

COMPUTERNETWERKEN 1

Axel Hamelrijck

EXAMEN JANUARI 2019 UCLL TOEGEPASTE INFORMATICA

Beste medestudenten,

Jullie mogen meegenieten van deze samenvatting die ik heb gemaakt, maar ik neem geen verantwoordelijkheden op voor slechte punten of voor onvolledigheid.

Vergeet natuurlijk jullie oefeningen niet te hermaken op Edublend en de oefenexamens te bekijken. Probeer de oefeningen van de examens (zie Toledo) ook eens te maken op papier, zo wordt het op het examen gevraagd.

Ik leg de oefeningen uit op de manier hoe ik het begrijp.

Ik wens jullie verder nog veel succes met de examens.

Donaties zijn altijd welkom voor mijn harde werk:

<https://paypal.me/axelele>

1.0 Hoofdstuk 1

1.1 Protocollen

Communiceren computerprogramma's → via afspraken

- Cliëntprogramma → initiatief nemend programma voor communicatie
- Serverprogramma → wacht voor een ander programma dat initiatief neemt

Afspraken die worden

Communicatie tussen computerprogramma's → vastgelegde afspraken →

- Welke berichten er kunnen verstuurd worden
- Wat elk bericht betekent
- Hoe op elk bericht kan gereageerd worden

→ protocollen

Principe van *verdeel en heers* wordt toegepast → afspraken voor elk welbepaald deel van computers; per aspect 1 protocol →

- Afspraken hoe een bit wordt voorgesteld
- Afspraken i.v.m. foutcontrole
- ...

Per aspect → set van afspraken → **protocol**

Verschillende aspecten worden vastgelegd → verschillende protocollen die samen → **protocolstack** vormen

Protocol op WWW → HTTP → HyperText Transfer Protocol

1.2 Protocollagen

Netwerk kan worden gezien als → infrastructuur die gedistribueerde applicaties voorziet van diensten

Verschillende browsers, verschillende soorten netwerken, ... → altijd zelfde dienst → **LAGEN**

- Applicatielaag
 - Communicatieboodschappen uitgewisseld tussen toepassingen die gebruik maken van het netwerk → kunnen gevolg zijn van interactie tussen applicatie en gebruiker
 - Presentatielaag
 - Sessiel laag
 - Transportlaag
 - Netwerklaag
 - Data-linklaag
 - Fysieke laag
- ← tussenlagen

Verschil protocol en programma →

- Protocol → set van afspraken
 - Bv. HTTP
- Programma → één van de mogelijke implementaties van de afspraken
 - Bv. Firefox, Apache → implementeren HTTP

Abstractielagen → lagen van andere niveaus (bv. toepassingslaag) hoeft geen kennis te hebben van het type netwerk of type communicatie medium (→ andere lagen). → elke laag is verschillend; complexiteit wordt verdeeld en heeft een eigen opdracht, om zo samen te werken om een bericht te versturen / ontvangen

Voordelen →

- Complexiteit kan verdeeld worden.
- Meerdere toepassingen kunnen gebruik maken van dezelfde programmatuur van de onderliggende laag.

Communicatie is virtueel = **berichten worden afgeleverd aan en ontvangen van een tussenlaag**. De bovenste van de tussenlagen.

Elke laag **bewerkt het bericht en levert het af** aan de laag eronder

Een stuk van een bericht, opgedeeld door de tussenlagen = **segment**

Redenen om grotere berichten op te delen in segmenten →

- **De kans dat een segment een fout bevat is kleiner en wanneer dit het geval is moet enkel dit segment opnieuw verstuurd worden.**
- **Meerdere kunnen het netwerk tegelijd gebruiken** = netwerk kan segmenten van verschillende communicaties door elkaar versturen.

1.3 TCP/IP

Lagen tussen toepassingslaag en fysieke laag → bijna altijd TCP/IP (Transport Control Protocol / Internet Protocol) gebruikt

Vb. Van TCP/IP afspraken

- Elke computer → aangeduid met uniek IP adres van 4x8 bits (4x getal tussen 0 en 255) → bv. 198.138.27.104
- Proces/toepassing/applicatie → aangeduid met **poortnummer**
 - vb. WWW → poort 80
 - Meerdere cliëntprocessen kunnen actief zijn op 1 computer
 - Willekeurig vrij nummer gekozen groter dan 1024

1.3.1 TCP/IP Toepassingen

Voor elke toepassing → apart protocol → server- en cliëntprogramma's moeten juiste implementatie van afspraken hebben

Vb. toepassingen

- WWW
- E-mail
- Bestandsoverdracht
- Terminaal-emulatie

Toepassingen hebben volgende eigenschappen gemeen :

- **Client- en serverprogramma's.**
- **Gecommuniceerd via TCP/IP.**
- **Berichten die uitgewisseld worden, zijn vastgelegd in een protocol.**

Client die dienst wil → IP van computer waar server op draait en poortnummer voor applicatie kennen

Protocollen → moeten publiekelijk en gestandaardiseerd zijn voor alle bedrijven → standaarden worden opgezet door IETF

1.4 OSI-model van tussenlagen

- TCP = protocol voor **transportlaag** = **TCP-segmenten**.
- IP = protocol voor **netwerklaag** = **IP-pakketten**.
- Besturingsprogramma voor de netwerkkaart = **datalink-laag** = **Ethernet-frames**.
- Frames van ene naar andere machine te sturen = **fysische laag** = **spanningsniveau's** (nullen en enen).
- Vroeger had de **International Standards Organisation** de verschillende aspecten van datacommunicatie opgedeeld in 7 lagen = **OSI-referentiemodel (Open Systems Interconnection)** =
 - **Toepassingslaag** = besturingssysteem met netwerktoepassingen
 - **Presentatielaag** = heeft betrekking op afspraken voor het coderen en decoderen van gegevens voor de toepassingslaag
 - **Sessiel laag** = heeft betrekking op de communicatie tussen twee toepassingsprocessen
 - **Transportlaag** = heeft betrekking op afspraken om een pakket zonder fouten van bron naar eindbestemming te sturen, maar op een hoger niveau dan de datalink-laag (omdat er meerdere tussenstations kunnen zijn en omdat pakketten kunnen opgesplitst worden)
 - **Netwerklaag** = heeft betrekking op het transport, de adressering en de routing van pakketten doorheen het netwerk alsook op het opzetten van een route van bron naar eindbestemming, nadat beide adressen fysiek bepaald zijn als fysische adressen.
 - **Datalink-laag** = heeft betrekking op de datatransmissie en afspraken om fouten te corrigeren, of te melden. Ook de afspraken over de wijze waarop bits in pakketjes gebundeld en weer uitgepakt worden, afspraken over de wijze waarop de handshaking gebeurt
 - **Fysische laag** = heeft betrekking op alles wat nodig is om de data fysiek over een netwerk te transporteren, inclusief de bekabelingsmethodes, maar niet de bekabeling zelf.
- Dit model geeft niet aan welke afspraken er gemaakt moeten worden, **wel hoe men de afspraken moet indelen**.
- Afspraken voor de ene laag **mogen de afspraken voor een andere laag niet beïnvloeden**.
- Met uitzondering van de fysische laag worden de lagen via **programmatuur geïmplementeerd**.
- Programmatuur in een bepaalde laag **ziet de programmatuur in de onderliggende laag als subroutines**.

1.5 OSI-referentiemodel van tussenlagen

Maar 4 lagen →

- **Toepassingslaag** = bovenste 3 OSI
- **Transportlaag** = zelfde als OSI
- **Internetlaag** = zelfde als netwerklaag OSI
- **Datalinklaag** = zelfde als OSI
- **Fysieke laag** = zelfde als OSI

TCP/IP model	Protocols and services	OSI model
Application	HTTP, FTP, Telnet, NTP, DHCP, PING	Application
		Presentation
		Session
Transport	TCP, UDP	Transport
Network	IP, ARP, ICMP, IGMP	Network
Network Interface	Ethernet	Data Link
		Physical

1.6 OSI vs TCP/IP

OSI →

- theoretisch model
- 7 lagen

TCP/IP →

- bestaat echt en wordt gebruik
- 5 lagen (4 als onderste 2 worden samengevoegd)

Hoofdstuk 2: Datalinklaag, netwerken en LANs

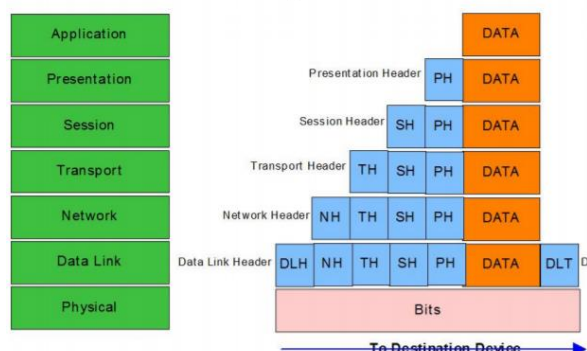
Fysieke laag → afzonderlijk bits (of in groep) verstuurd

Datalink laag → frames verstuurd

2.1 Terminologie: link, node, PDU en netwerk

- **Node** →
 - Machine die informatie uitwisselt over computernetwerk
- **Link** →
 - Communicatiekanaal tussen aangrenzende nodes
 - Broadcast
 - Meerdere nodes verbonden door gedeelde broadcastlink → verzenden van frames is voor iedereen
 - Point-to-pointlink
 - Rechtstreeks tussen twee nodes → frame naar 1 andere node
- **Netwerk** →
 - Twee of meerdere nodes die met elkaar verbonden zijn door middel van een gemeenschappelijk link en die rechtstreeks met elkaar communiceren
- **PDU** →
 - Data gespecificeerd in protocol die wordt verstuurd tussen twee nodes →
 - **PDU**

Encapsulation



2.2 Diensten van Datalinklaag

Primaire functie in datalinklaag → ervoor zorgen dat de data over een gemeenschappelijke link uitgewisseld wordt.

Datalink voorziet 3 basisdiensten →

- Interface met de netwerklaag voorzien en frames vormen
- Frames versturen d.m.v. media access control → coördineren van verzending / ontvangst van frames door verschillende nodes over een gemeenschappelijke link
- Frames ontvangen en foutdetectie → geen nut om frames / PDU's te ontvangen / versturen die fouten bevatten

Frame bestaat uit 3 elementen →

- Frameheader
- Framepayload → met hierin pakket van hogere netwerklaag
- Frametrailer / framefooter

2.3 Implementatie van datalinklaag

Datalinklaag →

- In hardware
 - op netwerkkkaart (NIC)
 - op moederbord (NIC geïntegreerd op moederbord)
- In software
 - Softwarecomponent van datalinklaag → driver van NIC

2.4 Multiple-Accesslinks en -protocollen

Multiple-access protocollen voor toegang →

- Kanaalpartitioneringsprotocollen
 - Niet verder kennen
- Willekeurige toegangsprotocollen
 - Computer wil zenden, wacht tot link (/drager) vrij is en begint te zenden
 - Botsingen → gebeuren wanneer twee of meerdere nodes tegelijk een frame verzenden
 - Na botsing → toestand hersteld
 - Wordt ook **toegang door contentie** genoemd
 - Vb → ethernet
- Deterministische toegangsprotocollen
 - Computer wacht tot hij toegang krijgt
 - Toegang → zekerheid dat andere computers niet zullen storen
 - Vaak gebruik van een **token**
 - Token → bewijs van toegangsrecht, is maar één van
 - Wie token heeft, mag zenden
 - Token wordt doorgegeven

Types LAN: Ethernet, token bus, token ring,...

Ethernet → meest gebruikt

2.5 Ethernet: Fysiek

Ethernetkaart → uniek adres

- Bestaat uit 48 bits → 12 hexadecimale cijfers
- → MAC-adres → medium access control

Bekabeling → UTP (Unshielded Twisted Pair), STP (Shielded TP), FTP (Foiled TP),...

Via een TP kabel → elke pc verbonden met Switch → star-topology

RJ45 → aansluiting voor kabels

Omwisseling →

- Zenden → via draden 1 en 2
- Onvtangen → via kabels 3 en 6
- Gebeurt in Switch → straight-through cable
 - 2 pc's met elkaar laten communiceren → crossover cable
- 2 switches met elkaar verbinden (door te weinig poorten)
 - Verbinden met crossover cable OF
 - Sommige switches hebben 1 dubbele poort → in 1 van de twee gebeurt omwisseling (met rechte kabel dan)
 - Beide aansluitingen kunnen niet tegelijk worden gebruikt

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

Foute kabel bij foute aansluiting gebruiken → netwerkfout

Toekomst → aansluitingen die zich automatisch aanpassen aan kabel

2.6 Ethernet: frames

Ethernet-bitrij bevat

- Adres van bestemming (MAC adres van 6 bytes)
- Adres van zender (MAC adres van 6 bytes)
- Lengte of type (2 bytes)
- Data (max 1500 bytes, ook **Maximum Transmission Unit (MTU)** genoemd)
- (eventueel) opvulbytes (lengte moet minstens 64 bytes bedragen)
- Controle informatie (4 bytes)

Totale max lengte → 1518 bytes

Totale min lengte → 64 bytes

Om een ethernetkaart te gebruiken moet de drive geïnstalleerd zijn → deze driver verstuurt en ontvangt frames

Verschil fysieke laag en datalinklaag → fysieke laag staan bits

→ datalinklaag groepeer bits tot frames, versturen en ontvangen

Preamble → aan een ethernet frame hangt een preamble van 7 bytes en een SFD (start of frame delimiter → 1 byte). Deze twee vormen een patroon en geven aan dat er een frame volgt

2.7 Ethernet a.k.a. CSMA/CD

Versturen van frames → drivers moeten zich aan afspraken houden → het **CSMA/CD** protocol

CSMA/CD → Carrier Sense Multiple Access / Collision Detection

Vroeger → Elke pc verbonden met 1 coax kabel → bus-(broadcast)netwerk

Elk bit die op de kabel gingen passeerde elke PC

Op uiteinden van de kabel was er een terminatie (kabelafsluitingen → voorkomen dat signalen weerkaatsen)

Later → coax kabel vervangen door hub (alle pc's verbonden aan hub)

Hub → verstuurt elk signaal dat binnenkwam naar alle andere pc's buiten naar de pc van wie hij het signaal kreeg → elk bit dat wordt gestuurd passeert ook elke pc

Ethernet principe → frame passeert langs alle pc's → alleen pc voorwie frame bestemd is mag frame lezen: elke kaart leest het bestemmingsadres van frame →

- Is dit mijn adres?
- Is dit een broadcast-adres? → 12 F's
- Is dit multicast-adres voor mijn groep → afgesproken adres voor groep kaarten

Ja → frame wordt gelezen

Afspraken (protocol) voor Ethernetkaarten → CSMA/CD

CS → **Carrier Sensing** (carrier = (gegevens)drager) → Als er twee frames tegelijk op de kabel worden geplaatst, is er een botsing

Om een botsing (**collision**) te voorkomen moet een kaart eerst nagaan (to **sense**) of de kabel (**carrier**) vrij is

Niet vrij → wachten tot de kabel vrij is

MA → **Multiple Access** → meerdere knooppunten hebben tegelijkertijd toegang tot de gegevensdrager of link (→ kabel)

CD → **Collision Detection** → Er kunnen botsingen voorkomen

De kaarten moeten de kabel inspecteren om te **detecteren** of er een botsing is

Botsing → frames verloren

→ de kaart gaat een **JAM**-signaal sturen gedurende een tijd

→ Zo weet elke kaart op het netwerk dat er een botsing was

Na botsing → kaarten die iets wouden sturen wachten een willekeurige tijd voor ze weer opnieuw iets sturen

Nog botsingen mogelijk; bv:

- PC A en B → 1 km van elkaar verwijderd
- Tijd voor eerste bit van frame is bv. 10μs om van A naar B te gaan
- Als B 9,9μs nadat A begonnen is een frame wilt sturen, zal B opmerken dat de bus vrij is
- Na 0,5μs → botsing; nog 0,5 μs later zal B merken dat er een botsing is
- A zal pas 19,9μs na het versturen van eerste bit vaststellen dat er botsing is
- **Als er een botsing is weet de zender dit reeds tijdens het versturen van het frame omdat een frame altijd uit 64 bytes bestaat**
 - 64 bytes sturen neemt genoeg tijd om te weten of er een botsing is
 - Lengte van ethernet kabel speelt een rol

- Botsing frames → frame verloren
- Restant van gebotst frame → **runt**
- Na sturen van 64^{ste} byte → normaal geen botsing meer
 - Wel nog mogelijk, noemt **late collision**
 - → wordt **NIET** opgemerkt door de kaarten → frame niet opnieuw herverstuurd
 - Programma van hoger niveau (op OSI-laag) zal dit moeten oplossen

2.8 Snelheid

Snelheid → bitsnelheid → **aantal bits dat per seconde kan verstuurd worden**

→ wordt ook wel (**digitale**) **bandbreedte** genoemd

Niets te maken met snelheid waarmee bits reizen (dat is 2/3 van lichtsnelheid)

Bandbreedte → snelheid waarmee de bits elkaar mogen opvolgen om zonder fouten onderscheiden te worden bij de ontvanger

Als

- Kabel niet langer is dan 2500m (5km in 5μs)
- Een frame minstens uit 64 bytes bestaat
- Bitsnelheid 10Mbit/sec bedraagt

Dan kan:

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

Ontwikkelregel bij CSMA/CD → Als een kaart een frame op de bus plaatst, gaat de kaart na of het botst. Als de eerste 64 bytes van het frame volledig op de bus kunnen geplaatst worden zonder dat de kaart vaststelt dat er een botsing is, is de kaart zeker dat er geen botsing meer kan zijn

2.9 Switch

CSMA/CD in kort → er kan niet meer dan 1 keer frames worden verstuurd, anders is er een botsing

Kan beter → intelligente hub (→ **switch**) gebruiken die botsingen 'vermindert'

- Switch →
 - Als ethernet frame in switch aankomt, leest de switch het eerste adres van de bestemming. Frame wordt dan verstuurd naar de poort waarop de bestemming zich bevindt
 - Kent Ethernet, kan frame analyseren en adres herkennen
 - Zit in **Datalinklaag**
 - Spreekt men over **geschakeld internet**
- Hub →
 - leest geen adressen en stuurt signalen door op elke poort (behalve op poort waar frame is binnengekomen)
 - elektrisch apparaat dat een bit herkent, regeneert en doorverstuurt
 - Zit in **fysieke laag**
 - Wordt ook **multiport repeater** genoemd
 - Spreekt men over **gedeeld internet**

Switch werkt als volgt →

- In begin gedraagt switch zich zoals hub
- Iedere keer dat frame binnenkomt → slaat de switch het poortnummer en ethernetadres op
 - Switch weet na korte tijd welke ethernetadressen op welke poorten komen
- Frame met bestemmingsadres dat switch niet kent → wordt doorgestuurd naar alle poorten (zoals hub)

Met switches → botsingen uitzonderlijk
→sniffen niet meer mogelijk

Zodat de switch taken vervult →

- Uitgerust zijn met CPU en RAM
 - CPU → ASIC (application specific IC)
 - IC ontworpen voor specifiek doel → ethernetframes checken
 - RAM → C.A.M. (**content addressable memory**) geheugen → niet zoals gewone RAM
 - Is inhoud-adresseerbaar
 - Inhoud → bv. "AB 01 34 00 23 A6 : 1"
 - MAC-adres : switch-poort waarop MAC-adres voorkomt
 - Er kan een andere switch aan een switch hangen → mogelijk dat er meerdere MAC-adressen zijn met zelfde poortnummer
 - Alle items overlopen → traag
 - Toegang geheugen → met inhoud (MAC-adres)
 - Principe van (volledig) associatief cache
 - Gebeurt als : "geef mij de inhoud van een geheugen-item waar in een ander geheugen-item 12 34 45 A9 26 FD staat"

Segment →

- een stuk coax (met op elk uiteinde een terminatie of een repeater)
- OF een hub met TP-kabels waarop alleen computers aangesloten zijn
- OF een TP-kabel die 2 hubs verbindt

Microsegment → één kabel die één pc verbindt met een **switch** wordt microsegment genoemd → ook als segment genoemd

Frame-behandeling → meerdere manieren waarop een switch omgaat met frames

- **Cut-through switching** → zodra de switch de eerste bytes van het frame met bestemmingsadres heeft gelezen zal de switch het frame verder sturen op de poort met de juiste bestemming
- **Store-and-forward switching** → switch leest eerst volledige frame, checkt op fouten (via CRC), geen fouten → frame verstuurd
- **Fragment-free switching** → pas nadat de switch de eerste 64 bytes heeft gelezen zal de switch het frame verder sturen

2.9.1 Virtuele Switch (VLAN)

Goedkope switch → geen mogelijkheid tot configuratie

Duurdere switch → wel mogelijkheid tot configuratie

Afsluiting van poorten om VLAN's (virtual local area network) te configureren

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

Broadcasts → beperkt tot VLAN (bv. management, verkoop)

2.10 Soorten Ethernet

Oude ethernet van 10Mbit/s → verdwenen

Opvolger → **Fast Ethernet**

2.10.1 Fast Ethernet

Eigenschappen

- CSMA/CD als protocol
- 100Mbit/s
- Niet via coax
- Kan zich aanpassen en samenwerken met oude Ethernet
- 10 maal sneller → lengte 10 maal korter (250m)
 - Afstand nog beperkter door eigenschappen TP → TP verbinding (verbinding tussen pc en hub/switch) is max 100m

Fast Ethernet ook bijna weg → Gigabit Ethernet → 1Gbit/s (1000Mbit/s)

2.10.2 Gigabit Ethernet

Max lengte → 25m

Te kort

Oplossing → full duplex Gigabit-switches gebruiken

Full duplex communicatie → elke netwerkkaart kan tegelijk frames versturen en ontvangen

Botsingen komen niet meer voor

1Gbit/s → vooral glasvezel gebruikt

Ook mogelijk via UTP-5e → alle 8 draden gebruikt

2 bits verstuurd → 5 spanningsniveaus (00,01,10,11 + besturingsinformatie)

Gigabit over UTP → limiet

Gebruik van glasvezel (full duplex) → 10 Gbit/s mogelijk, 100Gbit/s in ontwikkeling

2.11 Gestructureerde bekabeling

2.11.1 Fysieke structuur

yeet

2.11.2 Hiërarchische structuur van het (bedrijfs)netwerk

In bedrijf → onderscheid tussen

- Toegangsnetwerk (access network)
 - Network dat bestaat uit kabels die pc's verbinden met switch/hub
- Distributienetwerk (distribution network)
 - Network dat switches verbindt waarop geen gebruikers komen
- Kernnetwerk (core network)
 - Meerdere distributienetwerken worden verbonden door **kernnetwerk**
 - Bestaat uit routers en snelle verbindingen

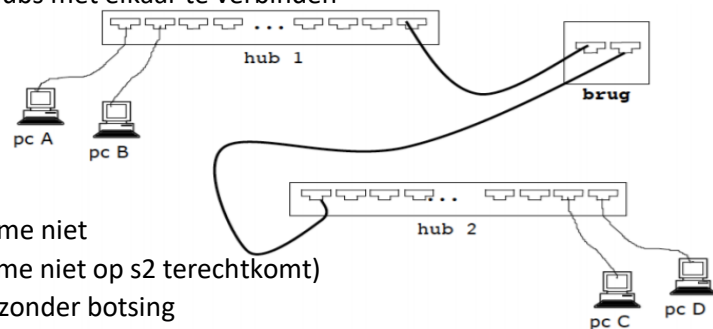


2.12 Brug

2 LAN's met elkaar verbinden gaat door de switches / hubs met elkaar te verbinden

Gaat ook door een brug te gebruiken

Hub1 en de computers vormen segment 1 (zelfde met hub2 en zijn computers)



In brug → adressen van zender en bestemming nagaan

Indien ze op dezelfde poort zijn aangesloten, wordt frame niet

doorgestuurd (A naar B → brug zorgt ervoor dat de frame niet op s2 terechtkomt)

Tegelijk kan een frame worden verstuurd van C naar D zonder botsing

Brug → leert zoals switch welke adressen bij welke poort horen

Verschillende soorten bruggen om 802 LAN's te verbinden (802.3, 802.5,..)

Brug → apparaat dat netwerken kan koppelen op data-linklaag

Switch wordt wel eens **multiport bridge** genoemd

2.13 Frametypes

4 type frames

- Ethernet II
 - Geen lengte veld maar code die aangeeft voor welke toepassing het frame verstuurd wordt (ARP, IP,...)
 - Lengte kan achterhaald worden uit toepassingscode
 - Frames van andere types → lengte aangegeven (2 bytes) → **length veld**
 - Die 2 bytes worden ook gebruikt voor toepassingscode
 - Lengte → nooit groter dan 1500 bytes
 - Code altijd groter dan maximale-frame lengte
- IEEE 802.3
- IEEE 802.2
- SNAP (SNAP → Subnet access point)

Gebruik van 802.x frames → LAN's van verschillende types (bv. Ethernet token-ring) te koppelen in datalinklaag d.m.v. een brug

Koppelen van verschillende netwerken gebeurt in netwerklaag door routers

Ethernet (niet Ethernet 2) → van jaren 80, gemaakt door Digital, Intel en Xerox (DIX)

Driver van netwerkkaart kan alle types aan

TCP/IP als communicatieprotocol kiezen → automatisch Ethernet II gekozen

2.14 Fysieke voorstelling van bits

Bitrijen versturen → fysiek kanaal gebruikt worden → bv. elektrische spanning op een draad

Aantal bits dat verstuurd worden → hangt af van bandbreedte

2.14.1 Binaire codering

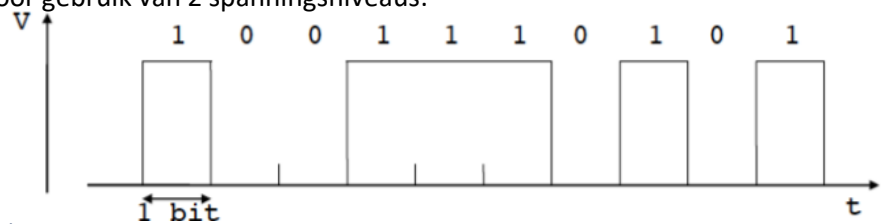
Ook gezegd als → **non return to zero** of **NRZ**

Bits worden voorgesteld door gebruik van 2 spanningsniveaus:

(0 Volt) = 0

(5 Volt) = 1

Bitrij 1001110101 →



2.14.2 Manchester-codering



Hoofdstuk 3: IP (Internet Protocol)

Gebruiksvriendelijk netwerk → IP Programmatuur

IP geïnstalleerd en geconfigureerd →

- Elke computer heeft eigen adres, IP adres
- Elke computer van IP pakketten sturen en ontvangen

Netwerktechnologie die wordt gebruikt speelt geen rol meer → afstand ook niet

Computer kan IP pakket sturen naar andere computer waarvan hij IP adres kent; kan in zelfde gebouw zijn of aan andere wereld

IP → zorgt voor versturing van IP-pakketten zonder garantie; IP is een **niet verbindingsgeoriënteerde, best-effort service**

Voorschriften waarin IP-programmatuur moet beantwoorden → **internet protocol**

Verschillende term voor IP

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

2 versie's van IP op internet: IPv4 en IPv6

Om verder dan eigen LAN te communiceren → **Default Gateways** en **routers** vereist

Moeten correct geconfigureerd zijn

Computers wisselen niet direct IP-pakketten uit maar de IP-pakketten worden door de router doorheen netwerken gerouteerd tot aan de bestemming → IP maakt gebruik van de datalinklaag

→ IP-pakket wordt ingepakt in een frame (type frame hangt af van netwerk)

3.1 IPv4 Adressen

IP-pakketten → moet instaan van waar ze komen en naar waar ze gaan → alle computers in netwerk en netwerk zelf moeten een adres hebben

IPv4 adressen →

- Elk netwerk en computer adres van 32 bits
- 2 delen:
 - Links netwerknummer
 - Rechts computer nummer

Onderscheid tussen 3 klassen netwerken

Klasse A

- Grote netwerken
- 7 bits gebruikt voor netwerknummer, dus 2^7 netwerken → 128
- 24 bits gebruikt voor computernummer → 2^{24} computers per netwerk van klasse A
- Eerste bit van adres is 0
- Vb: 0xxxxxxx. |xxxxxxx.xxxxxxxx.xxxxxxxx
- Netwerk computer

Klasse B

- 14 bits voor netwerk nummer aan te duiden → 2^{14} netwerken
- 16 bits voor computernummer aan te duiden → 2^{16} computers per netwerk van klasse B
- Eerste 2 bits → 10
- Vb: 10xxxxxx.|xxxxxxxx.xxxxxxxxx.xxxxxxxxx
- Netwerk computer

Klasse C

- 21 bits voor netwerk nummer aan te duiden → 2^{21} netwerken
- 8 bits voor computernummer aan te duiden → 2^8 computers per netwerk van klasse A
- Eerste 3 bits → 110
- Vb: 110xxxxx.|xxxxxxxx.xxxxxxxxx.xxxxxxxxx
- Netwerk computer

Adressen die met 1110 beginnen → multicast adressen (of adressen van klasse D)

Adressen die met 11110 beginnen → Klasse E (wordt niet gebruikt)

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

Nog andere uitzonderingen, geen computer met adres 0 → is voor hele netwerk

Allemaal 1 → broadcast-adres → voor alle computers in netwerk

Een computer kan dus nooit een IP adres (in het computernummer gedeelte) alleen maar 0 of 1 hebben

Gepunte notatie → elke groep van 8 bits wordt omgerekend naar een decimaal getal

Vb 11011010 01001101 00110111 01111111 → 218.77.55.127

Met deze notatie → meer dan 4 miljard adressen → te weinig → nieuwe standaard IPv6 → adressen van 128 bits

ICANN (Internet Corporation for Assigned names and Numbers) → bevoegd om netwerk-adressen toe te kennen

Sommige delen → andere bedrijven

Klassen → worden niet meer gebruikt

Nu → met CIDR (classless interdomain routing) notatie; bv: 185.200.24.0/21 → 21 eerste bits is netwerknummer, laatste 11 (32-21) computernummer

Notatie → basisadres/aantal bits

3.2 IP-Pakket

Gebruikers van IP-programmatuur zijn programma's uit transportlaag / hogere transportlagen

Afzender en bestemming niet op zelfde netwerk → router vereist

IP-pakket → **hoofding** en **datadeel**

Lengte hoofding en datadeel liggen niet vast

IP specificeert →

- Welke velden moeten en mogen voorkomen
- Aantal bits dat voor elk van deze velden voorzien is
- Waar (bij welke bit) elk van deze velden begint
 - Eventueel beginpositie afleiden uit opgegeven lengte
- Wat inhoud is van de velden
- Wat de inhoud betekent

Protocol specificeert niet hoelang hoofding en datadeel zijn → **length-veld moet** er zijn

Eerste veld geeft **IPv4** of **IPv6** aan

Hoofding IPv4 pakket →

- Versie (4 bits) → versie is nu 4
- Lengte van hoofding (4 bits: aantal woorden van 32 bits)
- Totale lengte van pakket (16 bits) → totale lengte kan 64Kbyte zijn
- DS-veld (differentiated services, vroeger TOS-veld, type of service)
 - Eerste 6 bits → DSCP (differentiated services code point) veld → hiermee kan type van toepassing weergegeven worden
 - Laatste 2 bits → ECN-veld (explicit congestion notification)
- Identificatie (16 bits) → nummer dat door zender toegekend wordt en wordt gebruikt bij fragmentatie
- Resterende bestaanstijd (time to live/TTL) → zorgt ervoor dat een pakket niet oneindig blijft ronddraaien op netwerk, als het door een router wordt gestuurd wordt het met 1 vermindert
- Protocolnummer → welke bovenliggende module in het TCP/IP model pakket verstuurd wordt
 - Bv 1 → ICMP, 2 → IGMP, 6 → TCP,...
- CRC → alleen voor hoofding wordt CRC berekend → fouten in hoofding kunnen vastgesteld worden door routers; fout → pakket vernietigd
- IP adres van afzender (32 bits)
- IP adres van bestemming (32 bits)
- Framgentatie offset → some bullshit dak ni snap

3.3 ARP (Adress Resolution Protocol)

Programma's die diensten van IP gebruiken → gebruiken IP adressen

IP maakt gebruik van datalink-laag en fysieke laag om bestemming te bereiken

Niveau onder IP-laag → fysieke adressen / MAC adressen

IP adres moet worden omgezet naar een MAC adres → broadcast-frame sturen →

- Op heel netwerk gestuurd
- Type 0x0806 (ARP module)
- ARP boodschap →
 - MAC-adres van verzender
 - IP adres van verzender
 - IP adres van bestemming
 - → IP adres waarvan we het MAC adres willen weten
 - 6 niet gebruikte bytes (voor MAC adres van bestemming)
 - Code die aangeeft dat het een ARP module is
- Alle computers op netwerk lezen de frame en geven de boodschap door aan hun ARP-module

In eigen woorden uitgelegd en niet volgens dit chinees boek →

- ARP broadcast wordt gestuurd op netwerk
- Elke computer krijgt ARP request en checkt of het zijn IP is
- Computer met het juiste IP antwoordt een ARP-boodschap terug met zijn MAC-adres in de 6 lege bytes van de ARP boodschap

3.4 Subnetten en VLSM

Fuck you als ge denkt dak dees uitleg, maak wa oefeningen van Toledo op papier en Packet Tracer

3.5 Router

Router → bevat programmatuur tot en met de 3^{de} laag in OSI model (tot en met netwerklaag)

3 soorten verkeer mogelijk:

- IP-Pakket sturen van computer in subnet X naar computer in subnet X
- IP-pakket sturen van computer in subnet Y naar computer in subnet Y
- IP-pakket sturen van computer in subnet X naar computer in subnet Y (en omgekeerd) → hiervoor wordt router gebruikt

IP pakket naar ander subnet → router stelt vast dat het naar ander subnet moet, pakt IP-pakket opnieuw in in frame en stuurt het door naar het tweede subnet

Router maakt dus deel uit van beide subnetten → router scheidt netwerken

3.6 IP routing

Internet → opgedeeld in allemaal subnetten verbonden met elkaar door routers

Taak van router voor elk IP-pakket

- Nagaan wat de bestemming van IP-pakket is
- Beslissen naar welk machine het IP-pakket moet worden gestuurd
- IP-pakket versturen naar die machine'

Pakket → bereikt bestemming in **etappes** (hops)

3.6.1 Routetabel

- Gebruikt door router = geeft voor elk bestemmingsadres de machine die **next hop (IP-adres van een router (of de interface langswaar deze router bereikt wordt), aanduiding dat directe levering mogelijk is = router heeft netwerkaansluiting met netwerk van bestemming)** is naar deze bestemming.
- Voor de meeste eindbestemmingen is de next hop hetzelfde.
- Moet toelaten om **snel** te weten welke de next hop is.
- Bestaat uit de vorm = **(netwerk, next_hop)**.
- Als in een LAN een computer een IP-pakket verstuurd kan dit zijn =
 - Verzoek aan de server op hetzelfde LAN.
 - Http-verzoek.
- Een IP-adres behoort tot een netwerk als men beide adressen kan enen en dezelfde uitkomst bekomen.
- Van elk IP-pakker zal de router =
 - Eerst nagaan of het voor hemzelf bestemd is.
 - Dan nagaan voor welk net het pakket bedoeld is, netwerknummer wordt bepaald = **netwerknummer van hetzelfde net = IP-pakket inpakken als frame voor eindbestemming (MAC-adres) OFWEL netwerknummer van ander net = IP-pakket in frame verstuurd naar router van Belnet.**
- Routetabellen geven volgende mogelijkheden =
 - Aparte next hop voor specifiek IP-adres.
 - Opdeling in subnetten.
 - Default router opgeven.
- Routetabel wordt gebruikt = **Voor een gegeven IP-adres, zoek** (het masker in elk item enen met het gegeven IP-adres) **vanaf het begin in de tabel tot er een net gevonden wordt, waartoe dit IP-adres behoort. Gebruik aangegeven next hop, indien niet gevonden = stuur foutboodschap naar afzender.**

3.7 ICMP (Internet Control Message Protocol)

- **Vereist voor elke implementatie van TCP/IP** en specificeert =
 - Structuur van de boodschappen.
 - Hoe de boodschappen geïnterpreteerd moeten worden.
- Deze boodschap wordt verstuurd als datadeel van een IP-pakket.
- Vooral gebruikt om **fouten te signaliseren** = wordt verstuurd naar afzender als een router een IP-pakket niet verder kan sturen.
- Geen foutmeldingen over foutmeldingen = **voorkomen van stroom van deze berichten** = eerst nagaan of datadeel van IP-pakket dit soort bericht al bevat (**hoofding van IP-pakket geeft dit aan**).
- Structuur = **ICMP-data** (hoofding en eerste 64 bytes van datadeel van IP-pakket waarvoor dit bericht gestuurd wordt OFWEL ander afhankelijk van type vb antwoord op echo-verzoek) / **ICMP-hoofding** (type 1 byte /code 1 byte(meer info over type)/CRC 2 bytes).
- Echo-verzoek wordt verstuurd na een ping van een gebruiker.
- Bij boodschap bestemming onbereikbaar geeft de code de oorzaak daarvan aan.

3.8 DHCP

- Communiceren via TCP/IP vereist dat =
 - Deze protocol-suite geïnstalleerd is.
 - De computers een IP-adres hebben.
- Aantal redenen om **IP-adres niet vast op te slaan in de computer** =
 - Computer heeft geen harde schijf.
 - Computer is mobiel en wordt in verschillende netwerken opgenomen.
 - Meer computers in netwerk dan IP-adressen.
 - IP-adressen kunnen best centraal beheerd worden door een server.
- Computer vraagt IP-adres aan server = vroeger **BOOTstrap Protocol** (liet geen dynamische toekenning van IP-adressen toe) = nu **DHCP** = computers kunnen IP-adres huren voor één sessie.
- Specificeert de boodschappen die kunnen uitgewisseld worden. DHCP-boodschappen worden via UDP (niet verbindinggerichte tegenstanden van TCP) verstuurd.
- UDP-bericht versturen naar niet gekend IP-adres = **broadcast** = **aanvraag voor IP-adres aan alle computers van het netwerk via UDP-bericht naar poort 67**. Bestemmingspoort voor antwoord is 68.

3.9 IPv6

- 128 bits.
- Geen subnetmasker, aantal netwerkbits expliciet aangegeven.

3.9.1 Adresnotatie

- 8 keer 4 hexadecimale cijfers, gescheiden door een dubbel punt.
- **Leidende nullen mogen weggelaten worden.**
- **Één opeenvolging van nullen** mag vervangen worden door **2 dubbele punten**.

3.9.2 IPv6 hoofding

- 40 bytes.
- Opties gebruiken = extra hoofdingen = na Ipv6-hoofding en voor de data.
- Volgende **velden** =
 - **Versie** (0.5 byte).
 - **Trafiek klasse** (1 byte) = DS-veld.
 - **Stroomlabel** (1.5 byte) = stroom van IP-pakketten met zelfde stroomlabel.
 - **Lengte** (2 bytes) = lengte in aantal bytes van rest van het pakket. Max 65535 bytes.
 - **Volgende hoofding** (1 byte) = wat komt na IPv6-hoofding = TCP-hoofding, UDP-hoofding, **Encapsulating Security Payload** (gebruikt voor encryptie), extra IP-hoofding (hop-by-hop opties hoofding, fragment-hoofding, routing-hoofding, bestemmingsopties-hoofding). Zelfde functie als protocolveld bij IPv4 maar met extra opties.
 - **Hop-limiet** (1 byte) = TTL-veld IPv4.
 - **Adres afzender** (16 bytes).
 - **Adres ontvanger** (16 bytes).

3.9.3 Fragmentatie

- Niet langer mogelijk voor routers om pakketten te fragmenteren als maximale frame-lengte te klein is (**Maximal Transfer Unit**).
- Pakket te groot = router stuurt ICMP-bericht naar afzender.
- Afzender moet zelf nagaan hoe groot pakketten mogen zijn en moet ze zelf fragmenteren = **extra hoofding = fragment-hoofding**.

3.9.3.1 Adressen toekennen

- CIDR-notatie = 2000::/3 = alle IP-adressen waarvan de eerste 3 bits hetzelfde zijn.
- **Regional Internet Registry** = 5 verschillende = deze kennen blokken van IP-adressen toe aan ISP's =
 - **RIPE** = Europa, Midden-Oosten, Noord-Azië.
 - **ARIN** = Noord-Amerika.
 - **LACNIC** = Latijns-Amerika.
 - **APNIC** = China, India tot Australië.
 - **AfriNIC** = Afrika.
- Grote van deze blokken = /23. RIR's kennen blokken van /32 toe aan ISP's.
- Elke ISP kent blokken van /48 toe aan zijn klanten/bedrijven.
- Van een IPv6-adres =
 - Eerste 48 bits = ISP.
 - Laatste 80 bits = toegekend aan bedrijf.

3.9.4 IPv6 subnetten

- Eerste 16 bits van /48 blok toekennen aan subnet. Laatste 64 bits kenmerken de machine uit het subnet.

3.9.5 Soorten IPv6 adressen en hun bereik

- 1 apparaat = meerdere IP-adressen = bereik verschilt. Type adres =
 - **Globaal unicast** = eerste 3 bits = 001 dan global routing prefix (45 bits), subnet-id (16 bits), host-id. Als bestemmingsadres van pakket zich buiten subnet bevindt en router juist geconfigureerd is = router verstuurd pakket op ander subnet.
 - **Lokale link** = behoren tot FE80::/10, andere bits afgeleid van MAC-adres, door router niet op ander subnet verder gestuurd, communiceren met apparaten op het subnet, gebruikt voor automatische adresconfiguratie en om vast te stellen welke andere apparaten er zijn.
 - **Lokale site** = zitten in range FEC0::/10, pakketten enkel verder gestuurd op ander subnet als dit een subnet is binnen dezelfde site, aparte toepassing = als Microsoft IPv6-besturingssysteem geïnstalleerd = automatisch gezocht naar DNS-server op adressen : FEC0:0:0:FFFF::1, FEC0:0:0:FFFF::2, FEC0:0:0:FFFF::3.

3.9.6 Configuratie van global unicast adres

Verschillende manieren

SLAAC → Stateless Address Autoconfiguration

Stateless → geen DHCP server die bijhoudt welke IPv6 adressen al gebruikt zijn

- Statisch → manueel als netwerkadministrator het IP-adres geven
- Dynamisch →
 - SLAAC: gebruik info in Router Advertisement voor IPv6 adres
 - Alleen prefix → gebruik van EUI-64 of random geselecteerd
 - SLAAC + DHCPv6: Adres verkregen via SLAAC; additionele informatie wordt verkregen via een DHCPv6 server, zoals een DNS server
 - DHCPv6 alleen: zoeken van DHCPv6 server

Routers zenden ICMPv6 RA boodschappen via **link local adressen**

3.9.7 Configuratie van link local adres

Link local adres wordt dynamisch gecreëerd gebruik makende van prefix **FE80::/10** en interface ID of MAC-adres of random gegenereerd

3.9.7.1 Multicast IPv6 adressen

IPv6 Multicast hebben prefix **FF00::/8**

Twee types multicast adressen →

- Assigned multicast
 - Adressen die zijn toegewezen aan specifieke functies zoals DHCPv6 server
- Solicited node multicast
 - Vergelijkbaar met all-nodes multicast address, komt overeen met laatste 24 bits van het IPv6 unicast address
 - Gecreëerd wanneer global unicast of link-local unicast adressen worden toegekend
 - FF02:0:0:0:0:1:FF00::/104 combineren met de laatste 24 bits van het unicast adres.
 - Wie is lid van de multicastgroep FF02:0:0:0:0:1:FF00::C?
 - Wordt gebruikt om mac-adressen op te vragen

2 voorbeelden van IPv6 multicast →

- **FF02::1** All-nodes multicast group
 - Alle IPv6-enabled devices behoren hiertoe
 - Analoog aan alle IPv4 **broadcast** adres
- **FF02::2** All-routers multicast group
 - Alle IPv6 routers behoren hiertoe
 - ipv6 unicast-routing global configuration mode command (packet tracer)

Opvragen macadressen bij IPv6 →

- Bij Ipv6, vb. pakket voor 2001::abc:def1
- Vraag : "wie is 2001::abc:def1" versturen naar solicited node adres
- ??? - Laatste 24 bits zijn bc:def1 - Vraag : "wie is 2001::abc:def1" wordt verstuurd naar ff02:0:0:0:0:1:ffbc:def1
- Alle nodes waarvan adres eindigt op bc:def1 krijgen vraag
- 1 node antwoordt

3.9.8 IPv4 en IPv6 samen

- **Dual stack** = zowel IPv4 en IPv6 worden geïnstalleerd en geconfigureerd = indien nodig pakket opnieuw gelicht en ingepakt in pakket van andere versie.
- **Tunnel** = oplossing voor dual stack want daarbij komt het oorspronkelijke bericht (hoofding) niet goed aan = **ganse IPv6 inpakken als data in IPv4-pakket en het zo versturen = IntraSite Automatic Tunnel Addressing Protocol – tunnel.**
- **Dual stack waar je kunt, tunnel waar je moet.**
- Niet ganse protocolstapel die anders is = **alleen het netwerkprotocol is anders.**
- Lagenstructuur = op één laag wordt iets veranderd maar dit heeft geen invloed op de andere lagen.

3.x shit da in dia's stond ma ni in cursus

- ICMPv6 bevat 4 nieuwe protocollen als onderdeel van Neighbor Discovery Protocol (ND or NDP):
 - Router Solicitation message
 - Router Advertisement message
 - Neighbor Solicitation message
 - Neighbor Advertisement message
- Router Solicitation and Router Advertisement Message – Sent between hosts and routers.
- Router Solicitation (RS) message – RS messages are sent as an IPv6 all-routers multicast message.
- Router Advertisement (RA) message – RA messages are sent by routers to provide addressing information

Hoofdstuk 4: transportlaag

4.1 diensten die transportlaag levert

- IP levert geen garantie op correcte aflevering van pakketten.
- Om bruikbaar te zijn voor toepassingen = betrouwbare verbinding nodig.
- Extra programmatuur in **transportlaag**. 2 protocollen =
 - **User Datagram Protocol.**
 - **Transmission Control Protocol**
- IP-module vervult volgende taken =
 - Gegeven IP-pakket naar gegeven IP-adres sturen.
 - Uit inkomend **IP-pakket** datadeel nemen en **doorgeven** aan ICMP-module, transportlaag of eventueel een toepassingsprogramma.
- In lagen boven IP-laag = meerdere processen gebruiken tegelertijd IP-module = sommigen via UDP of TCP andere rechtstreeks. IP-module **weet via protocolveld in IP-hoofding aan wie de inkomende data moet afgeleverd worden.**
- UDP- en TCP-module worden gebruikt door meerdere processen tegelertijd = worden **onderscheiden door poortnummers**. Houden per poort een **wachtrij** bij van inkomende data die gelezen kan worden door proces.

4.2 UDP (User datagram protocol)

- Berichten kunnen verloren gaan of fouten bevatten of in de verkeerde volgorde aankomen.
- UDP voegt volgende functies toe aan IP =
 - Mogelijkheid werken met controlebytes.
 - Gebruik poortnummers = **toelaten multiplexing** = meerdere processen kunnen IP-module tegelijkertijd gebruiken.
- Programma kan communiceren met programma op andere machine via UDP = deze programma's zullen **zelf foutcontrole en behandeling verzorgen**.
- Voordeel UDP = **weinig belasting**, gebruik van **broadcasting**.
- TCP = betrouwbaarder.
- Structuur UDP-bericht =
 - **Hoofding** = bron (poortnummer verzendend proces), bestemming (poortnummer ontvangend proces), lengte (aantal bytes in bericht), controle-veld (controle-bytes = 0000 → geen controle gebeurd, als wel controle gebeurd en 0 → andere code voor 0 gebruikt).
 - **Datadeel**.
- **Inhoud controleveld berekenen** =
 - Pseudo-hoofding toevoegen aan UDP-bericht (IP-adres van bron en bestemming + protocolnummer).
 - Controleveld voorlopig op 0000 gezet.
- Om ontvangende UDP-module controleveld laten berekenen = **IP-adres afzender doorgeven**.
Als meer dan één IP-adres = hetgene doorgeven via welk het is binnengekomen.

4.3 TCP (Transmission control protocol)

- Verbinding is **virtueel circuit** =
 - Verbinding moet eerst tot stand gebracht worden.
 - Daarna is er een soort pijp tussen 2 processen.
- Er loopt een **bytestroom** van de ene naar de andere machine (full duplex).
- Moet volgorde herstellen. Programma levert bytes aan TCP-module = plaatst deze in een buffer, voldoende bytes = TCP-**segment** naar andere machine gestuurd (afgeleverd aan eigen IP-module). Andere machine bewaart deze bytes in buffer (nadat eventueel volgorde hersteld is), **programma's kunnen uit deze buffer lezen**.
- Verbinding wordt bepaald door 2 **eindpunten** = koppel (**IP-adres, poortnummer**).
- Eindpunt moet **foutcontrole** doen en **ACK** (bevestiging=volgnummer van byte die verwacht wordt) sturen als er geen fout is.
- Als segment correct maar voor zijn beurt aankomt, kan ze niet bevestigd worden.
- **Geen foutmeldingen**, als geen bevestiging wordt gestuurd na **wachttijd** (ongeveer gelijk aan de tijd nodig om bericht te versturen en bevestiging terug te krijgen = **Heen- en Terug Tijd**), wordt het segment opnieuw verstuurd.
- Belasting speelt een rol bij HTT = exponentiële verdeling.
- Ontvangende proces leest bytes in de **volgorde waarin ze afgeleverd zijn**.
- TCP houdt 3 wijzers bij = **bevestigd, verstuurd en grens**.
- Deze bytes vormen samen het **venster** (begrensd door wijzers bevestigd en grens) =
 - Verstuurd maar correcte aankomst nog niet bevestigd.

- Mogen verstuurd worden maar nog niet verstuurd.
- Bytes tot voor bevestigd, **worden niet meer bewaard**.
- TCP-module verstuurd bytes zolang verstuurd < grens.
- Als er een bevestiging (bevat ook mededeling over grootte van het venster = dient voor **flow control**) komt wordt bevestigd aangepast.
- Zender kan zoveel bytes sturen als het venster groot is. Ontvanger moet die kunnen opslaan. Venster verkleinen = **zender afremmen**.
- Hoofding van TCP-segment bevat =
 - **Poortnummer van zendend proces**.
 - **Poortnummer van proces dat bestemming is**.
 - **Bytevolgnummer** (4 bytes) = volgnummer van de totale datastroom van de 1^{ste} byte van het datadeel van het segment.
 - **Lengte van de hoofding** = afhankelijk van de opties.
 - **Bevestigingsnummer** = volgnummer van byte die verwacht wordt, van de datastroom in de andere richting. Dit veld heeft enkel betekenis als **ACK = 1**.
 - **Code** (6 bits) = URG = 1 (dringende data), ACK = 1 (bevestigingsnummer heeft betekenis), PSH = 1 (segment wordt verstuurd als gevolg van PUSH-bevel), RST = 1 (reset-segment), SYN = 1 (SYN-segment), FIN = 1 (FIN-segment).
 - **Venstergrootte** = venstergrootte die correspondent mag gebruiken voor andere datastroom.
 - **Controlebytes** = berekend zoals bij UDP, ook met pseudo-hoofding die IP-adressen bevat.
 - **Lengte van het segment**.
 - (eventueel) **Opties**.
- Proces verklaart zich bereid in te gaan op verzoeken om te communiceren = **uitvoeren passieve open-functie** = signaliseren aan eigen TCP-module dat men bereid is te communiceren = besturingssysteem **kent poortnummer toe voor dit proces** = eindpunt is in luisterstand = proces in andere machine kan dan **actieve open-functie uitvoeren** = verzoek aan TCP-module om verbinding tot stand te brengen met **luisterend eindpunt** en ook poortnummer toegekend worden = TCP-module zal **SYN-segment** (SYN-bit = 1, willekeurig volgnummer) **naar ander eindpunt sturen**.
- Als TCP-module zul segment ontvangt = stuurt SYN/ACK-segment =
 - SYN-bit = 1.
 - Bytevolgnummer met willekeurig getal.
 - ACK-bit = 1 → bevestiging.
 - Bevestigingsnummer = willekeurig getal van eerste SYN-segment + 1 → ik verwacht vanaf byte met dat nummer.
- Hierop antwoord TCP-module met ACK-segment =
 - ACK-bit = 1.
 - Bevestigingsnummer = willekeurig getal van bytevolgnummer SYN/ACK-segment + 1 → ik verwacht vanaf byte met dit nummer.
- Verbinding kan nu **full duplex gebruikt worden**.
- Verbinding **beëindigen** (uitzonderlijk) = een eindpunt stelt uitzonderlijke voorwaarde vast = **reset-bewerking uitvoeren** =

- Eigen toepassing wordt verwittigd dat verbinding verbroken wordt.
- Reset-segment (RST-bit = 1) wordt verstuurd naar ander eindpunt.
- Buffers worden opgeheven.
- Eindpunt dat reset-segment ontvangt =
 - Verwittigd prices.
 - Heft buffers op.
- **Normale manier van beëindigen** = initiatief om te stoppen gaat uit van het proces = in één eindpunt geeft proces een **sluit-bevel** = hierna zal dit proces =
 - Overblijvende data versturen.
 - Wachten tot de andere module de ontvangst bevestigd.
 - FIN-segment (FIN-bit = 1, willekeurig bytevolgnummer) versturen.

Als andere module dit FIN-segment ontvangt =

- Stuurt ACK-segment (ACK-bit = 1, verwacht vanaf bytevolgnummer FIN-segment +1).
- Eigen proces wordt verwittigd dat er geen data meer komt.
- Geen data van eerste module wordt nog aanvaard = **één kant van de verbinding is gesloten**, zelf kan deze module blijven sturen en dat zal bevestigd worden.
- Dan stuurt de TCP-module een FIN-segment (FIN-bit = 1, bytevolgnummer willekeurig, door ACK verwacht hij nog altijd van hetzelfde bytevolgnummer want er komt geen data meer).

Als de eerste module dit FIN-segment ontvangt =

- Zendt een ACK-segment (ACK-bit = 1, verwacht vanaf bytevolgnummer van het FIN-segment dat naar deze module verstuurd werd).
- De module heft de verbinding op.

Als de tweede module dit ACK-segment ontvangt, wordt de verbinding opgeheven.

- Alhoewel TCP en IP slecht 2 protocollen zijn, wordt met TCP/IP vaak de ganse verzameling van protocollen bedoeld.
- **Protocol Data Unit =**
 - **TCP = segmenten.**
 - **IP = pakketten.**
 - **Ethernet = frames.**
 - **Fysisch = bits.**

4.4 Adresvertaling: NAT

- 2 computers zouden hetzelfde IP-adres kunnen hebben als =
 - Ze zich in verschillende netwerken bevinden.
 - En deze netwerken niet met elkaar verbonden zijn.
- Oplossing voor beschikbare IP-adressen =
 - Buitenwereld heeft niets te maken met de interne communicatie (**lokale of interne adressen**) van een bedrijf.
 - IP-adressen gebruikt binnen bedrijf verschillen van IP-adressen die in de buitenwereld (**globale adressen**) gebruikt worden.

- Pakketten die van de buitenwereld komen en bepaald IP-adres als bestemming hebben, daarvan wordt het IP-adres vervangen door het oorspronkelijke lokale adres.
- **Network Address Translation** = ganse IP-pakket bewerkt = nieuwe CRC moet berekend worden = moet alleen voor pakketten die voorbij de router moeten.
- Ook de bronpoort wordt vervangen = zo worden **meerdere verbinding opgezet**.
- Routing en NAT = kan ook door geconfigureerde pc met 2 Ethernet-kaarten.
- Een router **zal nooit een IP-pakket met als bestemming een lokaal adres op het Internet zetten**.

Hoofdstuk 5: Applicatielaag en ondersteunde applicaties

5.1 DNS

- Gebruik maken van cliënt/server-toepassing om IP-adres te weten van computer waarvan men de naam kent = er zijn **servers die naam en bijhorend IP-adres kennen van een computer**.
- Protocol =
 - Hoe worden namen toegekend aan computers.
 - Welke berichten kunnen worden uitgewisseld tussen cliënt en server om het IP-adres te vinden via de naam.

Namen-specificatie = ICANN verdeelt in **domeinen** = topdomeinen (nl, be, ...). Sommige duiden op aard, andere op nationaliteit. Per topdomein zorgt organisatie voor verdere verdeling = **subdomeinen**.

- **Organisatie die verantwoordelijk is voor een domein** =
 - Registreert namen.
 - Creëert subdomeinen.
 - Delegeert het beheer over deze subdomeinen.
- Computer is een host met een **hostnaam**.
- 2^{de} taak DNS = **namen vervangen door IP-adressen** = cliënt stuurt vraag naar naam-server, die antwoord. Cliënt moet minstens één IP-adres van een naam-server (**server-proces of computer die dit proces uitvoert, poortnummer 53**) kennen.
- Eerste naam-server binnen bereik van een cliënt = lokale naamserver.
- Cliënt houdt cache bij (naam/IP-adres) om aantal keer dat naam-server gecontacteerd wordt te beperken.
- Proces waarbij machine een IP-adres nodig heeft = **name resolver**. Naam-server wordt geraadpleegd als domeinnaam niet in cache staat. Item in cache wordt maar **beperkte periode** bewaard.

5.1.1 Hoe wordt DNS informatie bewaard?

- Oorspronkelijk bewaard in HOST.TXT, beheerd door Network Information Center van Research Instituut van Universiteit van Stanford.
- Nu = bewaard in **gedistribueerde databank** = elke DNS-server heeft stukje van de informatie = servers kunnen onderling communiceren.

5.1.2 Hiërarchische naamgeving

- Voor elk domein is er een instantie die bevoegd is voor namen binnen een domein =
 - Aan computer een naam toekennen.
 - Subdomeinen creëren.
 - Al of niet beheer van subdomeinen delegeren aan ondergeschikt bestuur.
 - Moet DNS-server installeren = IP-adres daarvan moet bekend zijn bij instantie van ouderdomein.

5.1.3 Zone

- Geen delegatie binnen domein = zone. Subdomeinen horen ook tot zone.
- Beheer van een subdomein gedelegeerd = nieuwe zone met aparte DNS-server.
- Hiërarchische structuur = **naamruimte**.
- **Fully Qualified Domain Name** = domeinnaam is fully qualified als hij eindigt bij de wortel (een punt) van de boom.

5.1.4 Hoe gebeurt een naam resolutie

- **Resolver** = naam vervangen door IP-adres = maakt deel uit van de browser.
- Adres van lokale DNS-server behoort tot TCP/IP-configuratie, samen met IP-adres, subnetmasker en IP-adres van de router.
- Resolver = maakt naam FQDN = vraagt welk IP-adres erbij hoort aan lokale DNS = als die het antwoord niet kent = **root-servers** = kennen het IP-adres van DNS-server van top-domeinen.
- DNS-servers moeten elkaars IP-adres kennen = **lijm waarmee het systeem aan mekaar hangt**.
- **Iteratieve wijze** = als server naam niet weet, geeft hij adres van andere server, zodat hij een stap dichterbij het antwoord is.
- **Rekursieve wijze** = raadpleegt zelf andere servers tot hij het antwoord weet.
- **Forwarder speciëren** = DNS-server die door andere DNS-server moet geraadpleegd worden als die zelf het antwoord niet weet.
- **Berichten verstuurd door resolver en naam-servers** = meestal via UDP = geen TCP-verbinding nodig.

5.1.5 Omgekeerde naamresolutie

- Redenen waarom voor een IP-adres de naam moet kunnen worden opgevraagd =
 - Webserver zou graag weten welk bedrijf/wie komt surfen.
 - Route naar bestemming traceren = **programma tracert** = via omgekeerde naamresolutie verneemt hij ook de namen van routers.
- Hiervoor speciaal domein gecreëerd = **in-addr.arpa**. Welke naam komt overeen met IP-adres x.w.y.z → wat hoort bij FQDN z.y.w.x.in-addr.arpa. ? Zo = komt vraag uiteindelijk terecht bij de server van y.w.x.-in-addr.arpa.

5.1.6 Primaire, secundaire DNS-server

- Voor goede werking DNS = voor elk domein minstens 2 servers. Verschil in functie tussen beiden =
 - Wijzigingen zonegegevens = gebeuren op primaire server.
 - Secundaire heeft kopie van zonedata = regelmatig ververs = **zonetransfer** = gebeurt over TCP-verbinding.
 - Secundaire server = kopie van primaire zone.
- **Dynamisch DNS** = pc die via DHCP IP-adres gekregen heeft, kan naam en IP-adres laten registreren bij zijn DNS-servers.

5.1.7 Records van de DNS-Bank

- Informatie in DNS-databank = **zonedata = resource records = gegevens worden bewaard per zone en niet per domein.**
- Records bestaan uit =
 - Naam (domein of computer).
 - TTL (mag weggelaten worden).
 - Bijna altijd IN van Internet.
 - Type (A, NS, SOA, ...)
 - Afhankelijk van type = aantal parameters die data vormen.
- Types =
 - **A-record** (adres) = IP-adres dat met naam overeenkomt.
 - **PTR-record** (pointer) = naam dat met IP-adres overeenkomt. Bevindt zich in andere databank dan A-record.
 - **NS-record** (name server) = naam DNS-server voor een domein = IP-adres is te vinden in het A-record hiervoor.
 - **SOA-record** (start of authority) = moet in elke zone zijn = geeft aan welke server autoriteit heeft over zone (**autoritatief**) = bevat gegevens over de zone = **naam van de zone, naam primaire DNS-server, email-adres verantwoordelijke van de zone, serienummer** (nummer wordt verhoogd met 1 telkens gegevens gewijzigd worden = gebruikt voor secundaire server om te zien of hij zonetransfer moet doen), **verversingsinterval** (wanneer moet secundaire server verversen, als hij geen verbinding krijgt met primaire server, moet hij een tijdje wachten), **secundaire server mag na een tijd van niet verversen geen gegevens meer verspreiden, records zonden TTL = default TTL.**

Videos om te helpen begrijpen

https://www.youtube.com/watch?v=PpsEqJV_A0

<https://www.youtube.com/watch?v=Pj-27Q27kEc>