

2020 samenvatting - Lars Lemmens

Met dank aan de [Github van Martijn](#) en natuurlijk Lars Lemmens

Alle afbeeldingen staan in de powerpoint!

Hoofdstuk 1

1.1 Protocollen

- Programma dat luistert en wacht tot een ander programma het initiatief neemt = **serverprogramma**.
- Programma dat het initiatief neemt = **cliëntprogramma**.
- Communicatie van computerprogramma's moeten er vastgelegde afspraken zijn →
- Welke berichten er kunnen verstuurd worden
- Wat elk bericht betekent
- Wat de onderlinge verhoudingen zijn tussen de berichten, o.a. hoe op elk bericht kan gereageerd worden
- Geheel van afspraken → **protocol**
- Principe verdeel & heers toegepast → afspraken voor een bepaald deel i.v.m. computercommunicatie
- Afspraken over wijze waarop een bit voorgesteld wordt.
- Afspraken i.v.m. foutcontrole.
- Afspraken om meerdere PC's dezelfde fysieke verbinding te laten gebruiken zonder elkaar te storen
- Verschillende afspraken vastgelegd → verschillende protocollen samen → **protocolstack**
- Protocol gebruikt voor WWW = **HyperText Transfer Protocol**.

Schematische voorstelling HTTP-berichten: (figuur 1.1)

1.2 Protocollagen

- Netwerk kan gezien worden als → infrastructuur die **gedistribueerde applicaties** voorziet van diensten
- Waarom kan eenzelfde browser gebruikt worden bij verschillende netwerken = **browser moet bericht niet zelf omvormen tot een bitrij + bitrij niet zelf op netwerk zetten, zal bericht afleveren aan programmatuur op lager niveau dat bericht omvormt tot een bitrij en deze op netwerk plaatst, idem voor inkomende berichten.**
- Netwerkprogrammatuur opgedeeld in verschillende lagen:

Bovenste laag = applicatielaag	Communicatieboodschappen uitgewisseld tussen toepassingen die gebruik maken van netwerk → applicatieboodschappen direct gevolg van interactie tussen applicatie & gebruiker
Onderste laag = Fysieke	verstuurde bitrij op het netwerk en zet ontvangen signalen van het fysieke communicatiemedium om in een bitrij.

- Indeling in lagen 2 belangrijke voordelen:
- De complexiteit kan verdeeld worden d.m.v. **abstractie**
- Meerdere toepassingen kunnen gebruik maken van dezelfde programmatuur van de onderliggende laag
- Communicatie in figuur 1.1 = **abstract / virtueel** → **berichten worden afgeleverd aan en ontvangen van een tussenlaag.** De bovenste van de tussenlagen.
- Een stuk van bericht, opgedeeld door 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.**
- **Meerderden kunnen het netwerk tegelertijd gebruiken** = netwerk kan segmenten van verschillende communicaties door elkaar versturen.

1.3 TCP/IP

- Voor lagen tussen toepassingslaag en fysieke laag wordt bijna altijd TCP/IP gebruikt.
- Geheel van afspraken =
- Elke computer moet een **uniek IP-adres hebben** van 4 cijfers die tussen 0 en 255 liggen.
- Een computer wordt aangeduid d.m.v. zijn IP-adres.
- Een proces op een computer wordt aangeduid door het **poortnummer**.

1.3.1 TCP/IP toepassingen

- Toepassingen hebben volgende eigenschappen gemeen :
- **Client- en serverprogramma's.**
- **Gecommuniceerd via TCP/IP.**
- **Berichten die uitgewisseld worden, zijn vastgelegd in een protocol.**
- Protocolspecificatie inkijken = **Request For Comment opvragen via FTP.**
- **Internet RFC's** : Cliënt en server voeren protocollen uit die in een computernetwerk het volgende regelen
- Boodschappen en de volgorde ervan die worden verstuurd
- Boodschappen en de volgorde ervan die worden ontvangen
- Structuur boodschappen
- Protocollen moeten **publiekelijk** en **gestandaardiseerd** worden.

1.4 TCP/IP-model van de tussenlagen

- De toepassingsprogramma's doen beroep op de programmatuur van een lager niveau. Meestal is dat het TCP-programma. Implementatie van TCP, transfer control protocol.
- Verstuurt en ontvangt berichten die voldoen aan het TCP → deze berichten worden TCP-segmenten genoemd. (TCP-programmatuur wordt gebruikt door de toepasingsprogrammatuur en staat dus op een lagere laag)
- TCP-programma's wisselen virtueel met andere machines segmenten uit
- Er wordt gebruik gemaakt van programmatuur op een lagere laag: IP-programma. Implementatie van het IP (internet protocol). Berichten die voldoen aan het IP worden IP-pakketten genoemd
- IP-programma cliënt en server communiceren virtueel
- IP-programma's maken gebruik van programmatuur op een lagere laag: bv. de programmatuur of driver voor de netwerkkaart. De berichten die op die laag verwisselt worden zijn frames
- Om frames van de ene naar de andere machine te sturen wordt er gebruik gemaakt van een lagere laag. Deze onderste laag gaat frames omzetten in bits.
- De onderste laag die de frames in bits omzet gaat d.m.v spanningen (1 en 0) de data overzetten van de ene naar de andere machine

Applicatielaag (HTTP, FTP)	Besturingssysteem met netwerktoepassingen
Presentatielaag (SSL, SSH)	Heeft betrekking op afspraken voor het coderen en decoderen van gegevens voor de toepassingslaag

Applicatielaag (HTTP, FTP)	Besturingssysteem met netwerktoepassingen
Sessiel laag (Winsock, Sockets)	Heeft betrekking op de communicatie tussen tweetoepassingsprocessen
Transportlaag (TCP, UDP)	Heeft betrekking op afspraken om een pakket zonder fouten van bron naar eindbestemming te sturen, maar op een hoger niveau dan de datalink-laag (er kunnen meerdere tussenstations zijn & omdat pakketten kunnen opgesplitst worden)
Netwerklaag (IP, ICMP)	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 fysieke adressen
Datalink-laag (Ethernet, PPP)	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 handshake gebeurt.
Fysieke (Coax, Fiber)	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 ene laag **mogen de afspraken voor andere laag niet**** beïnvloeden**.
- Uitzondering fysieke laag worden lagen → **programmatuur geïmplementeerd**.
- Programmatuur in een bepaald laag **ziet**** de **programmatuur** in **de** onderliggende **laag** als subroutines**.
- Elke laag communiceert minimaal met laag er onder / boven → alleen met die lagen → programmatuur ontwikkelt voor bepaalde laag → er vanuit compatibel met alle lagen compatibel

1.6 TCP/IP-model versus OSI-referentiemodel

OSI-model (theoretisch model)	TCP/IP-model (reëel model)
Toepassingslaag	Toepassingslaag
Presentatielaag	

| | Sessiel laag | | Transport laag | Transport laag | | Netwerkl laag | Internet laag | | Datalink-laag |
Datalink-laag | | Fysieke laag | Fysieke laag |

- **Maar 4 lagen** =
- **Toepassingslaag** = bovenste 3 OSI
- **Transportlaag** = zelfde als OSI
- **Internetlaag** = zelfde als netwerkl laag OSI
- **Netwerktogangslaag** = onderste 2 OSI

HOOFDSTUK 2: Datalinklaag: Netwerken en LANs

2.1 Terminologie: Link, node, netwerk, PDU

- **Node** = een apparaat dat informatie uitwisselt over een netwerk
- **Link** = communicatiekanaal tussen aangrenzende nodes → broadcastlink en point-to-point-link
- **Broadcastcommunicatie** = wanneer nodes verbonden zijn door middel van een gedeelde broadcastlink zal er een medium-access-protocol nodig zijn die toegang tot de link en het verzenden van frames op deze link in goede banen zal leiden
- **Point-to-pointcommunicatielink** = deze wordt gebruikt wanneer twee nodes rechtstreeks met elkaar communiceren
- **Netwerk** = twee of meerdere nodes die geconnecteerd zijn op een gedeeld medium/link en die daardoor rechtstreeks met elkaar kunnen communiceren
- **PDU** = de data dat verstuurd wordt tussen twee nodes en gespecificeerd wordt in een protocol

2.2 Diensten van de datalinklaag

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

Datalink voorziet 3 basisdiensten →

- Een duidelijk interface (API) met de netwerkl laag voorzien en frames vormen.

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

Frame bevat:

- Frameheader
- Framepayload, hierin pakket hoger gelegen laag
- Frametrailer (framefooter)

2.3 Implementatie van de datalinklaag

Datalinklaag →

- Inhardware
- op netwerkkkaart(NIC)
- op moederbord (NIC geïntegreerd opmoederbord)
- Insoftware
- Softwarecomponent van datalinklaag →driver vanNIC →Deze handelt adresinformatie en interruptafhandelingen van de NIC hardware af

2.4 Multiple-accesslinks en -protocollen

- **Willekeurige-toegangsprotocollen**
- De computer wil data verzenden, wacht tot link vrij is en begint met zenden. Mogelijke botsingen wanneer 2 of meerdere nodes op hetzelfde moment frames verzenden. Na botsing wordt toestand hersteld. Ook **wel toegang door contentie** genoemd. **Ethernet** is hier een voorbeeld van.
- **Deterministische toegangsprotocollen**
- Een computer wacht tot hij toegang krijgt. Als een PC toegang krijgt → zekerheid dat geen andere PC's zullen storen → vaak gebruik gemaakt van een token. Token = bewijs van toegang tot het verzenden van frames op de link. → Maar 1 token dus botsingen niet

mogelijk

Types LAN: Ethernet, token bus, tokenring,... Ethernet = meestgebruikt

2.5 Ethernet: fysiek

- Ethernetkaart → uniek adres
- Bestaat uit 48 bits → 12 Hexadecimale cijfers = **MAC-adres** (Medium Access Control)
- Of uit LAN-adres
- Op ethernetkaarten UTP-kabel gestoken die verbonden is met switch of hub
- Switch bevat 4,8,12,... poorten → poorten: aansluiting voor TP-kabel
- RJ-45 = connector
- TP-kabel 8 draden waarvan bij 100Mbit/s 4 gebruikt van worden
- Crossover-cable → omwisseling tussen 2 PC's → 1-3, 3-1, 6-2, 2-6
- Meerdere PC's verbinden → switch → alles wat door draad 1 & 2 verzonden wordt → in switch via 3 & 6 doorgestuurd → via rechte kabel verbonden met elkaar

2.6 Ethernet frames

- Via Ethernet kunnen pc's Ethernet-frames naar elkaar versturen en van elkaar ontvangen. Een **Ethernet-frame is een bitrij** en bevat:
- Adres van de bestemming en de zender (het **MAC-adres** van 6 bytes)
- **Type** (2 bytes)
- **Data** (max. 1500 bytes, ook maximum transmission unit (MTU) genoemd)
- Eventueel **opvulbytes** (lengte frame min. 64 bytes en max. 1518 bytes)
- Controle-informatie (**frame sequence checksum**) die nagaat of er geen fouten bij overdracht zijn gemaakt (4 bytes)
- Om ethernetkaart te kunnen gebruiken → drivers nodig → drivers die frame versturen & ontvangen
- **verschil fysieke laag & datalink** → op fysieke laag bestaan bits → bits groeperen tot frames → frames versturen → **frames versturen en ontvangen op datalinklaag**
- **Preamble:** Aan ethernetframe → preamble (7 bytes) & SFD (start of frame delimiter, 1 byte) → samen

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 via 1 coax kabel → bus(-broadcast)netwerk
- Elke bit die op kabel ging passeerde elke PC
- Op uiteinden kabel een terminatie (kabelafsluitingen → voorkomen 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 alle PC's → alleen PC voor wie frame bestemd is mag frame lezen → elke kaart leest bestemmingsadres frame
- Is dit mijnadres
- Is dit mijn 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
- **Carrier Sensing**
- **MA → Multiple Access** → meerdere knooppunten tegelijk toegang tot link
- **(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 kabel vrij is
- **Collision Detection**
- Kaarten moeten kabel inspecteren om **detecteren** of er botsing is
- Botsing → frames verloren → Kaart JAM signaal sturen een bepaalde tijd → Zo weet elke kaart op netwerk dat er botsing was
- **Na botsing** → kaarten die iets wouden sturen wachten willekeurige tijd voor ze weer opnieuw iets sturen

Nog botsingen mogelijk bv:

- PC A & B → 1km 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 gebotst frame → runt
- Na sturen 64ste byte → normaal geen botsing meer
- Wel nog mogelijk, noemt **late collision**
- **NIET** opgemerkt door kaarten → frame niet opnieuw herverstuurd
- Programma van hoger niveau (op OSI) zal dit moeten oplossen

2.8 Snelheid

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

- Een frame van de ene kant van de kabel naar de andere kanteizen
- 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 frame op bus plaatst, gaat kaart na of het botst → als eerste 64 bytes van frame volledig op bus kunnen geplaatst worden zonder dat kaart vaststelt dat er een botsing is, is kaart zeker dat er geen botsing meer kan zijn

2.9 Switch

- CSMA/CD in kort → er kan niet meer dan 1 keer frames verstuurd worden, anders is er een botsing
- Kan beter → intelligente hub (→ **Switch**) gebruiken die botsingen 'vermindert'
- Als ethernet frame in switch aankomt, leest switch eerste adres van bestemming → frame dan verstuurd naar poort waarop bestemming zich bevindt

SWITCH (geschakeld Ethernet)

- Krijgt frame en kijkt naar bestemming ervan. Het frame wordt alleen naar de poort van de bestemming verzonden.
- Kent ethernet en kan een frame analyseren en het adres herkennen
- Werkt op laag 3 van het OSI-model (Netwerk)

HUB (gedeeld Ethernet)

- Leest geen adressen en stuurt elk inkomend signaal (frame) naar elke poort (behalve die vanwaar het frame is binnengekomen)

- Elektrisch apparaat dat alleen een bit herkent (elektrische spanning)
- Werkt op laag 2 van het OSI-model (Datalink)
- Met switch → sniffen (wanneer iemand alle frames doet lezen op zijn netwerkkaart) vermeden
- Geheugen switch → inhoud adresseerbaar (CAM-geheugen) → opgeslagen info MAC-adressen → aan switchpoort bv. Hub of andere switch zitten d.w.z → op 1 poort meerdere MAC-adressen mogelijk
- **Microsegment**
- 1 kabel die 1 pc verbindt met switch (geen hub, maar een switch) → microsegment genoemd → dus als segment beschouwd worden
- **Frame behandeling**
- **Cut-through switching**
- Zodra switch eerste bytes van frame met bestemmingsadres gelezen heeft → switch frame verder sturen op poort waar bestemming zich bevind
- **Store-and-forward switching**
- Switch leest hele frame & checkt op fouten (Via CRC) → geen fouten → frame verstuurd
- **Fragment-free switching**
- Pas nadat switch eerste 64 bytes gelezen heeft → 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
- Configuratie
- Broadcasts → beperkt tot VLAN (bv. Management, verkoop)

2.10 Soorten Ethernet

- Oude ethernet van 10 Mbit/s → verdwenen
- Opvolger → **Fast Ethernet**

2.10.1 Fast Ethernet

- Ook CSMA/CD als protocol
- 100 Mbit/s
- Niet via coax
- Kan zich aanpassen aan het oude Ethernet

- 10 keer sneller → lengte 10 keer korter → max. 250m

2.10.2 Gigabit Ethernet

- Ook CSMA/CD als protocol (een netwerkkaart luistert of er een botsing is terwijl ze bits op het netwerk zet)
- 1000 Mbit/s
- 100 x sneller → lengte 100 x korter → max. 25m
- **Full duplex** = verbinding waar tegelijk info van zender naar ontvanger en andersom kan

2.11 Gesstructureerde bekabeling

2.11.1 Fysieke structuur

- Er bevinden zich evenveel RJ45 contactpunten als er computers moeten aangesloten worden.
- Er loop van elk contactpunt een TP-kabel naar een centraal schakelpaneel.
- Een pc wordt fysisch opgenomen in het netwerk door de netwerkkaart via een TP-kabel te verbinden met het contactpunt en het overeenkomstig contactpunt op het schakelpaneel te verbinden met een poort van de switch.
- Bekabeling van pc's tot schakelpaneel = **horizontale bekabeling**.
- Schakelpaneel = **Horizontal Cross Connect** = distributiebekabeling.
- Standaarden van **Electronic Industries Association / Telecommunication Industries Association** =
- Kabel van pc tot stopcontact = max 10m.
- Kabel tussen stopcontact en schakelpaneel = max 90m.
- Kabel tussen schakelpaneel en switch = max 5m.
- Verbinden van verschillende verdiepingen met elkaar of met de servers = **verticale bekabeling = backbone**.
- Switches aangesloten op contactpunten met UTP naar centraal schakelpaneel = **Vertical Cross Connect**.
- Als lengte verticale bekabeling meer dan 100m bedraagt, kan men geen TP-kabels gebruiken.

2.11.2 Hiërarchische structuur

- Netwerk enige omvang = onderscheid tussen =
- **Toegangsnetwerk** = het netwerk dat de pc's verbindt tot aan de eerste switch inclusief die switch.
- **Distributienetwerk** = het netwerk dat die switches, waarop geen gebruikers voorkomen.
- **Kernnetwerk** = verbinding tussen distributienetwerken, bestaande uit routers en snelle verbindingen.

2.12 Brug

- Gebruikt om **2 Ethernet-LAN's rechtstreeks te verbinden**.
- Brug is intelligent zoals een switch = nagaan van adres van zender en bestemming. Als ze op dezelfde poort aangesloten zijn, dan wordt het frame niet verstuurd via de andere poort.
- Kan **verschillende type's LAN's verbinden**. Complexe taak bij omzetten
- **802.3** = wachten tot drager vrij is en fouten corrigeren.
- **802.5** = wachten tot men het recht krijgt om te zenden.
- Ook kunnen de maximale frame-lengtes verschillen.
- **Apparaat dat netwerken kan koppelen in de datalink-laag**.
- Switch wordt soms **multiport bridge** genoemd.

2.13 Frametypes

- **4 Types:**
- **Ethernet II**
- **IEEE 802.3**
- **IEEE 802.2**
- **SNAP (= Subnet Access Point)**

2.14 Fysieke voorstelling bits

- Om bitrijen te versturen → 1 of ander fysiek signaal & medium gebruikt worden → bv (elektrische spanning op draad) → aantal bits per seconde verstuurd kan worden → afhankelijk bandbreedte

2.14.1 Frametypes

Non return to zero of **NRZ** → Bits worden voorgesteld door gebruik van 2spanningsniveaus: (0 Volt) = 0 & (5 Volt) = 1 Bitrij 1001110101 →

2.14.2 Manchester coding

- 0 wordt voorgesteld door sprong van laag naar hoog, 1 als sprong van hoog naar laag.
- Halfweg elke bit is er een sprong → **laat toe dat klok van de ontvanger perfect gelijk loopt met klok van de zender.**
- Voordeel = **voorstelling van een bit bevat ook klok-informatie.**
- Nadeel = **pulsen zijn maar half zo breed → 2 keer zoveel bandbreedte nodig.**
- Techniek oude versie van Ethernet.

Hoofdstuk 3 Internet Protocol (IP)

- LAN-frames kunnen niet naar buiten gestuurd worden.
- Eerste stap naar meer gebruiksvriendelijk netwerk = **IP-programmatuur** (volgens het **Internet Protocol**)=
- Elke computer heeft een uniek adres.
- Computers kunnen IP-pakketten naar elkaar sturen.
- IP-pakket wordt verstuurd naar een computer waarvan het IP-adres gekend is.
- IP-pakket wordt ingepakt in een frame dat verstuurd wordt, het type frame hangt af van het netwerk.
- IP-pakket = waar ze vandaan komen, waar ze naartoe moeten.
- PC krijg IP-adres van 32 bits → **netwerknummer & computernummer**

- De netwerktechnologie (Ethernet, Token Ring, ATM ,...) die gebruikt wordt om het netwerk op te zetten, **speelt geen rol meer.**
- IP zorgt voor versturing van IP-pakketten zonder garanties → = niet **verbindingsgeschiede, best-effort service**
- **IP** betekent strikt genomen: het geheel van afspraken
- **IP** wordt ook gebruikt in de betekenis van: de implementatie of het computerprogramma van deze IP afspraken

3.1 IPv4-adressen

- **Klasse A** = 7 bits gebruikt voor netwerknummer, max 128 netwerken. Eerste bit van adres is 0.
- **Klasse B** = 14 bits gebruik voor netwerknummer, max 16384 netwerken. Eerste twee bits van adres zijn 10.
- **Klasse C** = 21 bits gebruikt voor netwerknummer, meer dan 2 miljoen netwerken. Eerste drie bits van adres zijn 110.
- **Multicast-adressen** = adres begint met 1110.
- Geen computer met nummer 0. Computernummer gelijk aan 0 stelt het hele netwerk voor. Computernummer met allemaal enen = **broadcastnummer.**
- **NIET LANGER DE EERSTE BITS VAN HET IP-ADRES DIE BEPALEN HOEVEEL BITS HET NETWERKNUMMER VOORSTELLEN → / NOTATIE → basisadres / aantal bits = CIDR (Classless Interdomain Routing)**

3.2 IP-pakket

- Gebruikers IP-programmatuur = programma's uit transportlaag of uit hogere lagen.
- Afzender en bestemming niet op hetzelfde fysisch netwerk = tussenkomst van router nodig.
- Versturen IP-pakketten (datagrammen) = **verbindingloze dienst.**
- Bestaat uit **hoofding** en **datadeel**. Lengtes van deze delen liggen niet vast. IP specificeert =
- Welke velden moeten en mogen voorkomen.
- Aantal bits voor elk veld voorzien.
- Waar elk veld begint.
- Wat inhoud van de velden kan zijn.
- Wat deze inhouden betekenen.
- **Hoofding van een IPv4-pakket** bevat o.m. =
- Versie (4 bits).
- Lengte van de hoofding (4 bits : aantal woorden van 32 bits) = hangt af van aantal opties.
- Totale lengte van het pakket (16 bits).

- **Differentiated Services** = eerste 6 bits zijn **Differentiated Services Code Point** (type van toepassing weergeven), laatste 2 bits zijn **Explicit Congestion Notification** (melden van router of congestie komt/is).
- Identificatie (16 bits) = nummer door zender toegekend.
- **Resterende bestaanstijd** = zender vult deze tijd in, pakket door router verstuurd = tijd met 1 verminderend, als 0 = pakket weggegooid = afzender krijgt foutmelding = **voorkomt dat bij een fout een pakket zou blijven ronddraaien in het netwerk.**
- **Protocolnummer** = voor welke module wordt pakket verstuurd.
- **CRC** = wordt alleen voor hoofding berekend = router kan fout vaststellen = pakket vernietigen = elke router moet dit opnieuw berekenen.
- IP-adres afzender (32 bits).
- IP-adres bestemming (32 bits)
- Fragmentatie offset: Helpt eindbestemmingsapparaat fragmenten in juiste volgorde te plaatsen om het originele pakket te maken → 13 bits breed → van 0 tot 8191 gaan

3.3 ARP

- Programma's die diensten van IP gebruiken → gebruiken IP adressen
- IP maakt gebruik van datalink-laag & fysieke laag om bestemming te bereiken
- Niveau onder IP-laag → fysieke adressen / MAC adressen
- IP adres moet omgezet worden naar een MAC-adres → broadcast frame sturen
- ARP broadcast wordt gestuurd op netwerk
- Elke PC krijg ARP request en checkt of het zijn IP is
- PC met juiste IP antwoordt een ARP-boodschap terug met zijn MAC-adres in de 6 lege bytes van de ARP boodschap.

3.4 Subnetten & VLISM

- **Een subnet bestaat uit computers die deel uitmaken van hetzelfde fysiek netwerk.**
- DE REST ZELF KUNNEN

3.5 Router

- Router bevat programmatuur tot en met 3e laag OSI
- IP-pakketten van een computer van subnet X naar een computer van subnet X
- IP-pakketten van een computer van subnet Y naar een computer van subnet Y

- IP-pakketten van een computer van subnet X naar een computer van subnet Y (en omgekeerd)

3.5.1 Verbinden van netwerken

- Ethernet-LAN's in elkaars buurt verbonden worden door hubs/switches → kan leiden tot grote belasting van netwerk → hub stuurt alle frames door. Een switch stuurt nog altijd het broadcast-frame verder
- Om netwerken verschillende soorten te verbinden router gebruikt. Een router stuurt het broadcast-frame niet verder. Om deze reden routers ook gebruikt als netwerken van dezelfde soort zijn.

3.6 IP routing

- Om te weten waar IP-pakket naartoe moet → routetabel geeft → voor elk IP-bestemmingsadres de machine die **next hop** is naar deze bestemming
- **Next hop**
- het IP-adres van een router
- een aanduiding dat directe aflevering mogelijk is (router heeft netwerk aansluiting met netwerk bestemming)
- **Besluit:** De routetabel als volgt gebruikt. Voor gegeven IP-adres, zoek in tabel tot er een net gevonden wordt, waartoe dit IP-adres behoort. Gebruik aangegeven next hop. Indien geen gevonden → foutboodschap. Zoeken van net kan gedaan worden door masker in item van tabel te lenen met gegeven IP-adres. Als resultaat hetzelfde is dan item, dan is next hop gevonden

3.7 ICMP

- **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 verbindingsgerichte tegenstand 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).
- **Trafiel 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**.

IPv6 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 toe kennen aan subnet. Laatste 64 bits kenmerkend de machine uit het subnet.

3.9.5 Soorten IPv6 adressen en bereik

- **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, router juist geconfigureerd is = router verstuurt pakket op andersubnet.
- **Lokale** link**** = behoort 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 andersubnet 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

- SLAAC → Stateless Address Autoconfiguration
- Stateless → geen DHCP server die bijhoudt welke IPv6 adressen al gebruikt zijn
- Statisch → manueel admin IP-adres geeft
- Dynamisc
- SLAAC: gebruik info in Router Advertisement voor IPv6adres
- Alleen prefix →gebruik van EUI-64 of randomgeselecteerd
- SLAAC+DHCPv6:AdresverkregeenviaSLAAC;additioneleinformatiewordtverkregeenvia een DHCPv6 server, zoals een DNSserver
- DHCPv6 alleen: zoeken van DHCPv6 server

3.9.7 Configuratie van link lokal 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 Multicast IPv6 adressen

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

Twee types multicast adressen →

- Assignedmulticast
- Adressen die zijn toegewezen aan specifieke functies zoals DHCPv6server
- Solicited nodemulticast
- Vergelijkbaarmetall-nodesmulticastaddress,komt overeenmetlaatste24bitsvan het IPv6 unicastaddress

- Gecreëerd wanneer global unicast of link-local unicast adressen wordentoegekend
- FF02:0:0:0:1:FF00::/104 combineren met de laatste 24 bits van het unicastadres.
- Wie is lid van de multicastgroepFF02:0:0:0:1: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 behorenhiertoe
- Analooq aan alle IPv4 **broadcast** adres
- **FF02::2** All-routers multicastgroup
- Alle IPv6 routers behorenhiertoe
- ipv6unicast-routingglobalconfigurationmodecommand(packettracer) Opvragen macadressen bij IPv6→
- Bij ipv6, vb. pakket voor2001::def1
- Vraag : "wie is 2001::def1" versturen naar solicited nodeadres
- ???-Laatste24bitszijnbc:def1-Vraag:"wieis2001::def1"wordtverstuurdnaar ff02:0:0:0:1:ffbc:def1
- Alle nodes waarvan adres eindigt op bc:def1 krijgenvraag
- 1 nodeantwoordt

3.9.8 IPv4 en IPv6 samen

- **Dual** stack****=zowelIPv4enIPv6wordengeïnstalleerdengeconfigureerd=indiennodig pakket opnieuw gelicht en ingepakt in pakket van andereversie.
- **Tunnel** =oplossingvoordualstackwantdaarbijkomthetoorspronkelijkebericht(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=opéénlaagwordtietsveranderdmaarditheetgeeninvloedopdeandere lagen.

EXTRA:

- ICMPv6bevat4nieuweprotocollenalsonderdeelvanNeighborDiscoveryProtocol(NDor NDP):
- Router Solicitationmessage
- Router Advertisementmessage
- Neighbor Solicitationmessage
- Neighbor Advertisementmessage
- Router Solicitation and Router Advertisement Message – Sent between hosts androuters.
- RouterSolicitation(RS)message-RSmessagesaresentasanIPv6all-routersmulticast message.

- RouterAdvertisement(RA)message-RAmessagesare sent by routersto provide addressing information

Hoofdstuk 4: transportlaag

4.1 Diensten die de 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 tegelijkertijd IP-module = sommigen via UDP of TCP en 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 tegelijkertijd = 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 tezelfdertijdgebruiken.
- Programmakann 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 0 → andere code voor 0 gebruikt).
- **Datadeel**
- **Inhoud controle veld berekenen**
 - Pseudo-hoofding toevoegen aan UDP-bericht (IP-adres van bron en bestemming + protocolnummer).
 - Controle veld voorlopig op 0000 gezet.
 - Om ontvangende UDP-module controle veld laten berekenen = **IP-adres afzender doorgeven**. Als meer dan één IP-adres = het gene doorgeven via welk het is binnengekomen.

4.3 TCP

- TCP brengt zekerheid
- TCP werkt full duplex → bytestroom van zender naar ontvanger en omgekeerd op zelfde moment
- IP geen correcte volgorde → TCP herstelt volgorde

1. Een programma levert bytes aan TCP-module
2. Module regelt TCP-segmentatiestroom (flow control)
3. TCP-module neemt datastroom van gebruikersproces
4. Deelt datastroom in blokken
5. Verstuurt elk blok apart in een IP-datagram of pakket

! TCP-module bepaalt de segmentatiestroom en hoe groot de datablokken mogen zijn

! TCP-verbinding is een bytestroom en geen berichtenstroom

→ TCP-module weet niet wat die bytes betekenen

- De TCP-module plaatst de ontvangen bytes in een buffer
- Als genoeg bytes → TCP-segment gestuurd naar de andere machine
- Andere machine bewaart deze in een buffer (als deze correct en in volgorde zijn)

- Programma kan bytes lezen uit deze buffer

! UDP heeft geen verbinding, maar TCP wel (virtueel circuit)

TCP-verbinding = tussen 2 eindpunten → socket → socketadres

- **Socketadres:**
- IP-adres van de host
- Poortnummer

Bv. [(192.18.16.7, 317);(180.17.126.5, 25)]

! Socket kan door verschillende verbindingen op zelfde moment gebruikt worden

4.3.1) Het TCP protocol

- Elke byte in een TCP verbinding heeft eigen volgnummer (4 bytes)
- Zendende en ontvangende TCP-modules wisselen data via segmenten
- TCP-segment heeft header (20 bytes) gevolgd door 0 of meerdere databytes
- De databytes kunnen samengenomen worden uit meerdere schrijfacties van of naar socket

TCP = betrouwbaar:

- Eindpunt TCP-verbinding moet een ACK (bevestiging) sturen als alle data correct is ontvangen
- d.m.v foutcontrole
- ACK bevestiging vermeldt byte dat verwacht wordt
- Bv. X, d.w.z alles tot X-1 is correct ontvangen en zender mag vanaf byte met nr. X verder sturen

! Als een segment correct, maar voor zijn beurt aankomt kunnen we niet checken of deze juist is. Bevestiging is byte en niet segment gerelateerd

! Als er fout is → geen bevestiging → segment opnieuw versturen na voorafgelegde tijd → RTT (round trip time) = tijd dat nodig is voor segment van zender naar ontvanger en een bevestiging van ontvanger naar zender te sturen

(Hoe belaster het netwerk hoe groter RTT)

- **Venster**
- Verzendende proces levert bytes aan TCP-module → ontvangende proces leest deze bytes in volgorde waarin ze aangekomen zijn
- ! Wanneer zender een segment stuurt → start timer

- Ontvanger stuurt een segment (indien data aanwezig steekt dit ook in het segment) naar zender met bevestigingsnummer v/d volgende byte die hij verwacht en resterende venstergrootte
- ! Als timer op is voordat bevestiging is ontvangen → segment opnieuw verzenden
- TCP houdt aan zendende zijde rij bytes bij → TCP-module verstuurt byte → toegevoegd aan ri
- De rij heeft 3 delen (fig4.1)
- TCP houdt 3 wijzers bij (pointers)
- **De bytes die:**
 - Verstuurd zijn en waarvan de (correcte) aankomst nog niet bevestigd is
 - De bytes die mogen verstuurd worden maar nog niet verstuurd zijn
 - = venster
- **! De wijzers genaamd bevestigd en grens, begrenzen het venster**
 - Bv. De ontvangende TCP-module meldt dat de bytes tot nr. 500 correct zijn ontvangen
 - → Bevestigd krijgt waarde 500 en grens zal met 300 stijgen
- **! Een bevestiging bevat ook info die zegt of het venster vergroot of verkleint**
 - Het venster dient voor stroomcontrole (flow control)
 - De zender kan zoveel bytes sturen als het venster groot is
 - De ontvangende zijde moet die kunnen opslaan
 - ! Door het venster te verkleinen kan men de zender afremmen (bv. zodat ontvanger niet overbelast geraakt)
- Poorten
 - ! Zoals UDP, gebruikt TCP poorten voor de communicatie met de toepassingen
- TCP-segmenten
 - PDU dat verstuurt wordt door TCP-module heet segment
 - Segment heeft header (20 bytes + eventuele TCP opties mogelijk)
- **Headerveld:**
 - Poortnummer zendend proces
 - Poortnummer ontvangend proces
 - Bytevolgnummer (4 bytes) → nr van byte dat nog niet verstuurd is
 - Lengte header → geeft aan uit hoeveel 32 bit woorden de header bestaat (hangt af van extra TCP opties)
 - Bevestigingsnummer: volgnummer v/d byte die verwacht wordt in andere richting
 - 6 één bit TCP-vlaggen (zie uitgebreide cursus p76)
 - Venstergrootte → geeft de datastroom aan die mag gebruikt worden
 - Controlebytes = die van UDP
 - Lengte van het segment
 - Eventuele opties → extra functionaliteiten voor TCP-protocol
 - Een verbinding tot stand brengen

(zie cursus p77)

- Stroomregulering geregeld door sliding window
- De waarde van het window geeft aan hoeveel data er mag verzonden worden
- ! Als waarde venster = 0 dan is ontvanger niet in staat data te ontvangen
- Verbinding beëindigen

- Uitzonderlijk via reset → een eindpunt geeft aan de het een reset wilt:
- Eigen proces verwittigd dat verbinding verbroken wordt
- Er wordt een reset-segment (RST-bit = 1) naar ander eindpunt gestuurd
- Buffers worden opgeheven
- Eindpunt ontvangt reset-segment:
- Verwittigd het proces
- Buffers worden opgeheven

Normaal beëindigen (omdat full duplex → gezien als 2 simplex verbindingen)

1. **TCP-module (A) na sluit-bevel:**

- Overblijvende data in buffer versturen
- Wachten tot B ontvangst bevestigt
- Een FIN-segment sturen (FIN-bit = 1 en bytevolgnummer bv. X)

1. **TCP-module (B) ontvangt FIN-segment :**

- Een ACK-segment teruggestuurd (ACK-bit = 1 en bytevolgnummer X+1)
- Eigen proces verwittigd dat er geen data meer zal komen van A
- Geen data meer van A ontvangen, maar B kan nog steeds sturen
- A ontvangt data van B

1. **TCP-module (B) ontvangt sluit-bevel:**

- Een FIN-segment sturen (FIN-bit = 1 en bytevolgnummer bv. Y)

1. **TCP-module (A) ontvangt FIN-segment :**

- A zendt ACK-segment (ACK-bit = 1 en bytevolgnummer Y+1)
- A heft verbinding op

→Er zijn 2 FIN en 2 ACK-segmenten nodig voor een TCP-verbinding te stoppen op normale manier

4.3.2) TCP state diagramma

Zie cursus p81

4.3.3 TCP tijdsbeheersing

- Wanneer een TCP-module een segment stuurt, wordt er een timer gestart. Wanneer de ontvangende TCP-module het segment bevestigt heeft alvorens de teller verlopen is,

wordt de teller gestopt. Als die er niet op tijd geraakt wordt het segment opnieuw gestuurt

- ! Tijd te veel → systeem werkt traag
- ! Tijd te weinig → onnodig opnieuw versturen van segmenten

→ Tijd van de teller dynamisch d.m.v algoritme dat snelheid van netwerk analyseert

4.3.4) TCP congestie management

- Als netwerk teveel pakketten verwerkt dan dat het kan afhandelen → congestie
- **Congestievenster en flow-control-venster worden tegelijkertijd bijgehouden**
- Als ontvangende TCP-module zegt dat hij max. 64KB kan bufferen en dus ontvangen, maar congestievenster zegt dat het max. 32KB kan zijn → zal er maar 32KB verzonden worden

4.3.5 Protocolstapel

Afkorting TCP/IP gebruikt voor het geheel van protocollen

Met protocol-suite/protocol-stapel wordt het geheel van protocollen bedoeld

Revision #2

Created 17 June 2021 13:50:47 by Jasper G.

Updated 3 December 2021 22:13:08 by Jasper G.