TCP/IP

D. Leeuw

1 december 2023
v.0.5.0

Over Dit Document

0.1 Leerdoelen

Dit document geeft een korte introductie TCP/IP. TCP/IP is het protocol dat gebruikt wordt op het Internet en de meeste lokale netwerken om computers met elkaar kunnen laten praten.

De leerdoelen zijn:

- kennis van IP-adressen
- kennis van subnet-masks
- kennis van (default) gateway
- kennis van TCP en UDP als transport protocollen
- kennis van het routeren van data over het Internet

0.2 Voorkennis

Voordat je aan deze les begint is basis kennis van ethernet wenselijk:

- Je moet weten wat een MAC-adres is.
- Je moet weten hoe een ethernet frame eruit ziet.

Inhoudsopgave

Hoofdstuk 1

ARPA vs OSI

Voor moderne netwerken wordt bijna altijd het OSI-model gebruikt om duidelijk te maken wat er binnen een protocol-stack gebeurd. Het meest gebruikte netwerk protocol, TCP/IP, is echter veel ouder dan het OSI-model en TCP/IP kent dan ook zijn eigen model waarop het gebaseerd is. Dit model staat bekent als het ARPA-model.

het ARPA-model kent maar 4 lagen, namelijk de Application-layer, Host-to-Host-layer, Internet-layer en de Network-Interface-layer. Met een beetje goede wil kunnen we deze lagen ongeveer overeen laten komen met de lagen uit het OSI-model.

De Network-Interface is alles dat te maken heeft met de fysieke laag en de netwerkkaarten. De Internet laag zorgt voor de adressering van de netwerk (IP) pakketten. De Host-to-Host laag is verantwoordelijk voor het verzenden van de data in de juiste grote en de controle van de data op het juist aankomen. De Applicatie laag is verder verantwoordelijk voor alle andere zaken die te maken hebben met het opmaken en verzenden of ontvangen van de data.

Layer	ARPA	OSI
7	Application	Application
6		Presentation
5		Session
4	Host-to-Host	Transport
3	Internet	Network
2	Network Interface	Data link
1		Physical

Hoofdstuk 2

IP

IP is een afkorting en staat voor Internet Protocol, vandaar dat het wereldwijde netwerk het Internet heet. IP is het protocol dat op OSI-model laag 3 (network-layer) werkt.

De network-layer uit het OSI-model is verantwoordelijk voor het transporteren van data over het netwerk. Waar layer 2 (o.a. ethernet) verantwoordelijk is voor het bezorgen van data op een lokaal netwerk (LAN) is IP verantwoordelijk voor de wereldwijde data bezorging. De data die op deze laag getransporteerd wordt heet een "packet".

We kunnen de werking van IP vergelijken met het post systeem, waar het ook op gemodelleerd is. Vandaar ook de naam packet voor de data die over het netwerk gaat, het is vergelijkbaar met een postpakket. Om een postpakket te versturen heb je twee cruciale zaken nodig:

1. landcode + postcode (zipcode)
2. huisnummer

Alle andere zaken zijn voor de post niet relevant. De post gebruikt de postcode (en landcode) om te zorgen dat het packet in het juiste gebied terecht komt. Is het daar aangekomen dan zorgt de postbode ervoor dat het pakket bij het juiste huisnummer wordt afgeleverd.

Om deze verzending van een packet over het netwerk mogelijk te maken hebben we ook twee cruciale zaken nodig:

1. het IP-adres
2. het subnet-mask

De logica zit hier anders in elkaar dan bij de post, het subnet-mask is niet het huisnummer en het IP-adres is niet het netwerk. We hebben het

subnet-mask en IP-adres samen nodig. Hoe dat werkt dat bespreken we in dit hoofdstuk.

Een IP-adres bestaat uit vier blokjes gescheiden door een punt. Elk blokje bestaat uit 8-bits en daarmee kan je decimaal tellen van 0 tot 255. We noteren een IP-adres als een decimaal adres, waarbij elk blokje van 8-bits zijn eigen decimale getal vormt. Een voorbeeld van een IP-adres is 192.168.1.4. Binair zou dit zijn 11000000.10101000.00000001.00000100.

Ook het subnet-mask is opgebouwd uit vier blokjes gescheiden door een punt. Ook die blokjes zijn 8-bits en kunnen dus een waarde hebben van 0-255, maar ze mogen niet elke waarde bevatten. Voor het subnet-mask is het handig om te denken in binair, het linker deel van het subnet-mask moet uit enen bestaan en het rechterdeel uit nullen. Er mag geen nul tussen de enen staan en ook geen 1 tussen de nullen. Daarmee zijn dus bepaalde getallen niet mogelijk in een subnet-mask. Een voorbeeld van een subnet-mask is 255.255.255.0, ook 255.0.0.0 is een geldig subnet-mask. Als we het subnet-mask 255.0.0.0 omzetten naar binair dan komen we op 11111111.00000000.00000000.00000000. Alle linker bits zijn 1 en alle rechter bits zijn 0, dus is het een geldig subnet-mask. Nemen subnet-mask we bijvoorbeeld 11111111.11101111.00010000.00000000 als subnet-mask dan zien we dat tussen de linker enen een nul staat en dat tussen de rechter nullen een één staat. Beide is niet toegestaan dus 255.239.16.0 is geen geldig subnet-mask.

Het subnet-mask bepaalt welk deel van het IP-adres het netwerk-adres is en welk deel het host-adres. Het netwerk-adres is vergelijkbaar met de landcode + postcode van de post en het host-adres is vergelijkbaar met het huisnummer. Een voorbeeld maakt dit waarschijnlijk het snelst duidelijk. Nemen we IP-adres 192.168.1.4 met subnet-mask 255.255.255.0 dan is het netwerk-adres 192.168.1.0 en het huisnummer 4. Nemen we hetzelfde IP-adres, maar subnet-mask 255.0.0.0 dat is het netwerk-adres 192.0.0.0 en het huisnummer 168.1.4.

Een computer werkt met 1-en en 0-en en gebruikt die dan ook om te berekenen welk deel van het IP-adres het netwerk-adres is en welk deel het host-adres. De computer gebruikt hiervoor de AND-functie. De waarheidstabel van de AND-functie is:

Input 1	Input 2	Output
0	0	0
0	1	0
1	0	0
1	1	1

De computer neemt het IP-adres (192.168.1.4) en doet een AND met de subnet-mask (255.0.0.0). Om te snappen wat er gebeurt moeten we eerst beide getallen omzetten naar binair. Op deze binaire getallen laten we de AND-functie los. De wiskundige notatie voor de AND is $\wedge$. Het sommetje komt er dus zo uit te zien:
11000000.10101000.00000001.00000100 $\wedge$
11111111.00000000.00000000.00000000 $=$
11000000.00000000.00000000.00000000

Het adres van het netwerk is dus 192.0.0.0.

Het is hopelijk opgevallen dat het netwerk-adres altijd uit 4 blokken blijft bestaan (192.168.1.0 of 192.0.0.0). Het netwerk-adres is het laagste adres in het netwerk en is een “gereserveerd” adres. Het mag niet uitgedeeld worden aan een netwerk-interface. Een ander adres dat niet uitgedeeld mag worden is het zogenaamde broadcast-adres. Het broadcast-adres wordt gebruikt om een bericht te sturen aan alle huisnummers in een netwerk. Het broadcast adres is gedefinieerd als het hoogste adres in het netwerk. In ons voorbeeld van het 192.168.1.0 netwerk met subnet-mask 255.255.255.0 is het hoogste netwerk adres 192.168.1.255. Ook dit adres mag dus niet worden uitgedeeld aan een netwerk-interface. Met twee gereserveerde adressen blijven dus de adressen 1-254 over als huisnummers voor ons 192.168.1.0 netwerk.

Met een IP-adres en een subnet-mask hebben we dus een volledige beschrijving van adres op het Internet. We kennen het netwerk-adres (postcode) en we kennen het host-adres (huisnummer). We weten welke andere host-adressen er zijn binnen ons netwerk en wie dus onze mogelijke "buren"zijn. Alle buren zitten in hetzelfde netwerk.

2.1 De envelope

Het segment dat van de transport laag afkomt moet door IP verwerkt worden tot een packet dat het kan doorgeven aan de fysieke laag (OSI-layer 2). In het packet moet natuurlijk duidelijk zijn voor wie (destination IP) en van wie (source IP) het packet komt. Daarnaast zitten er nog wat extra velden in de “header” van een packet. Hieronder staat de header uitgeschreven in een tabel van 32 bits breed met daaronder de betekenis van elk veld. Niet elk veld wordt volledig uitgelegd, omdat dat buiten de scope van dit document zou gaan, het blijft een introductie op de werking van IP.

V	IHL	TOS	Total Length
ID		Flags	Fragment offset
TTL		Protocol	Header checksum
SIP			
DIP			
Options + padding			
Segment			

V Version - Voor IPv4 is dit 4

IHL Internet Header Length - De lengte in 32-bits woorden van header, minimum is 5 (dus 5x32=160 bits (20 bytes))

TOS Type Of Service - Is dit packet bijvoorbeeld latency gevoelig of niet (voice moet zo snel mogelijk doorgezet worden)

Total length Total length of packet - Totale lengte, header + segment, van het packet in bytes. Minimum is 20 bytes. Het veld is 16 bites en dus kunnen er maximaal 65535 bytes in een packet.

ID Identification - buiten de scope van dit document

Flags 3-bits veld - buiten de scope van dit document

Fragment offset 13 bits veld - buiten de scope van dit document

TTL Time To Live - 8 bits veld dat routing loops moet voorkomen. Wordt nader behandeld in het document over routing.

Protocol - Protocol in het segment - Protocal dat aangetroffen wordt in het segment deel.

Header checksum - De checksum van de header - Checksum berekent over de header. Als de checksum niet klopt wordt het packet weggegooid (door router of ontvanger)

SIP Source IP address - IP adres van de verzender

DIP Destination IP address - IP adres van de ontvanger

Options 0 of meer opties - buiten de scope van dit document

Padding 0 of meer padding bits - om te zorgen dat we op 32 bits woord einde uitkomen, zodat het begin van het segment op 0 komt. Dus afhankelijk van de opties.

Segment Segment afkomstig van UDP of TCP (of een ander protocol)

2.2 Address Resolution

Als we binnen ons eigen netwerk kijken, laten we zeggen op het 192.168.1.0 netwerk, hoe sturen we dan een packet van host 4 naar host 18, als we aannemen dat beide op het netwek zijn aangesloten en aan staan? We gaan hier uit van een LAN op basis van ethernet of Wi-Fi.

Een IP-adres is gekoppeld aan een netwerkkaart. We zouden natuurlijk een packet met daarin het destination IP-adres los kunnen laten op het netwerk en hopen dat er een host is die het packet op pakt, maar zo werkt het niet. We hebben te maken met het OSI-model (of ARPA-model) dat zegt dat een packet van laag 3 eerst door laag 2 moet voordat het bij de physieke laag (laag 1) aankomt. Dus IP moet eerst gebruik maken van het ethernet-protocol. Het zijn dan ook de ethernet-adressen die gebruikt worden op een packet bij een netwerkkaart af te leveren. Ons packet wordt dus ingepakt in een ethernet-frame (enveloppe) met daarop het ethernet-adres van de ontvanger. Dus de vraag is nu hoe komen we aan het ethernet-adres van de ontvanger als we alleen zijn IP-adres weten?

Om dit probleem op te lossen is er een protocol met de naam ARP, Address Resolution Protocol, ofwel een protocol om adres problemen op te lossen. Het protocol is heel simpel en doorloopt de volgende stappen:

1. Kennen we het MAC-address van de bestemming, dan zijn we klaar. Zo niet dan volgt stap 2

2. Stuur een broadcast uit op het netwerk opzoek naar het MAC-adres dat behoort bij het IP-adres. We maken dus een frame met daarin eerst het MAC-broadcast adres: FF:FF:FF:FF:FF:FF, dan ons MAC-adres, dan het IP-adres waarnaar we opzoek zijn (192.168.1.18) gevolgd door ons eigen IP-adres (192.168.1.4) en tot slot de vraag "who-has". Dit frame gaat het ethernet netwerk op.

3. Omdat het een ethernet-broadcast is zal elke netwerkkaart op het netwerk het frame oppakken en doorgeven naar de hogere laag (laag 3). De netwerk-laag bekijkt het packet en besluit of zijn IP-adres overeenkomt met het IP-adres in het packet.

4. Alleen als het IP-adres overeenkomt mag de netwerk-laag iets met de data doen, dus alleen dan zal het zien dat het een vraag bevat, namelijk "who-has". Het weet dan dat de verzender opzoek is naar ons MAC-adres. Het zal dus een nieuwe packet in elkaar zetten met daarin het destination-IP-address en zijn eigen IP-adres. Dat packet gaat terug naar de ethernet-laag die het ontvangen MAC-adres heeft

onthouden en dat gebruikt als destination-MAC-address en er zijn eigen MAC-address als source-address op zet.

5. en zo komt er bij ons een frame aan met ons MAC-address als destination, het gezochte MAC-address als source en in het packet de beide al bekend zijnde IP-adressen.

Als we dit bij elk packet dat we willen versturen moeten doen is dat behoorlijk omslachtig. De meeste systemen gebruiken dan ook een cache om tijdelijk MAC-adressen die behoren bij bepaalde IP-adressen op te slaan. Deze cache wordt een ARP-tabel genoemd. Het `arp` commando kan gebruikt worden om informatie uit de ARP-tabel op te vragen.

Hoofdstuk 3

Gateways/Routers

Als je netwerken aan elkaar koppelt heb je apparaten nodig die die koppeling maken. Deze apparaten worden gateways of routers genoemd. Een router verbindt dus verschillende netwerken aan elkaar en routeert data van het ene naar het andere netwerk.

Ook een router of gateway heeft een IP-adres op zijn netwerk-interface. Elke interface van een router is gekoppeld aan een netwerk en heeft een IP-adres (en subnet-mask) die behoort bij dat netwerk. In router in je netwerk hangen kost je op dat netwerk dus een IP-adres dat je niet voor een host kan gebruiken.

Doordat de router een IP-adres en subnet-mask heeft gekregen op een interface weet hij ook welk netwerk er op die poort zit. Van alle lokaal aangesloten netwerken weet hij dus welk netwerk waar zit. Een binnen komend packet kan dus direct gerouteerd worden naar de netwerken die hij kent.

Als er aan één netwerk meerdere routers hangen en deze routers maken verbinding met verschillende netwerken, dan kunnen we elke router gaan vertellen welke router wat weet. Stel je het volgende voor: We hebben een netwerk 192.168.1.0 en daaraan hangt een router met IP-adres 192.168.1.2 en een router met IP-adres 192.168.1.3. Aan de router met het IP-adres 192.168.1.2 hangen ook de netwerken 192.168.20.0 en 192.168.21.0 en aan aan de router met het adres 192.168.1.3 hangen de netwerken 192.168.30.0 en 192.168.31.0. Als er nu een packet binnenkomt bij de router 192.168.1.2 voor het IP-adres 192.168.31.5, dan zou het handig zijn als router 192.168.1.2 weet dat hij het naar 192.168.1.3 moet sturen. Om dit mogelijk te maken gebruiken we routing-tabellen.

Een routing tabel is vrij simpel en zou er op 192.168.1.2 bijvoorbeeld zo uit kunnen zien:

192.168.1.0	255.255.255.0	interface 0
192.168.20.0	255.255.255.0	interface 1
192.168.21.0	255.255.255.0	interface 2
192.168.30.0	255.255.255.0	192.168.1.3
192.168.31.0	255.255.255.0	192.168.1.3

Deze tabel zegt dat al het verkeer dat bij 192.168.1.2 binnenkomt en dat valt binnen de range van 192.168.30.0/2555.255.255.0 doorgestuurd moet worden naar 192.168.1.3.

Hiermee kunnen we alle netwerken aan elkaar knopen en allemaal voorzien van routing-tabellen. Dat is voor een kleine collectie van netwerken prima te doen, maar voor de duizenden netwerken die gekoppeld zijn aan Internet is dat onmogelijk. De routing tabellen zouden veel te groot worden.

3.1 Default gateway

Het is ondoenlijk om elke computer in het netwerk te voorzien van routing tabellen van het hele Internet. Het is makkelijker om één centrale machine in het netwerk op te hangen die de weg naar het Internet weet. Als een laptop dan bijvoorbeeld een packet heeft voor een ander netwerk dan verstuurt het dat naar die centrale machine en die centrale machine lost het op. Zo'n centrale machine heet een default gateway. Dat kan een router zijn, maar ook een firewall of een proxy server. Voor ons is de verdere functie van de machine niet van belang. We hebben een default gateway ofwel een apparaat waar we data naar kunnen sturen als we het niet meer weten.

Een computer die geen default gateway heeft ingesteld kan dus alleen data versturen naar zijn direct aangesloten netwerk. Hij kan niet communiceren met andere netwerken en al helemaal niet met andere machines die aangesloten zijn op het Internet.

Je kan de default gateway vergelijken met de brievenbus van de post. Ik heb geen idee hoe de post ervoor zorgt dat mijn brief in Canada terecht komt. Ik weet wel dat als ik de brief in de oranje brievenbus van de post gooi, dat zij hun best doen om te zorgen dat hij in Canada komt. Zo werkt dat ook met de default gateway. Ik weet niet hoe die het doet, maar hij gaat zijn best doen om te zorgen dan mijn packet bij bijvoorbeeld Google uitkomt.

Wat we kunnen doen voor computer kunnen we natuurlijk ook doen op routers. Een router kent zijn direct aangesloten netwerken, maar als hij het niet meer zou een default gateway voor een router ook wel makkelijk

zijn en tot op zekere hoogte kan dat. Helaas gaat dat bij de complexiteit van het Internet niet meer op.

3.2 Routing Protocollen

De hoeveelheid aangesloten netwerken op het Internet maakt het onmogelijk om handmatig alle netwerken met hun routers bij te houden. Om deze complexiteit op te lossen moeten we het beheren van de routing tabellen over laten aan geautomatiseerde processen. De meeste simpele vorm is dat elke router zijn routing tabel opstuurt naar de hem bekende routers en default gateways. Als elke router en gateway dat doet dan weten uiteindelijk alle routers op de wereld alle netwerken. Dit proces is min of meer zoals RIP werkt, het Routing Information Protocol.

Het gevolg van RIP is dat routers heel veel geheugen moeten hebben om alle informatie op te kunnen slaan. Er zijn in de loop van de tijd slimmere protocolen bedacht zoals OSPF en BGP om het wereldwijde routing probleem op te lossen.

Hoofdstuk 4

Opdrachten

4.1 Speciale IP adressen

Op een IP-netwerk zijn er twee adressen die special zijn en die niet als IP-adres op een netwerkkaart geconfigureert mogen worden. Welke zijn dat en waarvoor dienen ze?

4.2 IP configuratie

Elke computer op een TCP/IP netwerk, heeft minimaal één IP-adres nodig. De verschillende operating systems hebben hun eigen manier om een machine te voorzien van een IP-adres. Er zijn twee methodes om dit te doen. Er is de handmatige manier en de automatische. Om automatisch een IP-adres te verkrijgen is er op het netwerk een DHCP-server nodig. De meeste bestaande netwerken zijn hiervan voorzien, dus als je al een IP-adres hebt dan is dit gekomen door de DHCP-server.

Mocht het IP-adres op je netwerkkaart beginnen met 169.254.x.y dan is dat een teken dat er geen DHCP-server beschikbaar is.

1. Controlleer wat het IP-adres op je systeem is
2. Zoek uit waar in je besturingssysteem de configuratie van het IP-adres moet gebeuren (zoek op Internet)
3. Zoek uit of de configuratie op automatisch of handmatig IP-adres staat
4. Als de configuratie op handmatig staat noteer dan alle gegevens die nu geconfigureerd zijn

5. Als de configuratie op automatisch (DHCP) staat zet deze dan op handmatig

6. Geef je netwerkkaart het IP adres 192.168.42.42 en een subnet-mask van 255.255.255.255

7. Controlleer wat het IP-adres op je systeem is

8. Als je klaar bent zet dan alles terug naar de oorspronkelijke stand

9. Controlleer wat het IP-adres op je systeem is en of dat overeenkomt met wat je bij punt 1 van deze opdracht hebt aangetroffen

10. Wat is het IP adres van de default gateway op je machine? (zoek op Internet hoe je dat uitzoekt)

4.3 ping

Heb je een netwerkkaart configureerd dan is het handig om te weten of je alles goed hebt gedaan. Je zou willen weten of de machine in het juiste netwerk terecht is gekomen en of je andere machines op het netwerk kan bereiken. Uiteindelijk is de communicatie met andere machines het doel van netwerken.

Een belangrijk hulpmiddel bij het testen van IP verbindingen is `ping`. Ping stuurt een packet uit op een interface naar een bepaald adres en als dat adres antwoord (pong) weet je dat de machine beschikbaar is. Het niet terug keren van een packet kan dus meerdere oorzaken hebben:

- Je hebt het IP-adres niet goed geconfigureert op de netwerkkaart
- Er is uberhaupt geen netwerk aanwezig (niet-functioneerdende of niet aanwezige netwerkkabel)
- De machine die je probeert te pingen staat uit
- De machine die je probeert te pingen heeft niet het opgegegeven IP-adres

Om ervaring op te doen met ping nemen we een IP-adres dat “altijd” zou moeten werken, namelijk het IP-adres 8.8.8.8. Een ander adres dat je zou kunnen proberen is het adres van je default gateway.

1. Gebruik: `ping 8.8.8.8` op de commandline om te zien of je iets terug krijgt. Op Linux of Mac OS X kan je CTRL-C gebruiken om de test te onderbreken op `ping -c4 8.8.8.8` gebruiken om maar 4 packetten uit te sturen.

2. Test of je de default gateway kan pingen. Als je niets terug krijgt dan is de default gateway zo ingesteld dat hij geen antwoord mag geven op ping. Op Linux en Mac OS X kan de CTRL-C gebruiken om de ping te stoppen.

3. Tik op de commandline: `ping -h`

4. Neem de help beschrijving door, zodat je een beeld hebt wat je allemaal met ping kan doen. Met ping kan je weinig stuk maken dus test vooral een aantal van de beschreven opties.

4.4 traceroute

Een andere tool die we kunnen gebruiken om een IP-netwerk te testen en die vooral handig is voor het testen of we een Internet verbinding hebben en hoe die verbinding loopt is het traceroute commando. Traceroute doet zoals de naam als zegt het weergeven van de route van een packet. Op een Linux en Mac OS X heet het commando `traceroute` op Windows heet het `tracert`

1. Type op de commandline: `traceroute 8.8.8.8`

2. Het kan zijn dat de output op een regel een of meerdere sterren (*) geeft. Dat is niet erg het geeft aan dat een tussenliggende router geen antwoord wil geven,

3. De hulp van de tool kan je op Windows opvragen met `tracert /?` op Linux en Mac OS X met `traceroute -h`

4. Welke optie kan je gebruiken om geen address resolving te doen?