ARP protocol. '1' is 10 Mb Ethernet
hlen	**Hardware address length**. In het geval van 10Mb Ethernet: 6
hops	**Hops.** Door een client op '0' gezet. Wordt gebruikt door relay-agents om op te hogen.
xid	**Transaction ID**. Random nummer dat door de client gezet wordt en door client en server tijdens communicatie gebruikt wordt.
secs	**Seconds.** Geeft aan hoelang een client al aan het booten is. Dit kan voor een secundairy BootP server reden zijn om te gaan reageren, aangezien het booten al een bepaalde tijd duurt ende primairy BootP server dus wel eens down zou kunnen zijn.
flags	**Flags**. Eerste bit is BROADCAST bit, de rest is reserved (unused). Sommige clients kunnen geen unicast terugontvangen van de BootP server omdat ze immers nog niet (volledig) geïnitialiseerd zijn. Door het broadcast bit aan te zetten, geven ze de server te kennen dat deze dient de antwoorden middels een broadcast.
ciaddr	**Client IP address**. Door client ingevuld indien dit al bekend is.
yiaddr	**Your IP address**. Door server ingevuld client IP adres.
siaddr	**Server IP address**. Het IP adres van de (volgende te gebruiken) server.
giaddr	**Relay IP address**. Adres van de relay agent die gebruikt wordt bij booten over subnetten heen.
chadrr	**Client hardware address**.
sname	**Server host name**.
file	**Boot file name**. Indien nodig, voorafgegaan door een volledig pad.
options	**Options**. Dit zijn optionele parameters die meegegeven kunnen worden, waarvan een aantal vast ligt (DHCP message types bijvoorbeeld), en een aantal vrij te kiezen is. De options beginnen met het 4 octetten 'magic cookie' (decimaal 99, 130, 83, 99), evenals de vendor extensions bij BootP.

IIR TECHNOLOGY

6.3 DHCP

- **DHCP servers ondersteunen 3 mechanismen voor het alloceren van IP adressen**
  1. ***Automatic Allocation***
     Toewijzen van een permanent IP adres
  2. ***Dynamic Allocation***
     Toewijzen van een IP adres voor een bepaalde tijd ('*lease*') of totdat de client het adres teruggeeft
  3. ***Manual Allocation***
     Het IP adres wordt door de beheerder uitgedeeld, en door DHCP 'medegedeeld'

DHCP kent een belangrijke toevoeging op het oorspronkelijke BootP protocol als het gaat om het uitdelen van IP adressen. Daar waar BootP enkel vaste IP adressen kan uitdelen aan vaste Ethernet adressen, kan DHCP méér. DHCP kan ook uit een pool van beschikbare adressen tijdelijk IP adressen uitdelen aan clients, hetzij op basis van het Ethernet adres, hetzij op basis van een andere unieke client identifier.

IIR TECHNOLOGY

6.3 DHCP

- **DHCP messages verzonden middels speciale 'Message type' options (option code 53)**
- **Er zijn 7 DHCP message types gedefinieerd:**

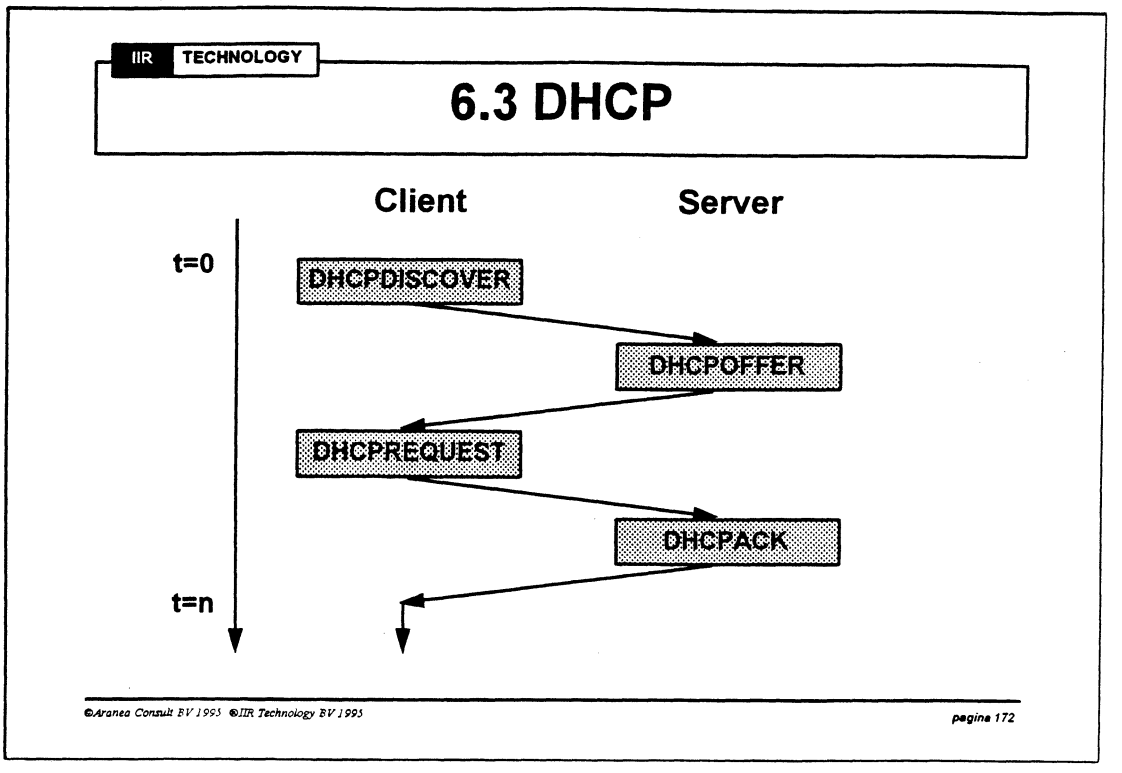
DHCPDISCOVER	***C -> S***
DHCPOFFER	***S -> C***
DHCPREQUEST	***C -> S***
DHCPDECLINE	***C -> S***
DHCPACK	***S -> C***
DHCPNAK	***S -> C***
DHCPRELEASE	***C -> S***

- ***Ieder* DHCP bericht bevat de 'Message type' option (is onderscheid met BootP pakket!!!)**

DHCPDISCOVER	Dit is een request van een client (broadcast) om te achterhalen of er een DHCP server op het net aanwezig is.
DHCPOFFER	Dit is het antwoord van een server op een DISCOVER, met daarin vermeld het aanbod van parameters.
DHCPREQUEST	Het antwoord van een client waarmee wordt aangegeven dat de parameters die worden aangeboden, zullen worden geaccepteerd.
DHCPACK	Antwoord van de server: de parameters.
DHCPNAK	Server geeft als antwoord aan client dat deze weigert de parameters te verstrekken.
DHCPDECLINE	Client geeft aan server door dat bepaalde configuratie parameters invalid zijn.
DHCPRELEASE	Client geeft parameters (in het bijzonder het IP adres) terug aan de server).

Andere DHCP option codes:

code	***omschrijving***
50	Requested IP address
51	IP address lease time
52	Option overload (ook *sname* en *file* veld in DHCP PDU kunnen worden gebruikt om options in op te nemen!)
54	Server identifier
55	Parameter request list
56	Message
57	Maximum DHCP message size
58	Renewal (T1) time value (binnen het aflopen van de lease tijd)
59	Rebinding (T2) time value (na het aflopen van de lease tijd)
60	Class identifier (om verschillende machines in één klasse onder te brengen en eenzelfde soort informatie naar toe te downloaden)
61	Client-identifier (anders dan het Ethernet address)



Bovenstaande een ***mogelijke*** conversatie tussen een DHCP client en een DHCP server. De client begint met een DHCPDISCOVER (code 53) broadcast over het netwerk, met mogelijk daarin opgenomen een 'Parameter Request List' (code 57) en de 'DHCP Message Size' (code 55). Ook mag een client hierin al een voorstel doen voor zijn eigen IP adres en de bijbehorende lease time. Eén of meerdere DHCP servers pakken dit op en zullen een aanbod gaan doen (DHCPOFFER). Dit aanbod bevat o.a. het 'toekomstige' IP adres, opgenomen als het 'yiaddr' veld. Bepaling van uit te delen IP adressen:

- als eerste komt het IP adres dat de client een vorige keer ook al gekregen heeft in aanmerking (database met 'previously bindings'), er vanuit gaande dat dit adres nog voorkomt in de pool van beschikbare adressen, anders
- het adres in option 'Requested IP Address' (code 50), als het een valide adres bevat dat nog niet gealloceerd is, anders
- een nieuw adres uit de pool van beschikbare adressen.

De DHCPOFFER (code 53) zal tevens informatie bevatten over de 'lease' die de client in acht moet nemen (minimum leasetijd 1 uur, maximum leasetijd 'infinite', weergegeven door 0xffffffff) (code 51). Als er geen adressen beschikbaar zijn, kan de server een DHCPNAK (code 53) terug sturen, met daarin in de 'Message' option (code 56) een foutmelding. Een client kan wachten tot een aantal offers van een aantal servers binnen is alvorens hij reageert middels een request. Middels een DHCPREQUEST (code 53) vraagt de client vervolgens aan een uitgeselecteerde server om inderdaad deze parameters te mogen gebruiken. De server waar dit aan gevraagd wordt, staat benoemd in de 'Server Identifier' option (code 54). De server bevestigt met een DHCPACK (code 53) dat de parameters, in het bijzonder het IP adres, geregistreerd staan in de 'previously bindings' database. Mocht de DHCPACK (code 53) niet helemaal in orde zijn, dan zal de client een DHCPDECLINE (code 53) sturen en opnieuw beginnen.

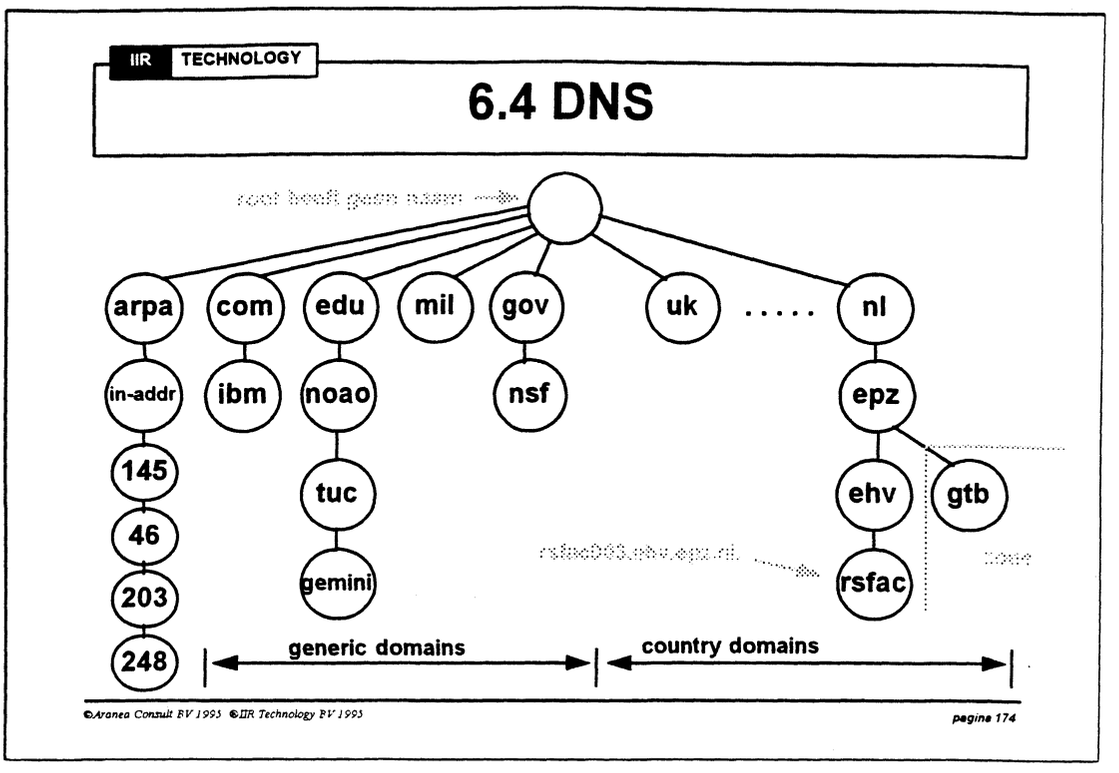
IIR TECHNOLOGY

6.4 DNS

- **DNS: Domain Naming System**
- **IP adressen zijn moeilijk te onthouden**
- **DNS server vertaalt *naam* naar IP-adres**
- **Een domain-name is samengesteld**
  voorbeeld: rsfac003.ehv.epz.nl.
- **Het naming system is gedecentraliseerd opgezet**
- **Boomstructuur met DNS servers zorgt ervoor dat één server niet alle namen hoeft te kennen**

Het is vrij lastig voor gebruikers om IP adressen te onthouden. Een telnet sessie naar 145.46.203.248 zegt allicht minder dan een telnet sessie naar PROLIN-SERVER. Met andere woorden, als er 'ergens' een vertaalslag gemaakt zou kunnen worden van PROLIN-SERVER naar 145.46.203.248, dan zou de gebruiker hier mee geholpen zijn. In eerste instantie is dit probleem opgelost door gebruik te maken van een HOSTS.TXT file. Deze ASCII file bevat een tabel met daarin begrijpelijke namen gekoppeld aan bijbehorende IP adressen. In den beginne werd deze file *centraal* bijgehouden door het NIC. Middels FTP werd deze file vervolgens gedistribueerd naar de diverse machines in het Internet. Het mag duidelijk zijn dat dit op een gegeven moment niet meer werkbaar was. Dus moesten de verschillende sites zèlf dit soort files gaan bijhouden. Ook daarvoor gold dat het ondoenlijk was om deze files up-to-date te houden, zeker toen ook PC's in TCP/IP netwerken werden opgenomen. Op dat moment zijn de ontwikkelingen begonnen omtrent het Domain Naming System. Dit is een gedistribueerde database, waarvan verschillende servers kennis hebben van verschillende delen van de complete name space, en in staat zijn om clients elkaar te verwijzen naar de juiste naming servers.

DNS is een client/server applicatie die op UDP óf TCP draait en achter well known port 53 zit. Vaak zal de resolver het proberen via UDP. Als vervolgens antwoord wordt gekregen waarin het TC (truncated, zie verderop) bit is gezet, dan betekent dit dat slechts de eerste 512 bytes van het UDP datagram zijn verstuurd en de rest is afgekapt. De resolver zal dan via TCP dezelfde vraag nog een keer stellen (aangezien TCP segmenten geen beperkingen kennen).



De boom van DNS namen begint bij de 'root'. Deze heeft geen naam. Onder deze root hangt een aantal bladeren die tezamen de verschillende top level domains vormen. Deze zijn onder te verdelen in 'generic domains' en 'country domains' (de country domain namen zijn de two-character namen zoals gedefinieerd in ISO 3166). De 'bladeren' van de boom mogen labels van maximaal 63 characters hebben. De labels moeten voldoen aan IP hostname requirements (beginnen met een letter, eindigen met een letter of cijfer, en daartussen een willekeurige combinatie van letters, cijfers en het '-' teken). De domain naam van een willekeurig blad in de boom wordt gevormd door de naam van het blad, gevolgd door alle hogere bladen tussen het betreffende blad en de root, gescheiden door een '.' character. De root zelf heeft geen naam, een valide naam zou zijn rsfac.epz.ehv.nl. (let op de afsluitende punt!). Een naam die eindigt op de '.', noemen we een FQDN (Fully Qualified Domain Name).

Geen enkele 'autoriteit' heeft kennis van het gehele netwerk, maar slechts de verantwoordelijkheid over een deel van de boom, '***zone***' genoemd. De verantwoordelijke voor een zone moet er voor zorgen dat zijn database up to date blijft.

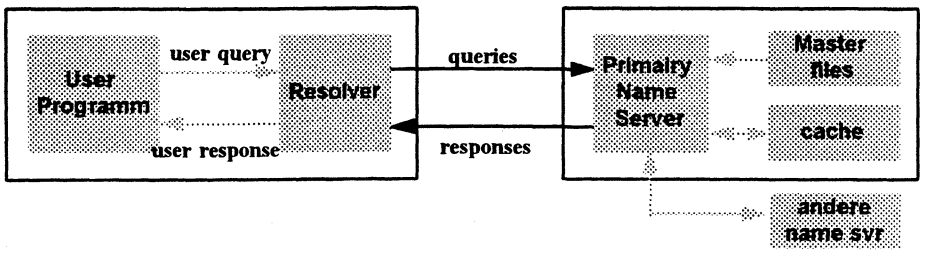
Een geval apart is de tak in-adrr.arpa. Deze tak wordt gebruikt bij zogenaamde ***pointer queries*** om op basis van IP adressen, bijbehorende host namen te vinden (ook onder authority van de betreffende IP netwerk beheerder!).

Generic domain	***Description***
com	commercial organisation
edu	education institution
gov	other U.S. governmental organisations
int	international organisations
mil	U.S. militairy
net	networks
org	other organisations

IIR TECHNOLOGY

6.4 DNS

- **DNS werkt met *Name Servers* en *resolvers***
- ***Primairy* DNS halen informatie van *diskfile*, andere *name servers* of *cache*,**
- ***Secundairy* DNS halen iedere 3 uur hun informatie van de primairy DNS: *zone transfer***



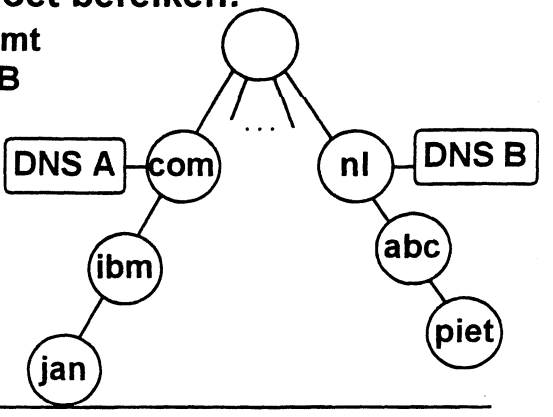
Vanuit de client bekeken, worden queries doorgestuurd naar een bekende Name Server met behulp van een resolver. Deze resolver kent tenminste één name server (achter well known port 53). Vragen van de gebruiker(s-programma's) worden door de resolver aan de name server geteld. Deze zal óf antwoord geven op de gestelde vraag, of doorverwijzen naar een andere name server, of zelf aan een andere server gaan vragen en vervolgens het antwoord geven.

Domain name servers hoeven niet alle andere name servers te kennen die in het netwerk zijn opgenomen. Ze moeten wèl het IP adres kennen van de root servers in het Internet (+/- 10 stuks: een exacte lijst is met anonymous FTP op te halen op server *nic.ddn.mil*). Deze root-servers kennen de IP adressen van alle geautoriseerde second-level domains. De root-server geeft dan aan de requesting server aan dat deze naar de second-level domain server moet gaan, etcetera. Dit is dus een iteratief proces. Root servers mogen ***nooit*** recursief benaderd worden: anders zou dit betekenen dat 'de hele wereld' de root servers zou gaan laten uitzoeken welk IP adres bij welke host hoort.

IIR TECHNOLOGY

6.4 DNS

- **Jan in domain ibm.com wil communiceren met Piet in domain acb.nl. Hiertoe vraagt hij aan DNS A hoe hij hem moet bereiken:**
  - ***Recursief*: DNS A neemt kontakt op met DNS B en geeft aan Jan het IP adres af**
  - ***Iteratief*: DNS A vertelt Jan dat hij kontakt op moet nemen met DNS B**



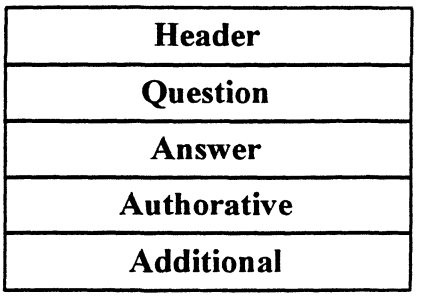
Resolvers hebben (soms) een configuratie file, genaamd resolve.conf, nodig. Deze file bevat dan informatie over de name server, namelijk diens IP address, en het domain waarop de machine zèlf, zich bevindt. PC implementaties hebben deze informatie vaak in algemenere configuratiefiles staan (zoals bijvoorbeeld PCTCP.INI).

Er is een aantal tooltjes beschikbaar dat gebruikt kan worden in combinatie met de DNS servers. Zo is er bijvoorbeeld het tooltje host dat gebruikt kan worden om alle IP adressen van een bepaalde host te achterhalen (roep in herinnering dat hosts *multi-homed* kunnen zijn; DNS servers zullen bij default het 'dichts bij gelegen' IP adres van de betreffende host afgeven). Verder is er nog het tooltje dig (domain internet groper) en nslookup die beiden ongeveer hetzelfde doen als host.

IIR TECHNOLOGY

6.4 DNS

- **DNS messages, zowel questions als answers, bestaan uit:**
  - ***Header*, waarin informatie is opgenomen over een aantal nog volgende velden**
  - ***Question*, waarin de vraag voor de name server staat**
  - ***Answer*, het antwoord van name server**
  - ***Authority*, zijnde een pointer naar een andere name server**
  - ***Additional*, extra informatie niet in answer opgenomen**



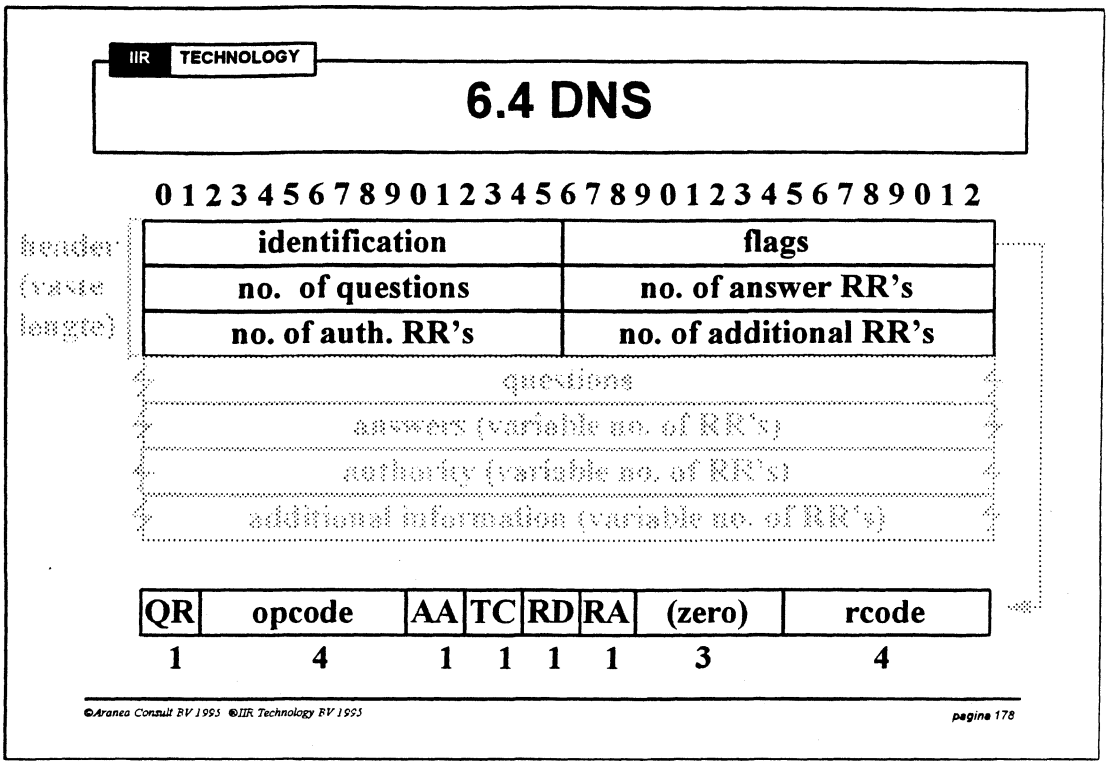
DNS messages kunnen bestaan uit vragen van clients (queries) en antwoorden van de servers (answers). Zowel vragen als antwoorden worden in een zelfde PDU verpakt, namelijk de DNS message. Op de slide staat de algemene structuur weergegeven van DNS messages, de volgende slides gaan in detail in op de velden in de DNS message.

DNS is één van de weinige TCP/IP protocollen die gebruik maakt van 'compressie'. Dat werkt als volgt. Normaliter worden domain names in een DNS message weergegeven door een opsomming van labels. Ieder label wordt voorafgegaan door een *count byte* (8 bits), het laatste label (root) heeft count '0'. Een voorbeeld om dit te verduidelijken:

```
6 g e m i n i 3 t u c 4 n o a o 3 e d u 0
```

Het is voor te stellen dat in een antwoord van een name server, meermalen één en dezelfde FQDN voorkomt, bijvoorbeeld een multi-homed host die bij eenzelfde domain name meerdere IP adressen heeft. Het is natuurlijk verspilling van bandbreedte om iedere keer weer de FQDN op te nemen. Dit kan ook vermeden worden en wel door compressie toe te passen. Door de twee hoogste orde bits van de counters op '1' te zetten, wordt daarmee aangegeven dat we niet te maken hebben met een 8 bits *counter*, maar met een 16 bits *pointer*! De 14 bits die nu nog beschikbaar zijn in de pointer verwijzen naar een offset in de DNS message waar de betreffende domain name staat.

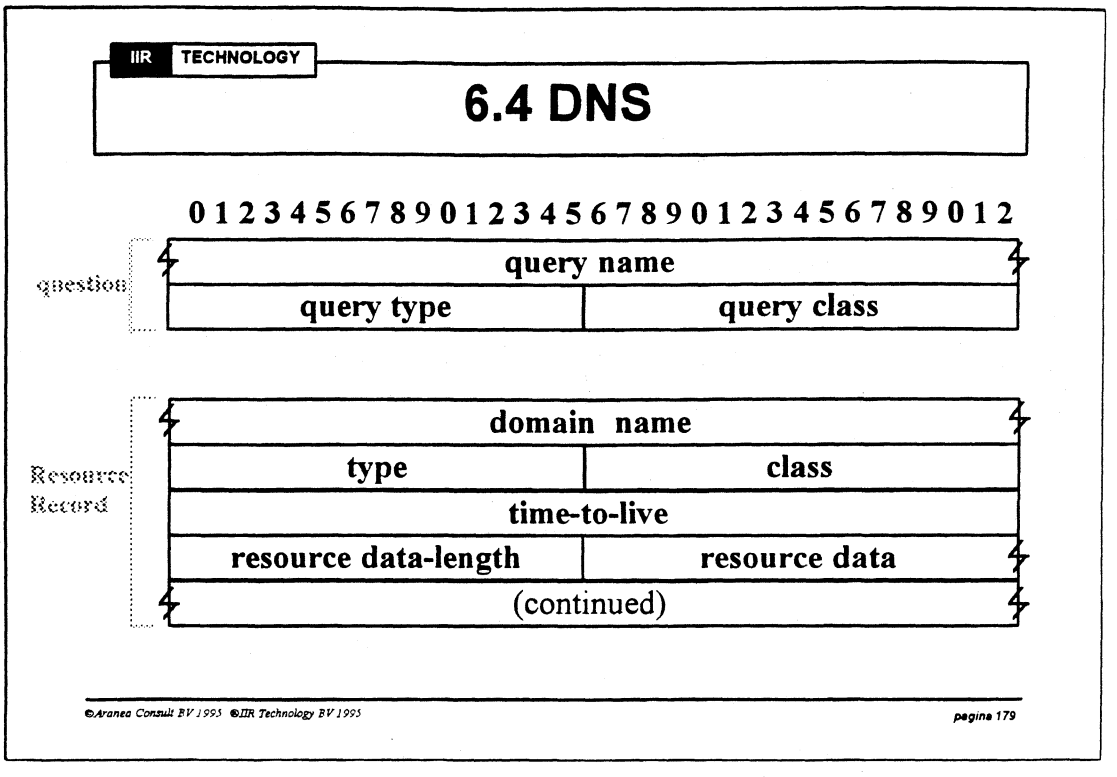
Over de slide: authority RR's specificeren de domain namen van DNS servers, en worden meestal door root servers gegeven. In de additonal RR's staan vervolgens de bijbehorende IP adressen, zodat we niet nóg een keer de name servers hoeven te consulteren. Een tweede voorbeeld van optimalisatie van het DNS protocol!



DNS messages worden gebruikt voor zowel de vragen van de resolvers, als de antwoorden van de servers. De volgende velden worden gebruikt:

1. identification - gezet door de client, gebruikt door server om vraag en antwoord te laten matchen.
2. flags - dit 16 bits veld is opgebouwd uit de volgende velden:
   - QR - dit bit geeft aan of het een query betreft ('0') of een response ('1').
   - opcode - 3 waarden gedefinieerd, te weten *standard query* (0), *inverse query* (1) en *server status request* (2).
   - AA - dit bit geeft aan of het een *authorative answer* betreft ('1').
   - TC - dit bit geeft aan of de response *trucated* is ('1'). Dit kan in het geval van UDP voorkomen als de message groter was dan 512, en door de server enkel de eerste 512 bytes worden teruggegeven (vaak herhaalt de resolver de vraag vervolgens, gebruikmakend van TCP).
   - RD - dit bit geeft aan dat *recursion* vereist is. Dit bit kan door de client gezet worden om aan te geven dat, als de server niet authorative is voor de betreffende query, de server op zoek moet gaan naar het antwoord. Als dit bit niet gezet is, zal de server met de naam van een andere server terugkomen die wel authorative is. Dit noemen we *iterative*. Bij default vragen clients Recursions Desired.
   - RA - dit bit wordt door de server gezet en betekent *recursion available*.
   - zero - veld gevuld met '0' bits.
   - rcode - dit is de return code met mogelijke waarden *no error* (0) en *name error* (3). Name error duidt erop dat de naam niet te ontsluiten is.

Na het flags veld komen vier velden die het aantal van de daarop volgende vier velden aangeven. De volgende slide behandeld het formaat van de *questions* en het formaat van de zogenaamde *Resource Records*.



Bovenstaand het formaat van de *questions* en *Resource Records* zoals deze door client en server worden gebruikt.

De *question* begint met een *query name*. Dit is de aanschakeling van labels zoals deze in de domain name zijn gedefinieerd (zie bespreking van FQDN's). Iedere question heeft een *query type*, iedere response, bestaande uit de RR's, heeft een *type*. De volgende *query types* (questions) en *types* (RR answers) zijn gedefinieerd (niet volledig!):

Naam	Waarde	description	type?	query type?	
A	1	Host address		X	X
NS	2	name server	X	X	
CNAME	5	canonieke naam (alias)	X	X	
PTR	12	pointer record (in-addr.arpa)	X	X	
HINFO	13	host info (CPU en OS)	X	X	
MX	15	mail exchange record	X	X	
AXFR	252	request for zone exchange		X	
ANY (of *)	255	request for all records		X	

Het meest gebruikt is uiteraard *query type* A, waar gevraagd wordt om een IP adres behorende bij *query name* en *query type* PTR, waar, op basis van een IP adres, de bijbehorende domain name gezocht wordt (gebruikt door host).

Het gebruikelijke *query class* in de question is IN, waarde 1, wat duidt op Internet, of ANY (of *), waarde 255.

Voor de RR's geldt: de *domain name* geeft aan waar de meegeleverde informatie betrekking op heeft. Het *type veld* is besproken, evenals het *class veld*, het *TTL veld* staat normaliter op 2 dagen (in seconden) en geeft aan hoe lang een client de informatie mag cachen, *resource data length* spreekt voor zich (bij type 1, een A record, is dit bijvoorbeeld 4), evenals *resource data*.

IIR TECHNOLOGY

6.4 DNS

- **Voorbeeld van een DNS sessie:**

```
id=0001 fl=0100
Questions
www.cec.lu 0001 0001
; size = 28
id=0001 fl=8180
Questions
www.cec.lu 0001 0001
Answers
www.cec.lu          A          1    587716 158.169.50.11
Authorities
cec.lu              NS         1    345600 tclux1.cec.lu
cec.lu              NS         1    345600 menvax.restena.lu
Additional records
tclux1.cec.lu       A          1    345600 158.169.9.11
menvax.restena.lu   A          1    345600 158.64.1.2
; size = 132
Hostent:
  h_name = www.cec.lu
  h_addrtype = 2
  h_length  = 4
  Address = 158.169.50.11
```

```
id=0002 fl=0100
Questions
info.cert.org 0001 0001
; size = 31
id=0002 fl=8580
Questions
info.cert.org 0001 0001
Answers
info.cert.org       CNAME      1    86400 cert.org
cert.org            A          1    86400 192.88.209.5
Authorities
cert.org            NS         1    86400 cert.org
cert.org            NS         1    86400 tictac.cert.org
Additional records
cert.org            A          1    86400 192.88.209.5
tictac.cert.org     A          1    86400 192.88.209.21
; size = 136
Hostent:
  h_name = info.cert.org
  h_addrtype = 2
  h_length  = 4
  Address = 192.88.209.5
```


Hierbij hebben de verschillende items de volgende betekenis:

```
www.cec.lu          IN    A            1     587716 158.169.50.11
```

www.cec.lu	owner name
IN	address class (Internet)
A	type (Adress)
587716	TTL
158.169.50.11	dotted-decimal IP adress
CNAME	canonical name (alias)
NS	Name server identification

IIR TECHNOLOGY

7. Het Internet

7.1 Inleiding
7.2 Service providers
7.3 Tools

Een van de redenen voor de populariteit van TCP/IP is natuurlijk het Internet. Daarom zal er een apart stukje van de cursus gewijd worden aan Internet.

Omdat dit een TCP/IP cursus is, en geen Internet cursus, wordt er niet diep op de diverse aspecten van Internet ingegaan.

IIR TECHNOLOGY

7.1 Inleiding

- **'Het' Internet is het wereldwijde netwerk van netwerken dat benaderd kan worden middels het TCP/IP protocol**
- **in augustus '94:**
  - **37.000 IP netwerken gekoppeld**
  - **3,2 miljoen computer hosts**
  - **46.000 domains**
  - **83 landen rechtstreeks aangesloten**
  - **154 landen middels gateways te bereiken**

Het wereldwijde Internet begint inmiddels astronomische proporties aan te nemen. Ooit begonnen als Universiteits- en DoD-netwerk in de Verenigde Staten, is het inmiddels uitgegroeid tot een wereldwijd netwerk waar vrijwel alle landen uit de westerse wereld op zijn aangesloten. Op de slide een aantal kengetallen.

Opmerkelijk is dat in Nederland, na de Verenigde Staten, relatief gezien de meeste aansluitingen op Internet zijn, en dat Nederland een van de eerste landen was die een Internet verbinding met de VS hadden (dank zij de universiteiten hier !).

IIR TECHNOLOGY

7.2 Service providers

- **Twee manieren om te koppelen aan het Internet**
  - **via een service provider**
  - **via een router koppeling**
- **service providers: NLNet, EUNet, TU Eindhoven, PTT, IBM, ...**
- **koppeling via routers: zèlf verantwoordelijk voor adressenbeheer, security, etcetera.**
- **koppeling via service providers: dial-up link**

Er zijn feitelijk twee manieren om te koppelen aan het Internet.

- Dit kan door een abonnement te nemen bij een zogenaamde service provider. Dit is een organisatie die zèlf toegang heeft tot het Internet, en vervolgens 'abonnementen' verkoopt aan derden om deze op die manier ook toegang te verlenen tot het Internet. Service providers zijn zo'n beetje de enige manier voor particulieren om op het Internet te komen. Immers, particulieren hebben doorgaans geen IP netwerk nummer... Service providers maken vaak onderscheid tussen single entries en business entries. De business entries zijn bedoeld voor bedrijven die zelf geen Internet aansluiting hebben (of willen). Het voordeel van een business entry is natuurlijk het beheer van één en ander. Dit blijft bij de service provider. Buitenstaanders hebben geen enkele mogelijkheid om via het Internet het bedrijfsnetwerk te penetreren.
- Behalve een abonnement bij een service provider kan een bedrijf er ook voor kiezen om een eigen routerkoppeling aan het Internet in te richten. In Nederland betekent dat vrijwel altijd koppelen aan Surfnet, het academische netwerk. Vervolgens is het bedrijf zelf verantwoordelijk voor een juiste werking van het eigen TCP/IP netwerk, het eigen domain naming system, etcetera. Vaak resulteert dit ook in de installatie van een zogenaamde fire wall. Een fire wall kan beschouwd worden als een bescherming tegen alle dreiging van buitenaf, èn van binnenuit. Met een fire wall kan het verkeer over de koppelende router heen geregeld worden.

IIR TECHNOLOGY

7.3 Tools

- **Na toegang op het Internet gekregen te hebben: *the world is yours!***
- ***Relatief* weinig kennis van TCP/IP of het Internet noodzakelijk**
- **Diensten die beschikbaar komen:**
  - **E-mail (leveranciers, partners, collega's)**
  - **File transfer (leveranciers, partners, collega's)**
  - **News (het gehele Internet...)**

Als de koppeling met het Internet eenmaal gerealiseerd is, komt er een schat aan informatie vrij! Het is nu immers mogelijk om 'de hele wereld' te bereiken. De twee meest in het oog springende functionaliteiten die nu beschikbaar komen, zijn natuurlijk electronic mail en file transfer. Denk hierbij vooral aan het uitwisselen van informatie (e-mail en file transfer) met leveranciers, collega bedrijven en collega's in andere vestigingen.

IIR TECHNOLOGY

7.3 Tools

- **E-mail: SMTP, EMSTP of MIME?**
- **File transfer: FTP, Archie, WAIS, Veronica of WWW?**
- **WWW kan als 'front end' dienen voor News, WAIS en Gopher en is zéér populair aan het worden**
- **Implementaties van WWW: *Mosaic* en *Netscape***

De beschikbare electronic mail *protocollen* zijn reeds besproken in hoofdstuk 3. Variërend van het oude, vertrouwde SMTP tot het moderne multi-mediale MIME (multi-purpose internet mail extensions).

De beschikbare file transfer protocollen zijn ook aan verandering onderhevig. Uiteraard wordt er nog steeds gebruik gemaakt van het oude, vertrouwde FTP (voornamelijk anonymous FTP), maar met name WWW (World Wide Web) lijkt de slag om het meest gebruikte information retrieval tool te gaan winnen. Met WWW krijgt de gebruiker een verbinding met een WWW-server waarop zich een hyptertext database bevindt. De zogenaamde WWW-pages worden verzonden naar de gebruikerscomputer (middels http protocol, hyper-text transport protocol) en daar getoond. Door te klikken op de hyptertext links, kan er weer een nieuwe pagina worden opgehaald, etcetera. Deze manier van zoeken is uiteraard vriendelijker dan het gebruik van FTP!

Andere protocollen in de information retrieval sfeer zijn Archie (overzicht van duizenden anonymous FTP servers over het gehele Internet; zoeken op keyword in file name mogelijk), WAIS (Wide Area Network Information Servers; servers die informatie bevatten over *files* en *databases* die bepaalde keywords bevatten, dus niet de naam van de file of database), Gopher (menu driven interface voor Archie, FTP en WAIS) en Veronica (Very Easy Rodent-Oriented Newide Index to Computerized Archives; overzicht van allerhande Gopher servers). Voor zowel Gopher als Veronica geldt, dat gebruik moet worden gemaakt van Gopher/Veronica servers. Deze zijn echter meestal onbereikbaar.

IIR TECHNOLOGY

7.4 Nadelen

- **De populariteit van Internet leidt tot een aantal nadelen ervan:**
  - ⇨**Het Internet begint dicht te slibben**
  - ⇨**Grote kans op malafide praktijken**
  - ⇨**Informatie wordt steeds minder toegankelijk**
  - ⇨**Vercommercialisering van Internet**

De grote belangstelling voor Internet heeft ook nadelen. Het grootste nadeel van de grote populariteit is de afname in de performance van Internet. Een groot aantal lokaties is gedurende de kantooruren al niet of nauwelijks meer te bereiken. 's Avonds of 's nachts werken kan soelaas bieden, maar heeft als nadeel dat dan in Amerika gewerkt wordt.

Ook wordt het steeds lastiger om op het Internet aangesloten computers te beveiligen tegen ongeautoriseerd gebruik. Hoe meer mensen gebruik van Internet maken, hoe groter de kans op hackers is. Veel ondernemingen zijn er daarom al toe overgegaan om zogeheten fire-walls te implementeren. Dit zijn gateways tussen interne netwerken en het Internet die, door de standaard applicatie protocollen te vervangen door eigen protocollen (proxies), allerlei controles kunnen uitvoeren alvorens iemand van het interne netwerk toegang te geven tot Internet en vice versa.

Eenmaal aangesloten op Internet komen steeds meer gebruikers er achter dat, hoewel er een schat aan informatie is opgeslagen, deze voor de gemiddelde gebruiker toch moeilijk te doorzoeken is, ondanks de mooie grafisch user-interface als Netscape of Mosaic.

IIR TECHNOLOGY

8. TCP/IP implementaties

- **TCP/IP verkrijgbaar voor iedere platform**
- **TCP/IP dè oplossing voor connectivity problemen (?)**
- **Ook leveranciers van 'traditionele protocollen' laten deze los cq. ondersteunen, naast proprietary protocollen, de TCP/IP stack**

Tegenwoordig is er voor ieder hardware platform wel een TCP/IP stack beschikbaar. Hierdoor kan met TCP/IP gerealiseerd worden wat de bedoeling van OSI was: iedere computer kan communiceren met ieder andere computer, ongeacht het merk.

Zelfs IBM, die toch jarenlang alleen SNA ondersteunde, biedt volwaardige TCP/IP implementatie voor mainframes en minicomputers. Ook de internetworking devices van IBM, zoals de 6611, bieden volledige IP ondersteuning.

Tot 1993 stelde de US en UK overheid nog als eis dat alleen hardware leveranciers die OSI compliant waren, mochten leveren. Sinds 1993 is deze eis veranderd en moeten de leveranciers TCP/IP ondersteunen.

De vraag is dan ook of met TCP/IP alle connectiviteits problemen opgelost kunnen worden, en hoe TCP/IP gebruikt kan worden op systemen die dat van oudsher niet ondersteunen.

IIR TECHNOLOGY

8. TCP/IP implementaties

- **Connectivity in PC LAN omgevingen (IBM Mainframe, AS/400, VAX/VMS, Unix dialecten):**
  - ➢**Meerdere 'kijk-dozen' per type computer...**
  - ➢**multiple protocol stacks...**
  - ➢**gateways...**
  - ➢**...of standaardiseren op TCP/IP**

In organisaties waar meerdere typen 'host-computers' staan, zijn er verschillende oplossingen denkbaar om vanaf client(PC')s deze systemen te benaderen.

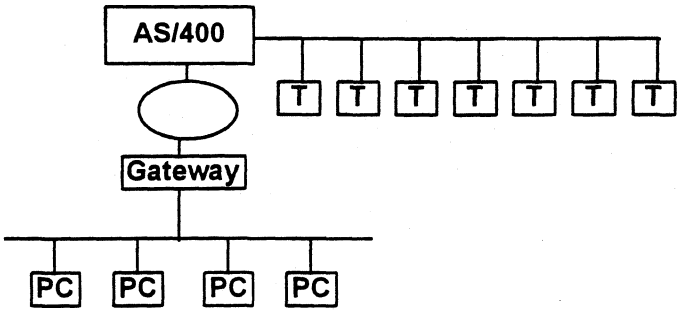
- per host een specifieke terminal op het bureau
- de PC uitrusten met verschillende protocol stacks per type host systeem
- inzetten van gateways om de vertaalslag te gaan maken

Een andere oplossing zou kunnen zijn het standaardiseren op TCP/IP, aangezien dit protocol door alle host machines ondersteund wordt.

IIR TECHNOLOGY

8. TCP/IP implementaties

- **Voorbeeld van een omgeving waarin op TCP/IP gestandaardiseerd zou kunnen worden:**



Hierboven is de situatie geschetst zoals deze voor de invoering van TCP/IP zou kunnen zijn.

In het LAN wordt een native protocol gebruikt, bijvoorbeeld IPX in een Novell omgeving, NetBEUI in een LAN Manager omgeving of Vines IP in een Banyan omgeving.

Op de PC's is voor de communicatie met de AS/400 software geladen die invulling geeft aan de toepassingsgerichte lagen van het OSI model (applicatie, presentatie en sessielaag). De PC's 'praten' dus met de AS/400 in de eigen AS/400 taal. Alleen vindt het transport van de gegevens plaats over de native LAN protocollen.

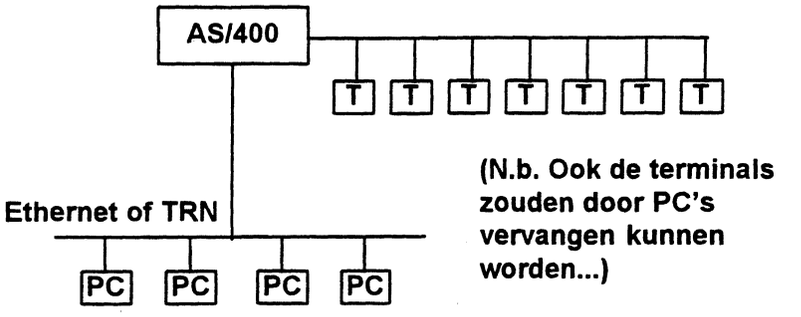
De taak van de gateway is om de native LAN transport protocollen om te zetten in de voor de AS/400 omgeving specifieke transport protocollen, in dit geval 5250 datastreams.

Hierdoor kan er alleen met de AS/400 gecommuniceerd worden als de juiste, leveranciersafhankelijke protocollen op de PC's geladen zijn. Zou er vanuit de PC's naar een tweede omgeving gecommuniceerd moeten worden, dan zouden dus andere protocollen geladen moeten zijn.

IIR TECHNOLOGY

8. TCP/IP implementaties

- **Vanaf versie 3, release 1, ondersteunt de AS/400 TCP/IP (geïntegreerd onderdeel van kernel)**



Als echter iedere fabrikant gebruikt maakt van standaard TCP/IP applicatie protocollen, zoals Telnet, FTP en NFS, en van TCP en IP als transport protocollen, dan kan een PC, mits daar een TCP stack op aanwezig is, communiceren met iedere in het netwerk aanwezige minicomputer.

Het voordeel is dat de keuze voor de software op de PC's onafhankelijk wordt van het soort minicomputer waarmee gecommuniceerd moet worden. Tevens vervalt de noodzaak om een gateway te gebruiken.

IIR TECHNOLOGY

8. TCP/IP Implementaties

- **Onderscheid in TCP/IP-implementaties**

  - **'Pure' TCP/IP produkten**
    ('OnNet' van FTP Software, 'IPSwitch' van Acadia, 'SuperTCP' van Frontier Technology, etc.)

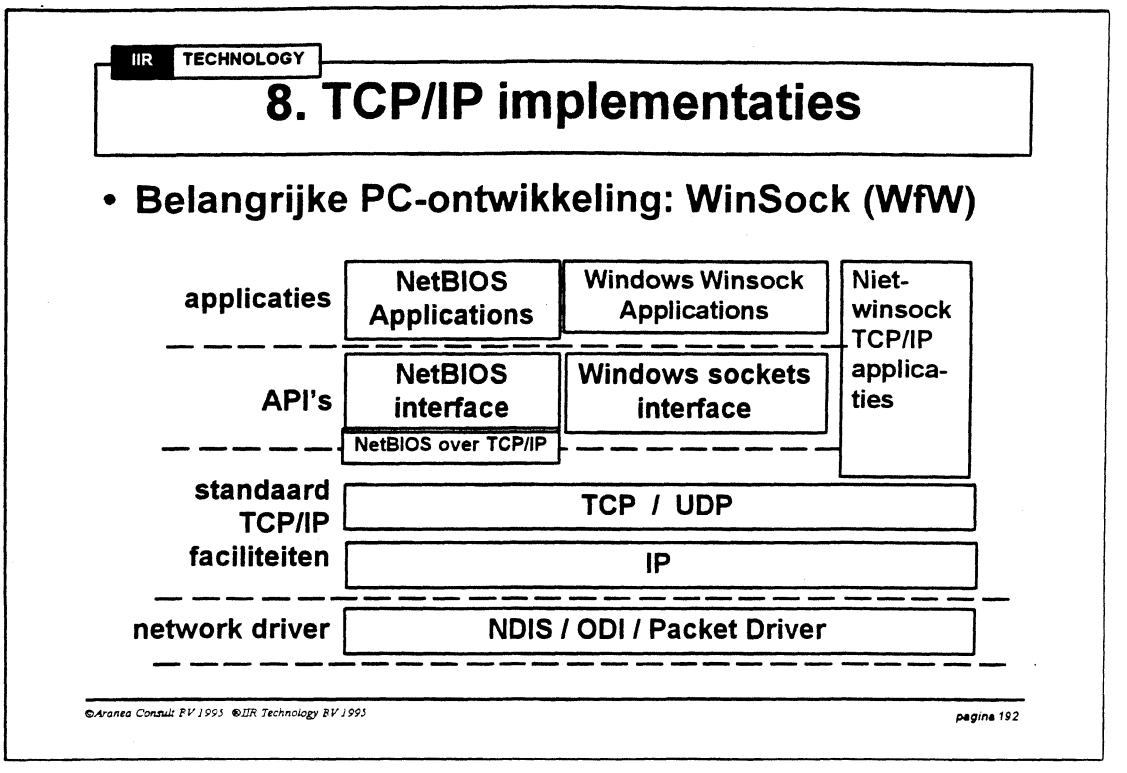
  - **'LAN OS-gerelateerde' TCP/IP produkten**
    ('IPClient' van Banyan VINES, 'PC-NFS' van Sun, 'Pathworks TCP/IP' van Digital, etc.)

  - **'Freeware' TCP/IP produkten**
    ('MS TCP/IP' voor WfW, 'Trumpet Winsock', 'Twinsock', e.a.)

Voor zover je al een onderscheid zou willen maken in de verschillende TCP/IP produkten op de markt, zou *een* onderscheid gemaakt kunnen worden naar produkten die niet specifiek gerelateerd zijn aan wat hier gemakshalve maar even genoemd wordt 'LAN Operating Systemen', en produkten die dat wel zijn.

Produkten die wel gerelateerd zijn aan een LAN Operating System, bieden vaak bepaalde specifieke functionaliteit die voor dat betreffende operating system noodzakelijk is. Zo kan het produkt IPClient gebruikt worden om VINES/IP pakketjes in te pakken en als IP pakketje (protocol nummer 83) te versturen over de infrastructuur. Als deze functionaliteit niet gewenst is, dan zou gekeken moeten worden of dit produkt dan wel voldoet onder de gegeven omstandigheden. Immers, in het produkt zit functionaliteit geprogrammeerd die niet gebruikt wordt en dus tot overhead (performance verlies) kan leiden.

Opgemerkt moet worden dat via Internet veel freeware (gratis) TCP/IP stacks verkrijgbaar zijn. De bekendste is Trumpet's Winsock met ondersteuning van SLIP en PPP. Ook Microsoft heeft voor WfW en Windows95 een gratis TCP/IP stack, echter nog zonder SLIP of PPP ondersteuning. Voor deze stacks geldt over het algemeen dat er slechts zeer elementaire applicatie services bij geleverd worden. Vanwege de Winsock interface is dat echter meestal geen probleem.



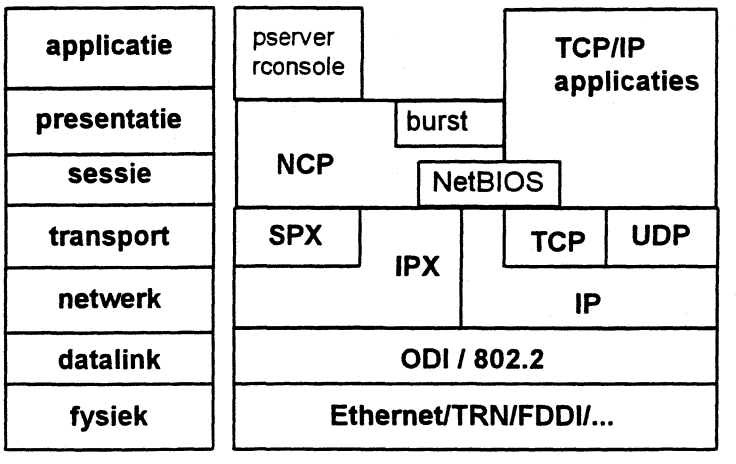
Eén van de belangrijkste ontwikkelingen op het gebied van TCP/IP connectivity op de PC is het ontwikkelen van de WinSock standaard. Hiermee is er een een interface (API) gedefinieerd in de MS Windows omgeving, dat *sockets* ondersteunt, vergelijkbaar met de Unix omgeving. Door deze standaard tot industrie-standaard te maken, kunnen leveranciers applicaties voor laag 7 gaan bouwen, die voldoen aan de Winsock standaard, en kunnen zich daarmee beperken en concentreren op enkel de applicatielaag software.

Wat je vervolgens ziet gebeuren is inderdaad dat er leveranciers opstaan die zich enkel nog op de bovenkant (laag 7) richten. Immers, produkten als 'Windows for Workgroups'(zie ook bovenstaande figuur) en Windows95 hebben standaard TCP/IP aan boord, dat wil zeggen, de communicatie gerichte lagen van TCP/IP, die 'naar boven' Winsock praten. Met andere woorden, waarom zou je als leverancier nog tijd, geld en moeite steken in het ontwikkelen van TCP/IP communicatie gerichte software als deze al (gratis) wordt meegeleverd? Het lijkt veel aantrekkelijker om relatief goedkope en goede terminal emulatoren, file transfer programmatuur, etcetera, op de markt te brengen. Op dit moment zijn er goede, maar eenvoudige telnet, FTP en SMTP winsock client toepassingen beschikbaar.

IIR TECHNOLOGY

8. TCP/IP implementaties

- **Multi-protocol oplossing in Novell NetWare:**



Bovenstaand de oplossing die Novell gebruikt om in een gemengde omgeving zowel gebruik te kunnen maken van TCP/IP, als van IPX/SPX. Daarmee wordt getracht de voordelen van IPX/SPX te combineren met de flexibiliteit van TCP/IP, maar het nadeel is de nog steeds dubbele protocolstack die nodig is op het werkstation.

Tegenwoordig kan bij Novell gekozen worden om ofwel IPX te gebruiken voor het transport van NCP berichten, ofwel IP. In dat laatste geval hoeven de werkstations slechts uitgerust te worden met een enkele protocolstack: TCP/IP. Uiteraard blijft wel de implementatie van NCP (Netx) nodig om de specifieke Netware functies te kunnen gebruiken. Overigens kan een Novell file-server (nog) niet fungeren als FTP of Telnet server.

IIR TECHNOLOGY

9. IP Ontwikkelingen

- **Adressering IP begint 'uit zijn voegen te barsten'**
- **32-bits adressen waren lange tijd voldoende, blijken nu te klein te zijn**
- **Aantal voorlopers (o.a. TUBA, SIPP, CATNIP), uiteindelijk geresulteerd in:**

IPng

Eerder zijn al de tekortkomingen van het huidige IP protocol aan de orde geweest. De belangrijkste tekortkoming is natuurlijk de veel te beperkte adresruimte, maar ook de wijze van adressering is een tekortkoming van het huidige IP protocol.

Daarnaast is de vaste overhead van de IP header, waarvan een groot deel niet of nauwelijks gebruikt wordt, ook voor verbetering vatbaar.

Tenslotte is ook de beveiliging (authentificatie van IP verkeer, versleuteling van data e.d.) iets dat meegenomen moet worden bij de vaststelling van IPng.

IIR TECHNOLOGY

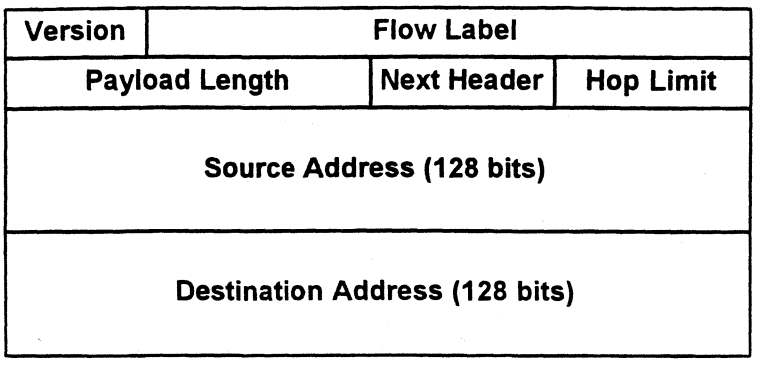
9. IP Ontwikkelingen

- IPng, ook IPv6, is de opvolger van IPv4
- IPng biedt:
  - Uitgebreide routing- en adresseringsmogelijkheden
  - Vereenvoudigde laag-3 header
  - Verbeterde 'options' mogelijkheden
  - QOS parameters mogelijk voor bepaalde pakketjes
  - Authentificatie mogelijkheden

IIR TECHNOLOGY

9. IP Ontwikkelingen

- **De header van IPv6 gaat er (waarschijnlijk!) als volgt uit zien:**



Version	4-bits getal, versie nummer, in deze versie dus nummer 6
Flow Label	28-bits getal. Gerelateerd aan QOS
Payload Length	16-bits getal, lengte van de rest van het IP pakket (bytes)
Next Header	8-bits getal, vergelijkbaar met 'Protocol'-veld in IPv4
Hop Limit	8-bits getal, vergelijkbaar met 'Hop count'-veld in IPv4
Source Address	128-bits veld, adres van de afzender
Dest. Address	128-bits veld, adres van de geadresseerde van dit specifieke pakket, wat niet overeen hoeft te komen met het adres van de uiteindelijke bestemming!

IIR TECHNOLOGY

9. IP Ontwikkelingen

- 'Options' worden in IPv6 anders afgehandeld dan in IPv4
- Options krijgen *aparte header*, tussen IP header en Transport header
- Options hoeven niet (allemaal) door routers afgehandeld te worden (performance!)

Option	Function
Routing	Extended routing (vgl. Loose SR)
Fragmentation	Fragmenation and reassembly
Authentication	Integrity and Authentication
Security Encaps.	Confidentiallity
Hop-by-hop	Hop-by-hop processing
End-to-end	End-to-end processing

IIR TECHNOLOGY

9. IP Ontwikkelingen

- IPng adressen 128 bits (t.o.v. oorspronkelijke 32 bits adressen)
- 15% van de nieuwe adresspace gealloceerd!
- 3 soorten adressen
  - »unicast adressen
  - »cluster adressen (groepsadres, packet wordt naar één van de groepsleden gestuurd)
  - »multicast adressen (groepsadres, packet wordt naar alle groepsleden gestuurd)

Om straks voor niet te veel conversie problemen te zorgen, moet er rekening gehouden worden met de huidige IP adressen. Het voorstel is om de laatste drie bytes van het nieuwe adres gelijk te laten zijn aan het huidige IP adres. Op die manier kan er eenvoudig een conversie van oud naar nieuw adres worden uitgevoerd.

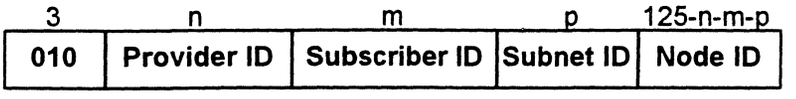
Ook moet bij IPng rekening gehouden worden met mobiele communicatie. Daarbij wordt het erg lastig om op locatie gebaseerde adressen te gebruiken. Er is dus tevens een mechanisme bedacht om adressen dynamisch toe te kennen. De impact daarvan is vrij groot. Denk maar eens aan de DNS systemen die nu een variabel adres bij een hostnaam moeten opslaan.

IIR TECHNOLOGY

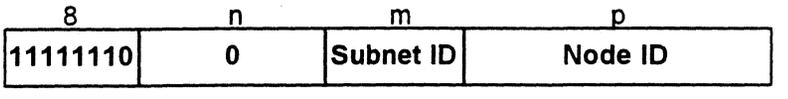
9. IP Ontwikkelingen

- **Enkele voorbeelden van unicast adressen:**

»Provider based unicast adressen



»Local-Use adressen



IIR TECHNOLOGY

9. IP Ontwikkelingen

- Routing in IPng is vergelijkbaar met CIDR
- Toevoegingen aan IPng routing hebben tot gevolg:

  »Provider selection (op basis van performance, cost, policy, etcetera)

  »Host mobility (route naar de huidige lokatie)

  »Auto-readdressing (route naar nieuwe adressen)

IIR TECHNOLOGY

Opgaven - netwerklaag

1) De IP checksum beslaat alleen de IP header en niet de data. Wat is hiervan het voordeel? En wat is het nadeel?

2) Wat doet een router met het Time to Live veld in een IP header? Waarom?

3) Is het mogelijk om een datagram te sturen naar een router's IP adres? Heeft dit nut?

4) Een machine heeft twee interfaces met de internet adressen I1 en I2. Is het voor deze computer mogelijk om een datagram, dat bedoeld is voor I2, te ontvangen over het netwerk met adres I1? Verklaar.

5) Speel detective: na het monitoren van het netwerkverkeer op een LAN gedurende 10 minuten, valt het iemand op dat alle frames bedoeld voor host A, IP datagrammen bevatten met A's IP adres, terwijl alle frames voor host B IP datagrammen met een IP adres niet gelijk aan B's adres bevatten. Leg dit uit.

Antwoorden:

1.

2.

3.

4.

5.

IIR TECHNOLOGY

Opgaven - netwerklaag (vervolg)

6) Hoeveel netwerken kunnen er bestaan in klasse A, B en C netwerken? En hoeveel hosts?

7) Zou ARP zijn cache moeten bijwerken als er al een entry bestaat voor een bepaald IP adres? Waarom of waarom niet?

8) ARP wordt vaak genoemd als een zwak punt m.b.t. de security. Leg dit uit.

9) Voor de beveiliging van netwerken wordt vaak gebruik gemaakt van Proxy ARP. Waarom ?

10) Waarom is er voor frame-relay wel een IARP methode en voor X.25 niet ?

11) Hoewel ICMP 'bovenop' IP draait, wordt het niet beschouwd als een transportlaag protocol. Waarom niet ?

12) Zou het mogelijk zijn routers alleen op basis van IP adressen te laten werken, zonder dat zij de extra route-informatie hebben van RIP, OSPF of andere routing protocollen ?

Antwoorden:

6. _____

7. _____

8. _____

9. _____

10. _____

11 _____

12 _____

IIR TECHNOLOGY

Opgaven - transportlaag

1) Waarom heeft UDP een eigen checksum terwijl ook IP een checksum heeft? Is er bezwaar tegen om één checksum in te voeren voor het complete IP datagram inclusief het UDP bericht?

2) Het niet gebruiken van checksums kan zeer gevaarlijk zijn. Leg uit hoe een enkel ARP pakket van host A het onmogelijk kan maken om een andere host Q te bereiken.

3) Wat is het grote voordeel van well known ports voor UDP? En welk nadeel heeft het?

4) Eén van de TCP opties staat een zender toe om de maximale segment grootte te bepalen die hij wil ontvangen. Waarom ondersteunt TCP deze optie terwijl het ook al een maximale window grootte ondersteunt?

5) Verloren geraakte acknowledgements forceren niet noodzakelijk retransmissions. Leg uit waarom niet.

Antwoorden:

1. _____

2. _____

3. _____

4. _____

5. _____

Afkorting	*Betekenis*	*Instelling*
2B1Q	Euro_ISDN codering voor U-interface (ISDN)	ETSI
4B5B	Vier bits/vijf toestanden codering (FDDI)	ANSII
5B6B	Vijf bits/zes toestanden codering	ITU
AAL	ATM Adaption Layer	ITU
AARP	Appletalk Address Resolution Protocol	AppleTalk
ABIC	Adaptive Bilevel Image Compression	IBM
ABM	Asynchronous Balanced Mode (HDLC)	ISO
ACF	Access Control Field	
ACK	Acknowledgement (BSC, TCP, e.a.)	Diverse
ACSE	Association Control Service Element (OSI)	ISO
ADM	Add/drop Multiplexer (SDH, ATM)	ITU
ADPCM	Adaptive Differentiel Pulse Code Modulation 32kb/s	ITU
ADSP	Apple Data Stream Protocol	Apple
AEP	AppleTalk Echo Protocol	Apple
AFP	Apple Filing Protocol	Apple
AM	Amplitude Modulation	ITU
AMI	Alternate Mark Inversion (lijncodering)	ITU
AMP	Adapter Management Protocol	3Com
AMPS	Advanced Mobile Phone System	
AMT	Address Mapping Table	Apple
ANSI	American National Standards Institute	ANSI
API	Application Programming Interface	Microsoft
APPC	Advanced Program to Program Communications, LU6.2 (SNA)	
APPN	Advanced Peer to Peer Networking (LU6.2+PU2.1)	IBM
ARP	Address Resolution Protocol (TCP/IP)	IETF
ARQ	Automatic Repeat Request (HDLC e.a.)	ISO
ASCII	American Standard Code for Information Interchange	ANSI
ASK	Amplitude Shift Keying	ITU
ASN.1	Abstract Syntax Notation #1 (SMTP, X.226)	ISO
ASP	AppleTalk Session Protocol	Apple
ATM	Asynchronous Transfer Mode	ITU
ATM	Automatic Teller Machine (Banking)	Banken
ATP	Appletalk Transaction Protocol	Apple
BA	Basic Access (ISDN, 2B+D)	ITU
BALUN	BALanced-UNbalanced adapter (ICS)	IBM
BCC	Block Check Character (BSC)	IBM
BECN	Backward Explicit Congestion Notification (Frame Relay)	ITU

BER	Bit Error Ratio (FR, ATM e.a.)	ITU
BISDN	Broadband ISDN	ITU
BISYNC	BInary SYnchronous Control (BSC)	IBM
BOC	Bell Operating Compagnies (Telecom)	
BOM	Begin Of Message (ATM)	ITU
BOND	Bandwith On Demand	
BOOTP	BOOTstrap Protocol (TCP/IP)	IETF
BPS	Bits Per Second	ITU
BRI	Basic Rate Interface (2B+D ISDN)	ITU
BSC	Binary Synchronous Communication (Bysinc	IBM
BSD	Berkeley System Distribution (Unix)	OTF
CAS	Channel Associated Signalling	ITU
CASE	Common Application Service Element (OSI)	ISO
CAU	Controlled Access Unit (TRN)	IBM
CBDS	Connectionless Broadband Data Service Europese SMDS)	ETSI
CBR	Constant Bit Rate	ITU
CCITT	Commite Consultatif International du Telegraphique et Telephonique, voormalig ITU-T	ITU
CCS	Common Communication Support (SAA)	IBM
CCS	Common Channel Signalling	ITU
CCSS7	Common Channel Signaling System No.7 (ISDN)	ITU
CDDI	Copper Distributed Data Interface (FDDI over Copper)	
CICS	Customer Information Control System	IBM
CIDR	Classless Inter Domain Routing (TCP/IP)	IETF
CLLM	Consolidated Link Layer Management	
CIR	Commited Information Rate (FR)	
CLNP	ConnectionLess Network Protocol (OSI)	ISO
CLNS	ConnectionLess Network Service (OSI)	ISO
CLP	Cell Loss Priority (ATM)	ITU
CMIP	Common Management Information Protocol (OSI)	ISO
CMISE	Common Management Information Service Element (OSI)	ISO
CMOT	CMIP Over TCP (TCP/IP)	ISO
COM	Continuation Of Message (ATM)	ITU
CONP	Connection Oriented Network Protocol	ISO
CPCS	Common Part Convergence Sublayer (AAL)	ITU
CPI	Common Programming Interface (SAA)	IBM
CRC	Cyclic Redundancy Check	ITU
CS	Convergence Sublayer (ATM)	ITU

CSMA/CA	Carrier Sense Multiple Access/Collission Avoidance	IEEE
CSMA/CD	Carrier Sense Multiple Access/Collission Detect (Ethernet)	IEEE
CSU	Channel Service Unit (HSSI)	ITU
CTS	Clear To Send (V.24)	ITU
CUA	Common User Access (SAA)	
DA	Destination Adress	ISO
DCC	Data Communication Channel)	ITU
DCC	Data Communication Controller (UART)	
DCE	Data Circuit-terminating Equipment	ITU
DCN	Data Communication Network (OSI)	ISO
DDCMP	Digital Data Communication Message Protocol	DEC
DDN	Defense Data Network	
DDP	Datagram Delivery Protocol (Appletalk)	Apple
DES	Data Encription Standard	DoD
DHCP	Distributed Host Configuration Protocol (TCP/IP)	IETF
DIS	Document Information System	
DIX	Digital Inter Xerox (Ethernet)	
DLC	Data Link Control protocol	ITU
DLCI	Data Link Connection Identifier (ISDN, Frame Relay)	ITU
DLE	Data Link Escape (BSC)	IBM
DLSw	Data Link Switching	IBM
DNA	Digital Network Architecture	DEC
DNS	Domain Name Server (TCP/IP)	IETF
DNS	Domain Name System (X.500)	ITU
DoD	Department of Defense	
DPA	Demand Protocol Architecture	3Com
DQDB	Distributed Queue Dual Bus, IEEE 802.6	IEEE
DS	Digital Section (B-ISDN)	ITU
DSAP	Destination Service Access Point (IEEE 802.2)	IEEE
DSR	Data Set Ready (BSC)	IBM
DSU	Data Service Unit	ITU
DTAM	Document Transfer and Access Methode	OSI
DTE	Data Terminal Equipment	ITU
DVA	Distance Vector Algorithm (routing)	
DXI	Data Exchange Interface (ATM)	ITU

E.164	International adressing scheme	ITU
EADPCM	Embedded ADPCM (ATM)	ITU
EBCDIC	Extended Binary Coded Decimal Interexchange Code	IBM
ED	End Delimiter	ITU
EDI	Electronic Data Interchange	
EEHLLAPI	Extended E? High level Language Application Programming Interface (DOS)	
EGP	Exterior Gateway Protocol (TCP/IP)	IETF
EHLLAPI	Extended High level Language Application Programming Interface (OS/2)	IBM
ELAP	Ethernet Link Access Protocol (AppleTalk)	Apple
ENQ	Enquiry (BSC)	IBM
EOM	End Of Message (ATM)	ITU
EOT	End-of-transmission (BSC)	IBM
ES	End System (OSI)	ISO
ESES	End System to End System protocol (OSI)	ISO
ESIS	End System to Intermediate System protocol (OSI)	ISO
ETB	End-of-text Block (BSC)	IBM
ETSI	European Telecommunications Standards Institute	
ETX	End-of-Text (BSC)	IBM
FCS	Frame Check Sequence (diversen)	
FDDI	Fibre Distributed Data Interface	ANSI
FDM	Frequency Division Multiplexing	
FEC	Forward Error Correction (FR)	ITU
FECN	Forward Explicit Congestion Notification (FR)	
FEP	Front End Processor	IBM
FM	Frequency Modulation	
FR	Frame Relay	
FS	Frame Status (802.5)	IBM
FSK	Frequency Shift Keying	
FTAM	File Transfer, Access and Management (OSI)	ISO
FTP	File Transfer Protocol (TCP/IP)	IETF
GGP	Gateway-to-gateway Protocol	IETF
GSM	Groupe Speciale Mobile/Global System for Mobile comm.	ETSI
GOSIP	Government OSI Profile	
GSVC	Global Signalling VC (ATM)	ITU
GSVCI	GSVC Identifier	ITU

HDB3	High Density Bipolar - 3 zeros (CCITT)	ITU
HDLC	Highlevel Data Link Control (OSI)	ITU
HDSL	High Speed digital Subscriber Line (ITU)	ITU
HEC	Header Error Control (ATM)	ITU
HIPPI	High Performance Parallel Interface	ITU
HSLAN	High-Speed LAN	ITU
HSSI	High Speed Serial Interface	ITU
ICMP	Internet Control Message Protocol (TCP/IP)	IETF
ICS	IBM Cabling System (TRN)	IBM
IDN	International Data Number (X.121)	ITU
IDRP	Iter Domain Routing Protocol (OSI)	OSI
IGP	Interior Gateway Protocol (TCP/IP)	IETF
ILMI	Interim Local Management Interface (ATM)	ITU
IMS	Information Management System	IBM
IP	Internet Protocol (TCP/IP)	IETF
IPX	Internetwork Packet eXchange (Netware)	Novell
IS	Intermediate System (OSI)	ISO
ISDN	Integrated Services Digital Network	ITU
ISIS	Intermediate System to Intermediate System protocol (OSI)	ISO
ISO	International Organisation for Standarization	
ITU	International Telecommunication Union (-T Telecommunications)	
JTM	Job Transfer and Manipulation (OSI)	ISO
LAN	Local Area Network	
LAPB	Link Access Protocol, Balanced (CCITT)	ITU
LAPD	Link Access Protocol, ISDN-channel D (CCITT)	ITU
LAPX	Link Access Protocol, eXtended (CCITT)	ITU
LAT	Local Area Transport	DEC
LCI	Logical Channel Identifier (X.25)	ITU
LCN	Logical Channel Number (X.25)	ITU
LLAP	Localtalk Link Access Protocol (Appletalk)	Apple
LLC	Logical Link Control	IEEE
LRC	Longitudal Redundancy Check - BCC (diversen)	
LSA	Link State Algorithm (routing)	
LSB	Least Significant Bit/Byte	
LU	Logical Unit (SNA)	IBM

MAC	Media Access Control	IEEE
MAN	Metropolitan Area Network	IEEE
MAP	Manufactors Automation Protocol (General Motors)	ISO
MAU	Multistation Access Unit (TokenRing)	IBM
MAU	Medium Attachment Unit (Ethernet)	IEEE
Mbps	Mega Bits Per Second	
MHS	Message Handling System (X.400)	ITU
MIB	Management Information Base (OSI, SNMP)	ISO
MID	Multiplexing Identifier (ATM)	ITU
MII	Media-Independent Interface (100BaseT)	IEEE
MIME	Multipurpose Internet Mail Extensions (TCP/IP)	IETF
MIPS	Milion Instructions Per Second	
MIR	Maximum Information Rate (FR, ATM e.a.)	ITU
MSB	Most Significant Bit/Byte	
MTA	Message Transfer Agent (X.400)	ITU
MTU	Maximum Transfer Unit (TCP/IP)	IETF
MUX	Multiplexer	
NAK	Negative Acknowledgement	
NAS	Network Application Architecture	DEC
NAU	Network Adressable Unit (SNA)	IBM
NBP	Name Binding Protocol (AppleTalk)	Apple
NCC	Network Control Centre	
NBP	NetBios Protocol (3Com 3+Open)	3Com
NCP	Network Control Point (SNA)	IBM
NCP	Netware Core Protocol	Novell
NDIS	Network Driver Interface Specification	Microsoft
NETBIOS	NETwork Basic Input Output System	IBM
NFS	Netwok File System (TCP/IP)	Sun
NIC	Network Interface Card/Controller	
NIS	Network Information Services (Yellow Pages, NFS)	Sun
NLSP	Netware Link State Protocol	Novell
NNI	Network-Network Interface	ITU
NPSI	NCP Packet Switching Interface (SNA)	IBM
NRM	Normal Response Mode (HDLC)	ITU
NRZ	Not-Return To Zero codering	ITU
NRZI	NRZ Inverted codering	ITU
NSAP	Network Service Access Point	ISO
NVT	Network Virtual Terminal	Dec

OAM	Operations, Administration and Maintenance (ATM)	ITU/ISO
ODI	Open Datalink Interface (Novell)	Novell
OEM	Original Equipment Manufacturer	
OLTP	On-Line Transaction Processing	
OSI	Open Systems Interconnection (ISO)	ISO
OSPF	Open Shortest Path First (TCP/IP)	IETF
PABX	Priviate Automatid Branch eXchange	
PAD	Padding field (IEEE)	IEEE
PAD	Packet Assembler Disassembler (CCITT)	ITU
PAP	Printer Access Protocol (AppleTalk)	Apple
PARC	Palo Alto Research Centre (Xerox)	
PBX	Priviate Branch eXchange	
PCI	Peripheral Component Interconnect bus	Industrie
PCI	Protocol Control Information (OSI)	ISO
PCM	Pulse Code Modulation, 64 kb/s	ITU
PDH	Plesiochronous Digital Hierarchy	ITU
PDS	Premises Distribution System, Systimax (AT&T)	AT&T
PDU	Protocol Data Unit (OSI)	ISO
PHY	Physical Layer (OSI)	ISO
PKC	Public Key Criptology	ISO
PM	Physical Medium (ATM)	ITU
PMD	Physical Media Dependent (FDDI)	ANSI
PNCP	Peripheral Node Control Point (IBM)	IBM
POH	Path Overhead (SDH)	ITU
POS	Point Of Sale	
PPP	Point to Point Protocol (TCP/IP)	IETF
PRA	Primary Rate Access (ISDN, 30B+D)	ITU
PRM	Protocol Reference Model (ITU)	ITU
PS	PostScript (Appletalk)	Apple
PSDN	Packet Switched Data Network	ITU
PSK	Phase Shift Keying	
PSTN	Public Switched Telephone Network	ITU
PU	Physical Unit (SNA)	IBM
PVC	Permanent Virtual Circuit (X.25)	ITU
QLLC	Qualified Logical Link Control (SNA oevr X.25)	IBM
QOS	Quality of Service (OSI)	ISO

RARP	Reverse ARP (TCP/IP)	IETF
RBOC	Regional Bell Operating Compagny	
RFC	Request For Comment (TCP/IP)	IETF
RFS	Remote File Sharing (TCP/IP)	IETF
RIP	Routing Information Protocol (Netware, TCP e.a.)	
RJE	Remote Job Entry	
RNR	Receive Not Ready (o.a. HDLC)	ITU
ROSE	Remote Operations Service Element (OSI)	ISO
RPC	Remote Procedure Call (TCP)	Sun
RR	Receive Ready (V.24, BSC, HDLC)	ITU
RS-232C	V.24/V.28 interface	EIA
RS-449	V.35/V.36 interface	EIA
RSA	Rivest, Shamir, Adleman (encription)	
RTMP	Routing Table Maintenance Protocol (Appletalk)	Apple
RTS	Request To Send (V.24)	ITU
SA	Source Adress	IEEE
SAA	Systems Application Architecture	IBM
SAP	Sevice Access Point (OSI)	ISO
SAP	Server Advertising Protocol (Netware)	Novell
SAPI	Service Access Point Identifier (ISDN)	ITU
SAR	Segmentation and Reassembly (ATM)	ITU
SCCP	Signalling Connection Control Part (B-ISDN)	ITU
SD	Start Delimiter (OSI)	ISO
SDH	Synchronous Digital Hierarchy	ITU
SDLC	Synchronous Data Link Control (SNA)	IBM
SDU	Service Data Unit	ITU
SEAL	Simple and Efficient Adaption Layer (AAL 5)	ITU
SIP	SMDS Interface Protocol	
SIR	Sustained Information Rate (FR, ATM e.a)	
SLIP	Serial Line Internet Protocol (TCP/IP)	IETF
SMB	Service Message Block Microsoft/IBM	
SMDS	Switched Multimegabit Data Service	
SMT	Station Management (FDDI)	ANSI
SMTP	Simple Mail Transfer Protocol (TCP/IP)	IETF
SN	Sequence Number (ATM)	ITU
SNA	Systems Network Architecture	IBM
SNAP	SubNetwork Access Protocol (IEEE 802.2)	IEEE
SNI	Subscriber Network Interface (SMDS)	

SNMP	Simple Network Management Protocol (TCP/IP)	IETF
SPX	Sequenced Packet eXchange (Netware)	Novell
SOH	Section Overhead (SDH)	ITU
SONET	Synchronous Optical NETwork, 78Mb/s	Bellcore
SQL	Structured Query Language	
SR	Source Routing (TRN)	IBM
SREJ	Selective Reject (HDLC)	ITU
SRPI	Server Requestor Programming Interface (SNA)	IBM
SRT	Source Routing Transparent	IBM/IEEE
SSAP	Source Service Access Point (IEEE 802.2)	IEEE
SSCP	System Services Control Point (SNA)	IBM
SSCP	Service Specific Convergence Protocol (ATM)	ITU
SSM	Single Segment Message (ATM)	ITU
STM	Synchronous Transport Mode/Module (SDH)	ITU
STP	Shielded Twisted Pair cabling (TRN)	IBM
STP	Spanning Tree Protocol (IEEE 802.1D)	IEEE
STX	Start of Text (BSC)	IBM
SVC	Switched Virtual Circuit (X.25)	ITU
SVC	Signalling Virtual Channel (ATM)	ITU
SVID	System V Interface Definition (Unix)	OTF
SYN	Synchronize (BSC)	IBM
TCAM	TeleCommunication Access Methode (IBM S/370)	IBM
TCAP	Transaction Capabilities Application Part (Q.2931)	ITU
TCP	Termination Control Point	ITU
TCP	Transmission Control Protocol (TCP/IP)	IETF
TDM	Time Division Multiplexing	ITU
TE	Terminal Equipment (ISDN)	ITU
TEI	Terminal Endpoint Identifier (ISDN)	ITU
TIC	Tokenring Interface Card (TRN)	IBM
TLAP	TokenTalk Link Access Protocol (Appletalk)	Apple
TLI	Transport Level Interface (Unix)	IETF
TMN	Telecommunications Management Network	ITU
TNM	Transmission Network Management	ITU
TOP	Technical Office Protocol (Boeing)	ISO
TP4	Transport Protocol 4 (OSI)	ISO
TST	Transparent Spanning Tree (IEEE)	IEEE
TTL	Time To Live (TCP/IP)	IETF

U(S)ART	Universal (Synchronous and) Asynchrounous Reciever Transmitter	
UA	User Agent (X.400)	ITU
UDP	User Datagram Protocol (TCP/IP)	IETF
UE	User Entity(OSI)	ISO
UNI	User-Network Interface (Public networks)	ITU
URL	Uniform Resource Locators (WWW)	IETF
USART	Universal Synchrounous Asynchronous Reciever Transmitter	
UTP	Unshielded Twisted Pair cabling	
UUCP	Unix to Unix Copy Program (Unix)	
V.24	DTE-DCE interface definition PSTN	ITU
VAN	Value Added Network	
VBR	Variable Bit Rate	ITU
VC	Virtual Container (SDH)	ITU
VC	Virtual Call (X.25)	ITU
VC	Virtual Channel (ATM)	ITU
VF	Voice Frequency (300-3000 Hz)	ITU
VOD	Video On Demand (B-ISDN)	
VP	Virtual Path (ATM)	ITU
VRC	Vertical Redundancy Check - pariteitsbit	
VSAT	Very Small Aperture Terminal, 1 m satellite dish	
VT	Virtual Terminal (OSI)	ISO
WACK	Wait Acknowledgement	IBM
WAN	Wide Area Network	
WWW	World Wide Web	IETF
X.121	adressing X.25	ITU
X.21	Signalling Replaces V.24 and V.25 standards	ITU
X.25	Packet switched network interface standard (CCITT)	ITU
X.3/28/29	Triple X PAD standard (X.25)	ITU
X.400	Message Handling (Email) standard (MHS)	ITU
X.500	Directory Services standard (DNS)	ITU
X.75	Packet switching inter-network interface (X.25)	ITU
XDR	eXternal Data Repesentation (NFS)	Sun
XNS	Xerox Network System	Xerox
YP	Yellow Pages (NFS)	Sun
ZIP	Zone Information Protocol, AppleTalk phase2 routing information protocol Apple	

Verklarende woordenlijst

100BaseT	IEEE standaard voor 100 Mbps CSMA/CD over twisted pair kabels
10Base2	IEEE standaard voor CSMA/CD over dunne coaxkabels
10Base5	IEEE standaard voor CSMA/CD over dikke coaxkabels
10BaseF	IEEE standaard voor CSMA/CD over glasvezel kabels
10BaseT	IEEE standaard voor CSMA/CD over twisted pair kabels
3174	IBM cluster controler (terminal server) waarmee 8 tot 32 terminals aan een SDLC, X.25, Token Ring of Ethernet worden gekoppeld.
3270	Algemene aanduiding voor een familie IBM terminals.
3745	IBM Front End Processor (zie FEP).
802.2	IEEE standaard voor LLC: zie LLC-1 en LLC-2.
802.3	IEEE standaard voor CSMA/CD toegangsmethode
802.5	IEEE standaard voor Token Passing Ring toegangsmethode
ANSI	American National Standards Institute
API	Application Programming Interface tussen communicatieprotocollen en applicatie of laag 7 protocol
ARPA	Advanced Research Projects Agency, onderdeel van het Amerikaanse ministerie van defensie (Department of defence, DoD). Dit bureau gaf de aanzet voor ARPAnet, de voorganger van The Internet en de proeftuin voor TCP/IP.
ASCII	American Standard Code for Information Interchange, codetabel die aan elke letter, cijfer en leesteken een 8-bits waarde toekent.
ASN.1	Abstract Syntax Notation #1 (OSI). 'Taal' voor het beschrijven van datastructuren, bv. de elementen in een SNMP MIB.
ATM	Asynchronous Transfer Mode (CCITT). Nieuwe standaard voor zeer hoge snelheidsnetwerken in principe voor WAN's.
AUI	Attachment Unit Interface, IEEE standaard voor de 15-polige D-connector tussen Ethernet station en transceiver. Zie ook DIX.
BOOTP	BOOTstrap Protocol (TCP/IP). Eenvoudig protocol waarmee een disk-less station aan een server in het netwerk zijn adres, de naam van zijn bootstrapfile en andere parameters kan vragen.
bps	Bits Per Second. Maat voor de transportsnelheid van een verbinding.
BSD	Berkeley System Distribution (Unix).
CADAM	Computer Aided Design And Manufacturing, programma voor mainframes, met eigen SDLC-achtig communicatie protocol.
CAU	Controlled Access Unit (Token Ring). Actief aansluitapparaat voor Token Ring stations.
CCITT	Comité Consultatif Internationale de Telegraphique et Telephonique, nu opgevolgd door ITU-TSS

CDDI	Copper Distributed Data Interface (FDDI over Copper). Standaard voor het voeren van FDDI signalen over STP kabels.
CRC	Cyclic Redundancy Check. Familie van checksum formules: CRC -12, CRC-16, CRC-32. Extra controlegetallen.
CSMA/CD	Carrier Sense Multiple Access/Collission Detect. Toegangsmethode van Ethernet-achtige netwerken
DIX	Digital, Inter, Xerox: het consortium dat Ethernet standaardiseerde. Ook: DIX connector: de 15-polige D-connector voor het aansluiten van een transceiver op een Ethernet station.
DLSw	Data Link Switching, IBM benaming voor het tunnelen van SDLC via TCP/IP.
EBCDIC	Extended Binary Coded Decimal Interexchange Code. Een codetabel die aan elke letter, cijfer en leesteken een 6-bits waarde toekent. Gebruikt door IBM systemen. Zie ook ASCII.
FCS	Frame Check Sequence. Meestal het laatste veld van een frame of pakket, bevat een checksum om te kunnen controleren of de inhoud nog correct is. Zie CRC.
FDDI	Fiber Distributed Data Interface (ANSI). Standaard voor 100 Mbps token ring LAN, oorspronkelijk alleen over glasvezel kabels.
FEP	Front End Processor (IBM), hulpcomputer voor IBM mainframes die de communicatie verzorgt.
FOIRL	Fiber Optic Inter Repeater Link, oude IEEE standaard voor glasvezel koppeling tussen twee repeaters.
FR	Frame Relay. WAN protocol, sterk vereenvoudigd, afgeleide van X.25.
FTP	File Transfer Protocol (TCP/IP). Standaard applikatie voor overdracht van bestanden met TCP/IP.
FTP	Foiled Twisted Pair: door middel van aluminium folie afgeschermde kabel.
FYI	For Your Information. Afkomstig uit de TCP/IP wereld. Documenten met allerlei achtergrond informatie.
HUB	As, spil: centraal opgestelde kast waarin verschillende LAN en netwerkfunctie kunnen worden ondergebracht (repeaters, MAUs, CAUs, bridges, routers, etc.)
ICMP	Internet Control Message Protocol (TCP/IP). Protocol voor 'dienstberichten' in een TCP/IP netwerk.
IGRP	Integrated Gateway Routing Protocol. Cisco's fabrieks-eigen routing informatie protocol voor IP.
IP	Internet Protocol (TCP/IP). Netwerk laag van TCP/IP.
ISDN	Integrated Services Digital Network (CCITT). Standaard voor volledig digitaal telefoonnet.
ISO	International Organization for Standardization.
ITU	International Telecommunications Union (UN)

ITU-TSS	ITU Telecommunications Standards Section (UN). Opvolger van CCITT.
kbps	Kilo-bits per second. Duizend bits per seconde.
LAM	Lobe Attachment Module. Onderdeel van CAU waarin zich de poorten voor de stations bevinden.
LAN	Local Area Network
LAT	Local Area Transport (DEC). Protocol voor communicatie tussen terminalservers en DEC computers. Niet routeerbaar, wijkt af van OSI/RM.
LLC	Logical Link Control. IEEE 802.2 protocollen voor bovenste helft DLC.
LLC-1	Logical Link Control type 1, een connection-less protocol voor LAN omgeving, afgeleid van HDLC.
LLC-2	Logical Link Control type 2, een connection-oriented protocol voor LAN omgeving, afgeleid van HDLC.
Lobe	Uitloper van een Token Ring netwerk naar een station; de kabel tussen MAU of CAU en het station.
MAC	Media Access Control. IEEE 802.x protocollen voor onderste helft DLC.
MAU	Multistation Access Unit (TokenRing). Passief apparaat voor het aansluiten van Token Ring stations.
MAU	Medium Attachment Unit. Bij Ethernet en dikke coaxkabel wordt hier de transceiver op aangesloten.
Mbps	Mega bits per second. Miljoen bits per seconde.
MF	MainFrame, algemene aanduiding voor een 'grote' computer.
MIB	Management Information Base (SNMP). Definitie (beschrijving) van gegevens (tellers statusvlaggen, etc.) die m.b.v. SNMP kunnen worden uitgelezen en/of ingesteld.
NCP	Network Control Program (SNA). Programma's die op FEP draaien.
NCP	Netware Core Protocol. Protocol van Novell voor communicatie tussen PC en server, wordt geimplementeerd door NETx.
NFS	Netwok File System (TCP/IP/Sun). Protocollen die in TCP/IP omgeving file- en print-sharing mogelijk maken.
OSI	Open Systems Interconnection (ISO), overkoepelend begrip voor alle ISO communicatie standaarden.
OSI/RM	OSI Reference Model. Het bekende 7-lagen model.
OSPF	Open Shortest Path First (TCP/IP). Link-state routing information protocol voor TCP/IP en andere protocollen.
PARC	Palo Alto Research Centre (Xerox). Onderzoekscentrum van Xerox in Californië waar windows, muizen, iconen en Ethernet zijn uitgevonden.

peer-peer	Een relatie tussen gelijken: bv. als IP in de ene computer 'praat' met IP in een andere computer, is dat communicatie op gelijk niveau, in tegenstelling tot de master-slave relatie in sommige andere protocollen.
Ping	Programma om in een TCP/IP netwerk pakketjes naar een andere computer te sturen om te controleren of de verbinding werkt.
PIR	Protocol Independent Routing, CrossComm's fabrieks-eigen protocol voor LAN-bridging en WAN-routing.
PVC	Permanent Virtual Circuit (X.25). Vaste logische verbinding tussen twee aansluitpunten in een X.25 of Frame Relay netwerk.
QLLC	Qualified Logical Link Control, protocol om SNA over X.25 te transporteren. Afgeleid van SDLC, maar voorkomt 'polling' over het X.25 netwerk omdat daar meestal per frame moet worden betaald.
RFC	Request For Comment (TCP/IP). Voorstel of beschrijving van een TCP/IP standaard of daarmee samenhangende zaak.
RIP	Routing Information Protocol. Algemeen: een protocol voor uitwisseling van routerings informatie tussen routers in een netwerk. Specifiek: een bepaald vector-distance protocol, gebruikt voor IP, XNS en andere.
SDLC	Synchronous Data Link Control (SNA). DLC protocol voor synchrone verbindingen in SNA.
SDLLC	Cisco benaming voor SDLC naar LLC conversie (en omgekeerd).
SMTP	Simple Mail Transfer Protocol (TCP/IP). Applicatieprotocol voor het uitwisselen van electronic mail.
SNA	Systems Network Architecture (IBM). Overkoepelende naam voor IBM netwerk protocollen en produkten.
SNMP	Simple Network Management Protocol (TCP/IP). Protocol voor het uitwisselen van netwerk management informatie tussen een netwerk management station en de te beheren netwerk apparatuur.
SRB	Source Route Bridging. Protocol dat de weg door een stelsel van gekoppelde Token Ring netwerken bepaalt.
STA	Spanning Tree Algorithme. Methode die ten grondslag ligt aan het STP protocol voor de uitwisseling van informatie tussen transparante bridges zodat zij de structuur van het netwerk tot een 'boom' kunnen reduceren door extra verbindingen tijdelijk buiten gebruik te stellen.
Stack	Stapel. Twee of meer protocollen die opeenvolgende lagen van het OSI/RM invullen en dus op elkaar 'gestapeld' kunnen worden. Ook wel: een programma dat enkele protocollen implementeert, bv. een "TCP/IP stack".
STP	Shielded Twisted Pair cabling. Kabeltype dat bestaat uit getwijnde paren die van een extra geleidende afscherming zijn voorzien, bv. IBM Cabling System's Type 1.

STP	Spanning Tree Protocol (IEEE 802.1D). Protocol voor de uitwisseling van informatie tussen transparante bridges zodat zij de structuur van het netwerk tot een 'boom' kunnen reduceren door extra verbindingen tijdelijk buiten gebruik te stellen.
STUN	Cisco benaming voor Serial TUNneling van SDLC en HDLC verkeer via TCP/IP.
SVC	Switched Virtual Circuit (X.25). Tijdelijke logische verbinding tussen twee aansluitpunten in een X.25 of Frame Relay netwerk, wordt dooreen van beide aangeslotenen opgezet en na het einde van de sessie weer afgebroken.
TCP	Transmission Control Protocol (TCP/IP). Transportlaag protocol .
TDM	Time Division Multiplexing. Een methode om een aantal verschillende signalen over één verbinding te sturen door ze in een vaste volgorde af te wisselen. Alle datacommunicatie technieken zijn vormen van TDM, specifiek Statistical TDM
Telnet	Virtual terminal protocol voor TCP/IP.
TIC	Token ring Interface Card (IBM). Netwerk kaart voor een Token Ring netwerk, bv. voor een FEP, cluster controler of PC.
TRN	Token Ring Network (IBM). LAN volgens IBM specificaties, ook gestandaardiseerd in IEEE 802.5.
Trunk	(letterlijk: stam). Hoofdkabel, bv. de verbinding tussen de MAUs en/of CAUs in een Token Ring netwerk.
UDP	User Datagram Protocol (TCP/IP). Eenvoudig transportlaag protocol in TCP/IP.
UN	United Nations, de Verenigde Naties.
UTP	Unshielded Twisted Pair cabling. Bekabeling op basis van getwijnde paren zonder extra (electrische) afscherming.
VTAM	Virtual Terminal Acces Methode. SNA communicatie software die op IBM mainframes draait.
WAN	Wide Area Network. netwerk dat zich over groet afstanden uitstrekt en (in het algemeen) gebruik maakt van de kabels en apparatuur van een telecom aanbieder.
WC	Wiring Closet, ruimte waarin bekabeling bij elkaar komt en patchpanelen zijn te vinden. Doorgaans ook de ruimte waar zich repeaters, MAUs, CAUs en HUBs bevinden.
X.25	Packet switched network interface standard (CCITT). Standaard voor een pakket-geschakeld netwerk zoals Datanet/1 van PTT Telecom.
XNS	Xerox Network System (Xerox). Protocollen ontwikkeld door Xerox, stond model voor veel nieuwere protocollen, zoals TCP/IP, DECnet en IPX/SPX

Literatuur

IBM Multisegment LAN Design Guidelines, gedetaileerde informatie over het ontwerpen en implementeren van grote Token Ring netwerken. IBM publicatie GG24-3398-01.

Internetworking Technology Overview, Cisco Systems. Onderdeel van de Cisco dokumentatie set. Bevat een korte uitleg van alle protocollen, interface standaarden, etc.

Local Area Networks - Datacommunicatie #6, Verdonk en Hermelink. Uitgave Samson/PTT Telecom. ISBN 90-14-04445-3. Deel uit een serie leerboeken.

SNA, een inleiding, Peelen. Uitgace Kluwer. ISBN 90-201-2588-5.

Datacommunicatie, met local area networks, Den Heijer en Tolsma, uitgave Kluwer. ISBN 90-201-2107-3.

Internetworking with TCPIP, Vol 1, Comer. Prentice-Hall. ISBN 0-13-474321-0.

Guide to Loacl Area Networking, verkrijgbaar bij Cabletron Systems Benelux (Woerden).

Computer Netwerken, 2e druk A. Tanenbaum. Prentice Hall, ISBN 90-6233-497-0.

The Simple Book, M. Rose. Prentice Hall, ISBN 0-13-177254-6.

Handboek *Netwerk Management*. Kluwer Bedrijfswetenschappen.

TCP/IP Illustrated, Volume 1, The protocols, W. Richard Stevens, Addison Wesley, ISBN 0-201-63346-9, 1994

Internet System Handbook, Daniel C. Lynch en Marshall T. Rose, Addison WesleyISBN0-201-56741-5, 1993

Interconnections, Radia Perlman, Addison Wesley, ISBN 0-201-56332-0, 1992

Telecommunications: Protocols and Design, John D. Spragins, Addison Wesley, ISBN 0-201-09290-5, 1991

NOTITIES

NOTITIES

NOTITIES

NOTITIES

NOTITIES