VRAGENLIJST COM. NETW. ’12 – ‘13

INFO

Auteur: Nikolas Taillieu
Bron: slides van Prof. Piet Demeester, boek ‘Computer Networking: International Version: A Top-Down Approach’ (5de editie), Computer Networking: Study Companion’, vorige antwoordenlijsten

INHOUDSOPGAVE

1 INLEIDING

1.1 BESPREEK DE STRUCTUUR VAN HET INTERNET ALS "NETWORK OF NETWORKS".

Het internet bestaat uit vele onderling verbonden netwerken, allen Internet Service Provider (ISP) genaamd. Elke ISP is een netwerk van packet switches en communicatielinks. Dus, het internet is een "netwerk van netwerken". ISP's zijn ruwweg georganiseerd in een hiërarchie. ISP's onderaan de hiërarchie zijn residentiële ISP's, universiteits-ISP's en bedrijfs-ISP's. ISP's helemaal bovenaan de hiërarchie noemt men tier-1 ISP's, deze zijn allen onderling verbonden aan zeer hoge snelheden. Tier-n ISP's verlenen diensten aan tier-(n+1) ISP's. Elke ISP wordt onafhankelijk beheerd.

1.2 LEG UIT WAT EEN PROTOCOL IS.

Een protocol definieert het formaat en de volgorde van boodschappen die uitgewisseld worden tussen twee of meer communicatie-entiteiten, alsook de acties die ondernomen moeten worden bij het ontvangen en versturen van een boodschap of andere gebeurtenis. Computernetwerken maken veel en uitgebreid gebruik van protocollen.

1.3 GEEF DE VERSCHILLENDE LAGEN VAN HET TCP/IP REFERENTIEMODEL. GEEF BIJ ELKE LAAG EEN VOORBEELD.

1. Application Layer: applicatie-naar-applicatie communicatie. Dit zijn de toepassingen die verschillende netwerkfuncties implementeren. (FTP, HTTP)
2. Transport Layer: deze laag stuurt boodschappen van de applicatielaag op de host naar een applicatielaag op een andere host. (TCP, UDP)
3. Network Layer: stuurt pakketten van host naar host met behulp van het IP-protocol, en verschillende routeringsprotocollen.
4. Link Layer: verstuurt pakketten van de ene node naar de andere. Afhankelijk van het gebruikte protocol worden verschillende services aangeboden (bvb. reliable-delivery). Protocollen: Ethernet, WiFi, Point-to-Point protocol (PPP).
5. Physical Layer: staat in voor het uitwisselen van bits tussen twee nodes. Afhankelijk van het protocol gebruikt in de Link Layer. Als in de Link Layer bijvoorbeeld het Ethernet protocol gebruikt werd, zorgt deze laag voor implementaties voor Fiber, Coax etc.

1.4 BESPREEK DE ALGEMENE WERKING VAN FTP.

FTP is een applicatielaagprotocol, dat op de transportlaag gebruik maakt van twee TCP-connecties:

- control connection (poort 21): voor het inloggen en uitvoeren van commandos
- data connection (poort 20): voor het versturen van de bestanden. Voor elk bestand wordt een nieuwe connectie geopend.

1.5 BESPREEK ALGEMEEN DE EIGENSCHAPPEN VAN TCP EN VAN UDP. LEG UIT EN VERGELIJK.

TCP	UDP
State – connection oriented	Geen state - connectionless
3-way handshakeprotocol	Eenvoudig protocol
2-richtingsverkeer	1-richtingsverkeer
Flow control, congestion control	Geen controles
Aflevering gegarandeerd sequentieel	Geen garanties
vb: HTTP, FTP	vb: games, streaming

1.6 BESPREEK DE ALGEMENE WERKING VAN HET INTERNETPROTOCOL (IP) EN DE TYPISCHE EIGENSCHAPPEN.

Werking: routeert berichten van de ene host naar de andere op basis van een unieke cijfercombinatie, het IP-adres en een routing table in een router

Eigenschappen: connectionless, unidirectional, datagram = dataeenheid, best effort

1.7 BESPREEK HET PRINCIPE VAN ENCAPSULATIE.

De verschillende internetlagen moeten onafhankelijk van elkaar kunnen werken. Elke laag neemt daarom het pakket over van de laag boven zich als payload en hangt er een eigen header aan vooraleer het opnieuw door te geven aan de onderliggende laag.

1.8 LEG HET VERSCHIL UIT TUSSEN EEN HOST EN EEN ROUTER.

Een host is een eindsysteem, het verwerkt de pakketten tot en met de applicatielaag

Een router (packet switch) is een tussenpunt, het verwerkt de pakketten slechts tot en met de netwerklaag. Het ontvangt de pakketten eerst volledig en stuurt ze dan pas door (=store and forward).

1.9 LEG UIT : CLIENT EN SERVER LAAG. GEEF EEN VOORBEELD. VERGELIJK MET CLIENT/SERVER BIJ APPLICATIES.

Een laag die afhankelijk is van een onderliggende laag is een clientlaag, een laag die services voorziet aan een bovenliggende laag is een serverlaag. *Voorbeeld: de transportlaag is serverlaag voor applicatie-laag, ze biedt namelijk een service aan, bijvoorbeeld TCP voor betrouwbare gegevensoverdracht.*

Bij applicaties biedt de server eveneens diensten aan, aan haar clientapplicaties. Bij gedistribueerde toepassingen worden client- en server elk op een computer uitgevoerd

1.10 BESPREEK IDENTIFICATIE IN DE APPLICATIE-, TRANSPORT- EN NETWERKLAAG.

- Applicatielaag: poortnummer (20 en 21 voor FTP)
- Transportlaag: protocolnummer (6 = TCP, 17 = UDP)
- Netwerklaag: IP-adres (bv. 74.125.136.100)

1.11 HOE NOEMT MEN INFORMATIEBLOKKEN IN DE APPLICATIE-, TRANSPORT-, NETWERK- EN DATALINKLAAG?

- Applicatielaag: message
- Transportlaag: segment
- Netwerklaag: datagram
- Datalinklaag: frame

1.12 WAT IS : IETF, RFC, ISP ? GEEF KORT UITLEG.

IETF: Internet Engineering Task Force, groep die internetstandaarden ontwikkelt, test en implementeert
RFC: Request For Comment, de documenten waarin de IETF standaarden worden gepubliceerd. Ze beschrijven onder andere protocollen zoals TCP, IP, HTTP en SMTP (er bestaan meer dan 3000 RFC's).
ISP: Internet Service Provider, iemand die zijn klanten van internet voorziet

2 APPLICATIELAAG

2.1 BESPREEK HET CLIENT-SERVER PRINCIPE, OP APPLICATIENIVEAU.

Een client vraagt een dienst door één of meerdere boodschappen te sturen naar het serverproces. De client doet dit actief en zet hiervoor een verbinding op met de server (active open) via een willekeurige poort. De servers luisteren passief (passive open) op een vast IP-adres en poort (bv. 80 voor HTTP) en verlenen een dienst door de clientrequest te lezen, een actie uit te voeren en één of meerdere antwoorden terug te sturen. De berichten worden verstuurd naar en vanuit zogenaamde sockets.

2.2 BESPREEK HET CONCEPT VAN "THREADS". GEEF EEN VOORBEELD.

Het doel van threads is om meerdere remote clients toe te laten voor hetzelfde serverprogramma. Het doet dit door elke request van een client een nieuwe thread te creëren.

Voorbeeld: een client (webbrowser) verstuurt een verzoek voor een pagina. De server ontvangt dit op poort 80 en maakt een nieuwe thread aan om dit verzoek te voldoen. In de tussentijd kunnen er nog andere requests van andere clients toekomen op de server. Deze krijgen dan elk een (nieuwe) thread toegewezen. Als een thread klaar is, verstuurt deze dan het antwoord naar de desbetreffende client die op het antwoord wacht: zo kan een server meerdere clients tegelijk bedienen.

2.3 WELKE TRANSPORTDIENSTEN KAN EEN APPLICATIE VEREISEN? GEEF ENKELE VOORBEELDEN.

Een applicatie kan bescherming tegen dataverlies eisen, een al dan niet een verzekerde bandbreedte (vs. elastisch) en/of kan tijdsgevoelig zijn.

Voorbeelden: webradio kan omgaan met dataverlies, maar heeft een zekere bandbreedte nodig en is tijdsgevoelig, bij filesharing daarentegen eist men betrouwbaarheid, maar mag de bandbreedte elastisch zijn

2.4 BESPREEK HET HTTP PROTOCOL EN DE BELANGRIJKSTE PROTOCOLBOODSCHAPPEN.

HTTP: HyperText Transfer Protocol, maakt gebruik van TCP om haar boodschappen te versturen: de client stuurt een HTTP request naar de server, die antwoordt met een HTTP response. HTTP is stateless.

HTTP messages bestaan uit een request/status lijn, meerdere header lijnen, een lege lijn en dan de body. Er zijn twee soorten messages:

- request messages
  request/status line: [method][url][version]
  beschikbare methods:
  - GET: URL opvragen
  - POST: invullen van een form, de body bevat nieuwe informative
  - HEAD: GET, maar enkel de headers worden teruggestuurd
  - PUT: uploaden van een object
  - DELETE: verwijderen een object
- response messages
  request/status line: [version][statuscode][phrase]
  voorbeelden status codes / phrases:
  - 200 OK
  - 404 NOT FOUND

2.5 WAARVOOR STAAT : HTTP, URL, HTML? GEEF KORT UITLEG.

- HTTP: HyperText Transfer Protocol (zie 2.4)
- HTML: HyperText Markup Language: opmaaktaal voor webpagina's
- URL: Uniform Resource Locator: adres van document/object die beschikbaar is op het internet, bestaat uit een hostname en een pad (bv. ugent.be/ex, hostname: ugent.be, pad: /ex)

2.6 BESPREEK DE VERSCHILLENDE HTTP CONNECTIEMOGELIJKHEDEN.

- Non-persistent (default in HTTP 1.0): elk opgevraagd object gebruikt nieuwe TCP-verbinding
- Persistent (default in HTTP 1.1): de server laat na het verzenden van het object de TCP-connectie open, zodat deze later hergebruikt kan worden en sluit ze bij inactiviteit. Pipelining laat toe dat meerdere objecten verstuurd worden zonder te wachten op een response.

2.7 BESPREEK EEN EENVOUDIG MODEL VOOR RESPONSTIJD BIJ HTTP.

- Non-persistent: $\mathrm{RTT}_{\mathrm{TCP}} + \mathrm{RTT}_{\mathrm{request/response}} + \text{file transmission time}$
- Persistent:
  - zonder pipelining: $\mathrm{RTT}_{\mathrm{TCP}} + (\mathrm{RTT}_{\mathrm{request/response}} + \text{file transmission time}) \times \#\text{obj}$
  - pipelining: $\mathrm{RTT}_{\mathrm{TCP}} + \mathrm{RTT}_{\mathrm{request/response}} + (\text{file transmission time} \times \#\text{obj})$

2.8 BESPREEK HET PRINCIPE VAN COOKIES.

Cookies laten toe met state te werken, ondanks het stateless HTTP-protocol. Hun doel is informatie van de gebruiker bij te houden, zodat de gebruiker een 'sessie' kan hebben op een website. Zo kan op een e-commerce website bijvoorbeeld boodschappen onthouden worden in een winkelwagentje.

HTTP-pakketten bevatten een headerregel voor de cookie-informatie die de client bij elke request meestuurt en die server ook in elk antwoord zal opnemen. Aan de hand van de informatie in de cookie kan een server een bezoeker 'herkennen' en de bijhorende informatie opzoeken in een databank.

2.9 BESPREEK CONDITIONAL GET EN WAARVOOR IS DAT NUTTIG?

Bij een conditional GET wordt een HTTP request gestuurd die een GET-methode oproept en een if-modified-since methode heeft in zijn headers. Hierdoor worden enkel objecten teruggegeven als ze gewijzigd zijn sinds de gespecifieerde datum. Hierdoor wordt bandbreedte uitgespaard.

2.10 GEEF EEN OVERZICHT V/D VERSCHILLENDE E-MAILPROTOCOLS (AFKORTING+KORTE UITLEG WAT HET DOET).

- SMTP, Simple Mail Transfer Protocol: transfereert e-mailberichten van de mailserver van de verzender naar de mailserver van de ontvanger. SMTP is een push protocol.
- POP, Post Office Protocol: mail access protocol, werkt in drie fasen:
  - autorisatie: controle van username en password
  - transactie: afhalen van berichten
  - update: gedane bewerkingen worden opgeslagen
- IMAP, Internet Message Access Protocol: mail access protocol, werkt met mappenstructuur op de server en er is de mogelijkheid afzonderlijke componenten van berichten op te halen
- RFC822: message format, eenvoudige plain tekst messages
- MIME, Multipurpose Internet Mail Extensions: message format, oplossing voor non-plain text messages, zoals afbeeldingen en multimedia

2.11 BESPREEK HET GEBRUIK VAN NAAM EN ADRES. GEEF EEN VOORBEELD.

Een host kan aangeduid worden door zijn IP-adres of de daaraan gelinkte hostnaam.

- IP-adres
    - 4 bytes, omgezet in getallen (bv. 74.125.136.100)
    - vaste lengte, hiërarchisch, makkelijk te interpreten in netwerkcontext
- Hostnaam
    - leesbare tekens (bv. www.google.be)
    - makkelijk te onthouden, logische structuur, kan organisatie tonen

2.12 BESPREEK DE DNS HIERARCHIE. GEEF EEN VOORBEELD.

Er zijn verschillende soorten DNS-servers (Domain Name Service):

- Local: lokale DNS-server van de ISP
- Root: top-level DNS server, slechts 13 in de wereld, bevat adressen van alle TLD-nameservers
- Authoritative: de originele host voor de gegevens van een bepaald adres
- Intermediate: server die de gegevens van de authoritative server in zijn cache heeft

Voorbeeld: Wanneer een URL wordt opgegeven in een HTTP-request, wordt eerst de lokale DNS server aangesproken. Die bevindt zich het dichtst bij de host. Indien het IP-adres van de gevraagde site niet aanwezig is binnen het netwerk van de ISP, dan zal de lokale DNS server de vertaling meestal niet kunnen doen. Hij zal dan een aanvraag doen bij een root DNS-server. Het kan zijn dat zo'n root DNS-server ook niet het gevraagde IP-adres heeft staan. Een root DNS-server kan echter wel steeds uit de URL opmaken waar de authoritative DNS-server van de gevraagde host gelegen is. Iedere host moet namelijk minstens twee dergelijke DNS servers hebben (in het geval de ene crasht). De authoritative DNS-server is vaak de lokale DNS-server van de host en weet dus steeds antwoord te geven.

2.13BESPREEK HET ITERATIEF EN RECURSIEF MAPPEN IN DNS.

- Iteratief: de DNS-aanvraag komt bij de lokale DNS-server, die de aanvraag afhandelt. Wanneer een door de lokale DNS-server aangesproken DNS-server niet verantwoordelijk is voor de gevraagde domeinnaam, zal hij het IP-adres teruggeven van de volgende nameserver in de ketting, waardoor de lokale DNS-server verder kan zoeken totdat men bij een authorative DNS-server terechtkomt, die het IP-adres teruggeeft voor gevraagde domeinnaam.
- Recursief: de DNS-aanvraag komt bij de lokale DNS-server. Wanneer een door de lokale DNS-server aangesproken server niet verantwoordelijk is voor de gevraagde domeinnaam, zal hij zélf een aanvraag indienen bij de volgende DNS-server in de ketting. Vanaf het moment dat een antwoord mogelijk is, zal het antwoord terugkeren langs de ketting van aangesproken DNS-servers, tot bij de lokale DNS-server, die het antwoord aflevert bij de client.
- Recursief + iteratief: in de praktijk wordt eerst door de lokale DNS-server iteratief bij de root DNS-server het adres van de TLD nameserver opgevraaagd, vanaf daar zal de lokale DNS-server het verzoek recursief afwerken. Dit doet men om de root servers te sparen, die anders nóg meer verkeer te verwerken zouden krijgen

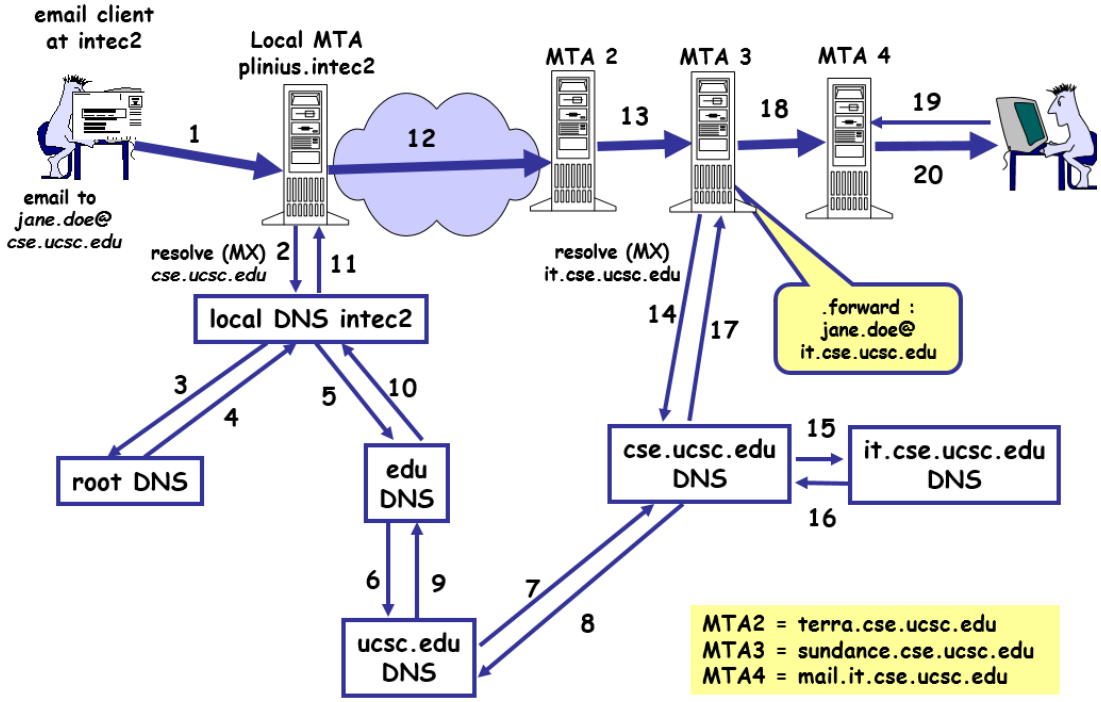
2.14 WAT IS (IN DE CONTEXT VAN DNS) : RR, A, NS, CNAME, MX (GEEF OOK EEN VOORBEELD)

Resource Records (RR) zijn de records in de gedistribueerde DNS-database die de hostname/IP-mapping bevatten. In een DNS-reply zit een RR. Een RR bestaat uit vier velden: (name;value;type;TTL). Dit zijn de verschillende types:

- A: hostname ↔ IP *(hello.foo.com;23.24.25.26;A;?)*
- NS: domain ↔ authoritative DNS-server *(foo.com;dns.foo.com;NS;?)*
- CNAME: alias hostname ↔ canonical hostname *(foo.com;server1.servers.foo.com;CNAME;?)*
- MX: hostname ↔ mailserver *(foo.com;mail.servers.foo.com;MX;?)*

2.15 OP HET EXAMEN WORDT EEN VOORBEELD GEGEVEN (B.V. MIME HEADER, HTML FILE, DNS REQUEST) EN ER WORDT GEVRAAGD DAT TE BESPREKEN.

2.16 BESPREEK HET VOORBEELD DNS + E-MAIL (LAATSTE PARAGRAAF OVER DNS). FIGUUR WORDT GEGEVEN.



1. De email wordt gepushed naar de lokale mailserver (MTA)

2. MTA doet een DNS request (MX) om cse.ucsc.edu te resolven.

3, 4. De lokale DNS server vraagt aan de root server het adres van de .edu TLD DNS

5-10. De lokale DNS server vraagt aan de .edu TLD DNS naar het adres.

11. De lokale DNS server stuurt het gevonden adres naar MTA.

12. MTA pusht de email naar MTA2

13. MTA2 pusht de email naar MTA3

14-17. MTA3 stuurt een DNS query om het adres van it.cse.uscs.edu te kennen

18. De mail wordt gepusht naar MTA4

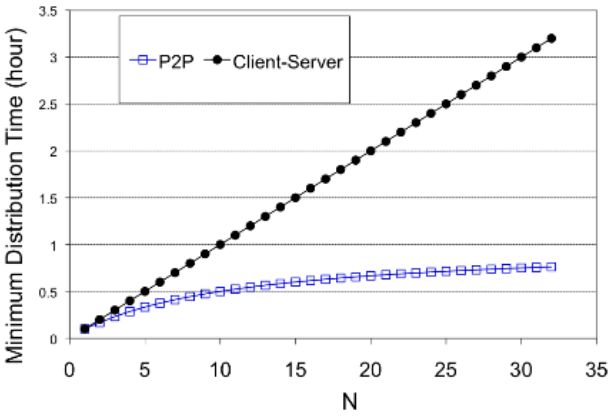
19. De ontvanger vraagt zijn mail op

20. MTA4 stuurt de opgevraagde mail naar de ontvanger.

2.17 LEG DE FIGUUR UIT DIE DE PRESTATIE VAN C/S EN P2P VERGELIJKT (DE FIGUUR WORDT OPGEGEVEN).

De grafiek geeft weer hoe lang het duurt om een bestand van grootte F te verspreiden over N peers. Voor een client/server-applicatie stijgt de functie lineair. Dit komt omdat de peers niet helpen bij het verspreiden van een bestand.

Bij de P2P-applicatie zien we dat het verspreiden van het bestand, ongeacht het aantal peers, minder lang duur dan bij een client/server-applicatie. Als het aantal peers stijgt, vlakt de functie af, dit is omdat de peers ook uploaden naar andere peers.



2.18 BESPREEK HET PRINCIPE VAN DHT.

Een belangrijke component van veel P2P-applicaties is een eenvoudige database, die search en update-operaties ondersteunt. Wanneer deze database gedistribueerd is, kunnen de peers aan content caching doen en queries onder zichzelf uitvoeren. Distributed Hash Tables zijn een techniek die dit toelaat.

DHT werkt als volgt:

- elke peer krijgt een nummer uit $[0, 2^n - 1]$
- de keys komen uit hetzelfde domein, niet-numerieke strings kunnen gehashed worden en als ze te groot zouden worden, nog $mod(2^n)$
- het key-value paar wordt bijgehouden op de peer met het nummer dat gelijk aan, of dichtst bij de (gehashte) waarde van de key ligt
- valt een peer weg (peer churn), dan moeten de opvolgers worden geüpdatet

3 TRANSPORTLAAG

3.1 BESPREEK MULTIPLEXERING (CONNECTION-ORIENTED AND CONNECTIONLESS)

Op een ontvangende host kunnen meerdere applicaties simultaan draaien. Dus, wanneer de transportlaag een segment ontvangt van de netwerklaag, moet het bepalen welke socket de payload van dit segment moet krijgen. Het mechanisme dat de payload aflevert bij de juiste socket wordt demultiplexering genoemd. Bij de verzendende host wordt verzamelen van de data van de verschillende sockets, het toevoegen van de headerdata en het doorgeven van de resulterende segmenten aan de netwerklaag multiplexering genoemd.

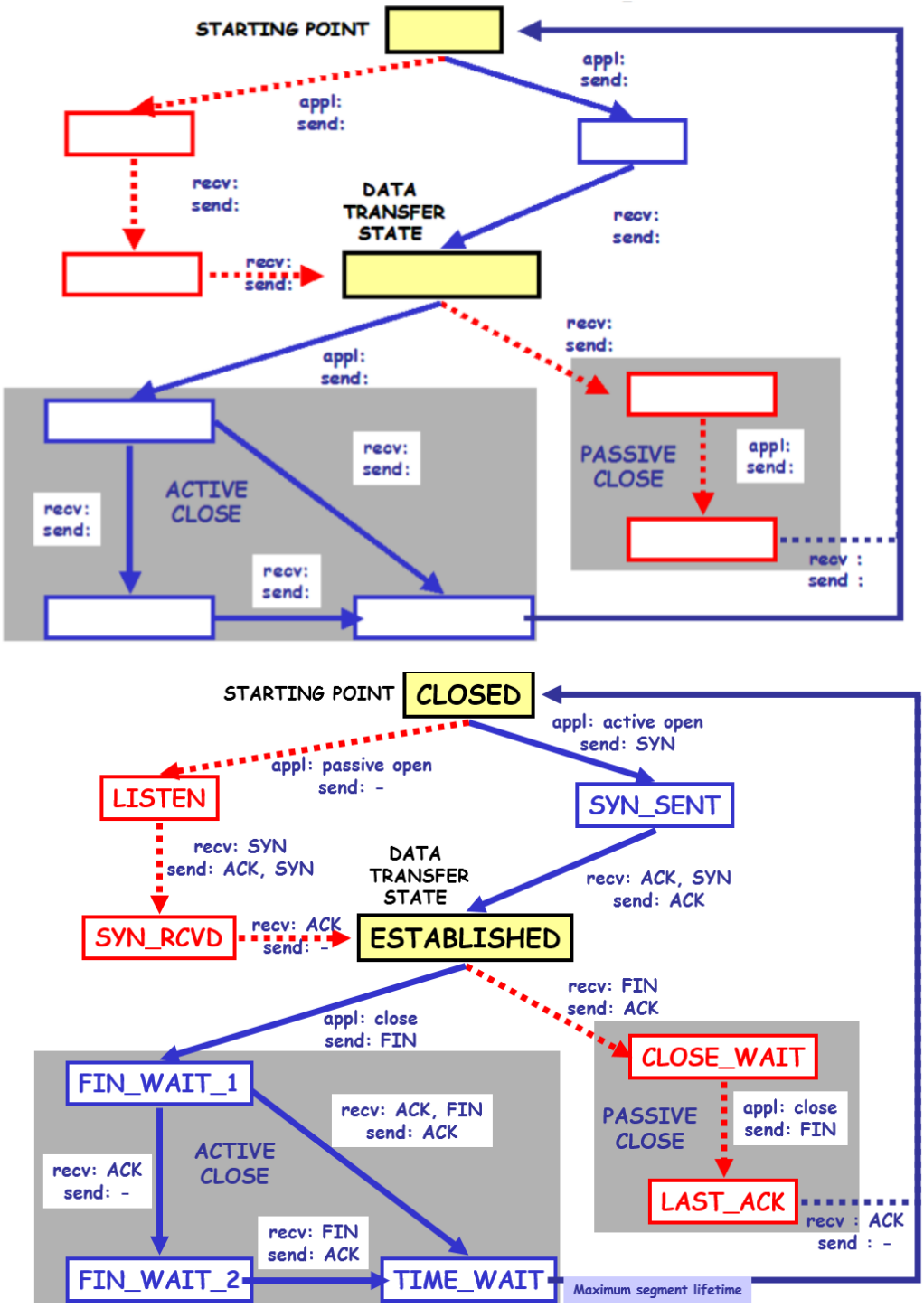
- connection-oriented (TCP): verbinding wordt gedefinieerd door 4-tuple:
  source IP, source port, destination IP, destination port
  → connecties met verschillende oorsprong komen bij verschillende sockets terecht
- connectionless (UDP): verbinding wordt gedefinieerd door 2-tuple:
  destination IP, destination port
  → alle connecties met dezelfde poort komen in dezelfde socket terecht

3.2 BESPREEK DE VERSCHILLENDE VELDEN VAN EEN TCP SEGMENT (SEGMENT WORDT GEGEVEN).

<table>
<tr><td colspan="8">16-bit source port number</td><td colspan="8">16-bit destination port number</td></tr>
<tr><td colspan="16">32-bit sequence number</td></tr>
<tr><td colspan="16">32-bit acknowledgement number</td></tr>
<tr><td colspan="4">4-bit header length</td><td colspan="3">unused (6 bits)</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td colspan="8">16-bit window size</td></tr>
<tr><td colspan="12">16-bit TCP checksum</td><td colspan="4">16-bit urgent pointer</td></tr>
<tr><td colspan="16">Options (if any)</td></tr>
<tr><td colspan="16">Data</td></tr>
</table>

- source en destination port: gebruikt voor (de)multiplexering
- sequence number: gebruikt voor reliable data transfer, bij start wordt een willekeurig getal gekozen waarbij het nummer van de eerste byte in het segment wordt opgeteld
- acknowledgement number: het sequence number van het volgende segment dat men verwacht te ontvangen
- header length: de lengte van de header in 32-bit woorden
- flags
  - RST / SYN / FIN: opzetten en afsluiten van verbinding
  - URG: geeft dringende data aan
  - PSH: data moet rechtstreeks aan bovenliggende laag gegeven worden
- window size: flow control, max. aantal bytes die de zender van dit segment nog kan ontvangen
- TCP checksum: header- en gegevensveld zijn 16-bit integers die 1-complement worden opgeteld, het resultaat wordt in een checksum geplaatst, dit laat foutdetectie toe
- urgent pointer: relatieve positie van de urgente gegevens in het segment
- options: optionele voorkeuren, ev. voor MSS of timestamping
- data: payload

3.3 BESPREEK HET TCP TOESTANDSDIAGRAMMA (EEN SKELETFIGUUR WORDT OPGEGEVEN, ZONDER ENIGE TEKST). MAAK BIJ DE UITLEG EVENEENS GEBRUIK VAN EEN TIJDSVERLOOP (TIJDSAS CLIENT- EN SERVERZIJDE AANGEVEN EN WELKE BOODSCHAPPEN ER UITGEWISSELD WORDEN).



3.4 BESPREEK : ACKNOWLEDGMENT, TIMEOUT RETRANSMIT, DUPLICATE RECEPTION, PIGGYBACKING, DELAYED ACK, ACCUMULATED ACK, SELECTIVE RETRANSMIT, FAST RETRANSMIT, RETRANSMISSION TIMER, RETRANSMISSION TIME-OUT, MEASURED ROUND TRIP TIME.

- acknowledgement: bevestiging dat alle voorgaande bits ontvangen werden, gerealiseerd door een bit in de TCP-header (ACK) die op 1 wordt gezet en een acknowledgement number dat aangeeft wat het sequence number is van de data die men verwacht te ontvangen
- timeout retransmit: wanneer na een bepaald interval geen ACK wordt ontvangen voor een segment wordt het opnieuw verzonden
- duplicate reception: als data correct ontvangen wordt, maar de ACK ervoor verloren gaat, zal data opnieuw verstuurd worden en dus dubbel ontvangen worden
- piggybacking: een ACK voor een segment wordt meegezonden met andere data
- delayed ack: men wacht met het verzenden van de ACK voor correct ontvangen data, bijvoorbeeld om hem te laten piggybacken of om een accumulated ACK te versturen
- accumulated ack: in de plaats van meerdere ACKs te sturen, wordt één (delayed) ACK gestuurd voor de laatst correct ontvangen bit
- selective retransmit: wanneer in een hele reeks van segmenten, slechts 1 verloren gaat, zal enkel dit segment opnieuw verzonden worden zonder al de opvolgende.
- fast retransmit: wanneer de ontvanger ontdekt dat tussenliggende segment(en) verloren zijn gegaan (adhv volgnummers segmenten), zal hij meteen ACK versturen voor het laatst correct ontvangen segment vóór de fout. Zo kan de verzender meteen het ontbrekende deel opsturen.
- retransmission timer: timer aan verzendzijde voor ontvangen van ACK, wanneer deze afloopt, wordt het segment opnieuw verzonden
- retransmission time-out: $RTO = RTT_{estimated} + 4 * RTT_{deviation}$
- measured round trip time: gemeten round trip time van laatst verzonden segment, dit is de tijd die verstrijkt tussen het moment dat het segment verzonden wordt en het moment dat een bevestiging voor het segment wordt ontvangen

3.5 HOE BEREKENT MEN DE RTO ? EN HOE MEET MEN DE ROUND TRIP TIME M ?

Voor elk verzonden segment wordt de RTT gemeten, met die beperking dat op een gegeven moment slechts één meting kan gebeuren. Dit is de tijd die verstrijkt tussen het moment dat het segment verzonden wordt en het moment dat een bevestiging voor het segment wordt ontvangen.

Het resultaat hiervan is RTT_{sample}. TCP houdt een gemiddelde van deze gemeten waarden bij aan de hand van de formule (voor α is 0.125 een goede waarde)

$$RTT_{estimated}(EWMA) = (1 - \alpha) * RTT_{estimated} + \alpha * RTT_{sample}$$

De gemiddelde afwijking hierop wordt dan gegeven door

$$RTT_{deviation} = (1 - \beta) * RTT_{deviation} + \beta * \left|RTT_{sample} - RTT_{estimated}\right|$$

De RTO wordt dan berekend door:

$$RTO = RTT_{estimated} + 4 * RTT_{dev}$$

3.6 BESPREEK HET PRINCIPE VAN FLOW CONTROL IN TCP. LEG IN DETAIL UIT A.D.H.V. DIVERSE "WINDOWS". WAAROM WORDT FLOW CONTROLE GEBRUIKT ?

Omdat de receive buffer van een connectie slechts een beperkte hoeveelheid data kan bijhouden, is er het gevaar dat een buffer overloopt als er meer data de buffer binnenkomt dan eruit gelezen wordt. TCP gebruikt flow control om dit te vermijden. Bij flow control vertelt de ontvanger de verzender hoeveel vrije ruimte hij heeft in zijn buffer. De zender zal dan de hoeveelheid data die hij in de pijplijn lager houden dan de vrije ruimte die de ontvanger nog heeft. Flow control doet aan speed matching: het houdt de verzendsnelheid van de verzender en de leessnelheid van de ontvanger gelijk.

3.7 BESPREEK HET PRINCIPE VAN CONGESTION CONTROL IN TCP. WAAROM WORDT DIT GEBRUIKT ?

Congestion (opstopping) heeft een hoge kost. Er ontstaan lange wachttijden als de pakketsnelheid de maximale linkcapaciteit benadert, o.a. door onnodige retransmissies (retransmission timer loopt af)

De netwerklaag zelf doet niets om congestion tegen te gaan, daarom gebruikt TCP end-to-end congestion control en rekent het dus niet op de netwerklaag om dit voor haar te doen. De hoeveelheid data die een TCP-verbinding in de pijplijn kan brengen, wordt beperkt door het congestion window van de ontvanger. Het congestion window is dynamisch: TCP vermindert de grootte van het congestion window wanneer een pakket verloren gaat (time-out / triple duplicate ACK). Wanneer geen pakketverlies optreedt, wordt het congestion window vergroot.

De exacte regels voor congestion control worden bepaald door drie mechanismes: Additive Increase, Multiplicative Decrease (AIMD), slow start/congestion avoidance en fast retransmit/fast recovery.

3.8 BESPREEK HET VERBAND TUSSEN : SEND WINDOW, RECEIVER WINDOW, CONGESTION WINDOW

- send window: min(receive window, congestion window) – sentButNotAckedBytes
  dit zijn het aantal MSS- segmenten die verzonden mogen worden opdat de ontvanger zijn buffer en ook de buffers van de routers (net) niet zouden overlopen
- receive window: aantal MSS-segmenten die nog ontvangen kunnen worden alvorens de buffer van de ontvanger overloopt
- congestion window: aantal MSS-segmenten die nog verzonden kunnen worden, zonder dat de buffers van het netwerk (routers, links, …) vollopen

Het send window wordt dus beïnvloed door de andere 2: geen van beide mag overflowen

3.9 HOE WORDT CONGESTION GEDETECTEERD EN WAT IS DE REACTIE (GEEN DETAILS)?

Congestion wordt gedetecteerd bij een time-out of triple duplicate ACK. Het congestion window halveert dan in grootte. Reactie:

- time-out: een ACK is verloren gegaan, de ACK komt te laat of data niet aangekomen (dit heeft op zich niets te maken met congestion, maar wordt wel als dusdanig herkend)
  ⇒ slow start + congestion avoidance
- triple duplicate ACKs: segment is verloren gegaan, maar de volgende segmenten zijn wel goed toegekomen, door fast retransmission worden meerdere dezelfde ACK's ontvangen
  ⇒ fast retransmission + fast recovery

3.10 BESPREEK HET PRINCIPE VAN SLOW START EN CONGESTION AVOIDANCE.

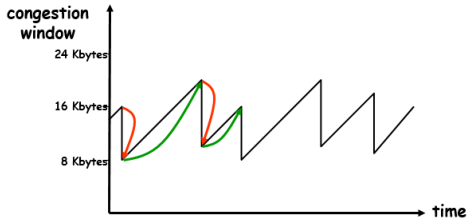
- slow start: wanneer TCP-connectie start, is cwnd 1 MSS groot, bij slow start zal bij deze waarde 1 MSS opgeteld worden voor elke ontvangen ACK. De cwnd zal dus voor elke RTT verdubbelen. Dit gaat verder tot
    - slow start treshold bereikt: congestion avoidance
    - time-out: sstresh = cwnd/2 en cwnd = 1, waarna bijgevolg opnieuw slow start
    - triple duplicate ACK: sstresh = cwnd/2 en cwnd = cwnd/2 , waarna bijgevolg opnieuw congestion avoidance
- congestion avoidance: per segment dat doorgezonden wordt verandert de cwnd als volgt per ACK: cwnd = (cwnd + 1 / cwnd)

3.11 BESPREEK HET PRINCIPE VAN FAST RETRANSMIT EN FAST RECOVERY.

- fast retransmit: bij triple duplicate ACK gaat men ervanuit dat het segment verloren is gegaan en het segment wordt opnieuw verstuurd, zonder te wachten op het aflopen van de timer
- fast recovery: bij triple duplicate ACK wordt de congestion window verhoogd met 1MSS per ontvangen ACK (min. 3). De slow start wordt geannuleerd, vandaar 'fast recovery'.

3.12 LEG UIT : AIMD.

Additive Increase, Multiplicative Decrease: de cwnd wordt lineair verhoogd (telkens +1 MSS per RTT zolang geen loss events optreden) maar in het geval van een loss event wordt de cwnd gehalveerd. Dit zorgt voor 'saw tooth behaviour' als men het in een grafiek zou voorstellen.



Long-lived TCP connection

3.13 LEG UIT HOE TCP "FAIRNESS" ONDERSTEUNT. GEEF EEN VOORBEELD HOE MEN DAT KAN OMZEILEN.

Fairness: elke connectie krijgt een even groot deel van de beschikbare bandbreedte. TCP is fair omdat er telkens een loss event (triple duplicate ACK, time-out) optreedt, het verschil in bitrate tussen twee TCP-sessies wordt gehalveerd. Uiteindelijk zullen de connecties convergeren naar een mooie verdeling.

Dit kan omzeild worden door meerdere parallelle connecties te starten vanuit één proces, waardoor dit proces in totaal een groter deel zal krijgen.

4 NETWERKLAAG

4.1 BESPREEK VERSCHILLENDE IPV4-ADRESKLASSEN. GEEF ENKELE SPECIALE ADRESSEN.

- klasse A:
  - begint met 0 ⇒ 0-127.0.0.0/8
  - 2^7netwerken met 2^{24} interfaces
- klasse B
  - begint met 10 ⇒ 128-191.0.0.0/16
  - 2^{14} netwerken met 2^{16} interfaces
- klasse C
  - begint met 110 ⇒ 192-223.0.0.0/24
  - 2^{21} netwerken met 2^8 interfaces
- klasse D
  - begint met 1110
  - gebruikt voor multicast
- klasse E
  - begint met 1111
  - gereserveerd voor toekomstig gebruik
- speciale adressen:
  - private netwerken: 10.0.0.0/8, 192.168.0.0/16, …
  - loopback: 127.X.Y.Z
  - broadcast op eigen netwerk: 255.255.255.255
  - broadcast op ander netwerk: X.Y.255.255

4.2 BESPREEK HET PRINCIPE VAN DIRECT CONNECTED NETWORKS EN SUBNETWORKS.

- direct connected: netwerk waarbij de hosts rechtstreeks met elkaar verbonden zijn, dus zonder tussenliggende routers. Ook de verbinding tussen twee routers wordt aanzien als een direct connected network.
- subnetwork: direct connected network wordt gerealiseerd onder een overkoepelend netwerk. Iedere interface in een subnetwerk heeft dezelfde subnetmask. Het subnetwerk IP-adres bestaat uit een netwerkgedeelte, een subnetwerkgedeelte en een hostgedeelte.

4.3 BESPREEK SUBNET ADRESSERING. GEEF EEN VOORBEELD.

Een IP-adres in een subnet bestaat uit 3 delen: het netwerkgedeelte, het subnetwerkgedeelte en het hostgedeelte. Bijvoorbeeld 157.193.103.12

- 157.193: netwerkgedeelte (bv. UGent)
- 103: subnetwerk (bv. Plateau)
- 12: host (specifieke PC)

Het subnetwerkgedeelte wordt aangeduid door het hostgedeelte met nullen te voorzien. In dit geval is de mask dus FF.FF.FF.00

4.4 BESPREEK CIDR. GEEF EEN VOORBEELD.

Classless Interdomain Routing: adressen worden opgedeeld in twee delen: een netwerkgedeelte en een hostgedeelte. Dit wordt voorgesteld als a.b.c.d/x, waarbij x de lengte van het netwerkgedeelte in bits is. Het netwerkgedeelte heeft een arbitraire lengte. Voor CIDR er was, gebruikte men ‘Classful

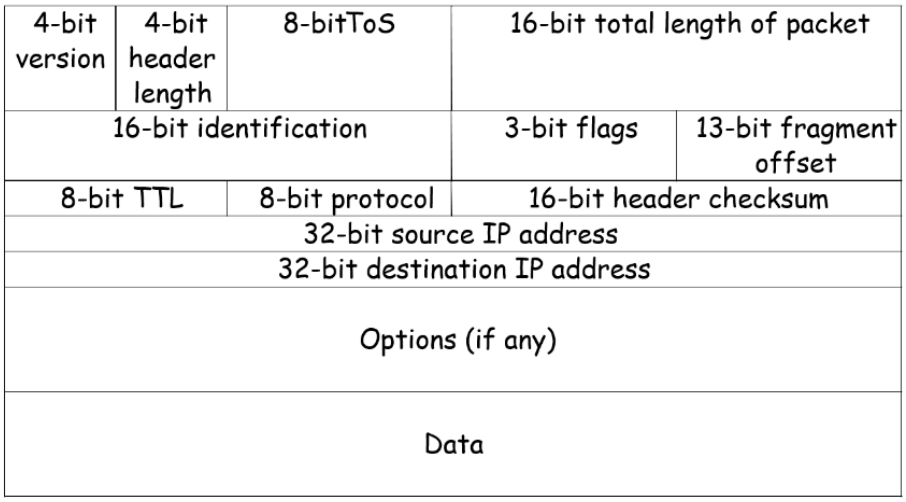
addresering', wat resulteerde in een opdeling van de adressen in klassen (A-E). Dit zorgde voor een ongelijke verdeling van de internetadressen.

Voorbeeld: bedrijf heeft 2000 hosts, dan is dus een klasse B-netwerk vereist. Het bedrijft verkrijgt ruimte voor 65364 adressen, waardoor er dus ruwweg 63.000 onbenut blijven. Met CIDR krijgt het bedrijf een a.b.c.d/21-netwerk toegewezen, waardoor hij plaats krijgt voor 2^{11}-2 = 2046 interfaces.

4.5 BESPREEK HET VERSCHIL TUSSEN ROUTING EN FORWARDING.

- routing: het proces op de netwerklaag waarbij het end-to-end pad van een pakket doorheen het netwerk bepaald wordt.
- forwarding: het proces binnen een router waarmee bepaald wordt naar welke uitgaande poort een binnenkomend pakket doorgestuurd wordt. Daarvoor wordt in een forwarding table het bestemmings IP-adres van het datagram opgezocht aan de hand van longest prefix matching. Deze tabel bevat voor dat adres de poort waar het datagram naartoe moet worden gestuurd.

4.6 BESPREEK DE VERSCHILLENDE VELDEN VAN EEN IP-DATAGRAM (DATAGRAM WORDT GEGEVEN).



- version: versie van datagramindeling (IPv4 of IPv6, hier IPv4)
- header length: grootte van de header, normaal is dit 20 bytes, maar kan variëren
- type of service: 3 bits voor serviceniveau, 4 bits voor delay/bandwith/…, 1 ongebruikte bit
- total length: grootte van volledig IP-datagram in bytes (incl. header)
- identfication, flags en offset: afhandelen van fragmentatie
- time to live: start typisch met 32 of 64 en wordt elke hop gedecrementeerd, doel is voorkomen dat datagram in een lus terechtkomt en steeds verzonden blijft, wanneer de TTL 0 wordt, wordt het pakket gedropt
- protocol: bepaalt het transportlaagprotocol (6 --> TCP, 17 --> UDP, 1--> ICMP)
- header checksum: foutdetectie, hiervoor wordt de header opgedeeld in 16-bit getallen, die bitsgewijs worden opgeteld en vergeleken met de checksum. Gebeurt zowel op de transportlaag als op de netwerklaag omdat de transport er niet op kan vertrouwen dat de netwerklaag dit doet
- source / destination IP-adress: 32 bit-adressen
- options: zelden gebruikt, maar kan gebruikt worden voor timestamps, route-info, etc.
- data: payload

4.7 BESPREEK FRAGMENTATIE.

De MTU, maximum transfer unit, bepaalt de maximale hoeveelheid gegevens die een datalink frame kan bevatten. Wanneer het datagram te groot is, wordt het opgesplitst in fragmenten.

Hiervoor worden de identification, flags en fragmentation offset flags in de datagramheader gebruikt. Gefragmenteerde pakketten hebben altijd dezelfde identification, maar hebben elk een verschillende offset. Het laatste pakket uit de reeks zal een fragmentation flag van 0 hebben.

Gefragmenteerde pakketten worden omwille van performantieredenen niet terug samengesteld op de routers, maar enkel op de eindsystemen.

4.8 WAT IS ICMP ? GEEF EEN VOORBEELD BIJ HET GEBRUIK IN EEN REDIRECT EN TRACEROUTE.

Internet Control Message Protocol, het wordt door hosts, routers en gateways gebruikt om netwerkinformatie met elkaar uit te wisselen. ICMP-berichten worden getransporteerd in een datagram met protocolnummer 1.

Een ICMP-bericht bestaat uit een type-, een code-, een checksum- en een datasectie. Bij foutmeldingen zal de datasectie de datagramheader en de eerste 8 bytes van het pakket bevatten die de fout veroorzaakte. Bv. type 3, code 1: destination host unreachable.

Voorbeelden:

- *redirect: wanneer host een bericht via R1 naar R2 verstuurt, terwijl hij ook rechtstreeks verbonden is met R2, zal R1 de host hiervan op de hoogte stellen met een 'redirect'-bericht, waarop de host zijn routing tables zal aanpassen.*
- *traceroute: het doel van een traceroute is het bepalen van het pad dat een datagrampakket langsheen het netwerk heeft gemaakt. Dat gebeurt door eerst een pakket te zenden met TTL=1, waarop de eerste hop een ICMP Time Exceeded zal terugzenden met informatie over zichzelf. Daarna wordt hetzelfde gedaan met TTL=2 enzoverder tot het pakket aankomt bij de eindhost en geen ICMP-foutmessage meer wordt gestuurd.*

4.9 BESPREEK NAT. GEEF EEN VOORBEELD. WAT IS LARGE SCALE NAT ?

Network Address Translation, dit is een systeem waarbij een reeks hosts waarvoor slechts één publiek IP beschikbaar is, toch op het internet kunnen surfen. Hiervoor moet elk van de hosts verbonden worden met een switch, die op zich verbonden is met een NAT-router.

Deze router gedraagt zich dan tegenover de buitenwereld als een device met slechts één IP-adres. De router maakt gebruik van een NAT Translation Table om een mapping te maken van poorten op de router, naar IP-adressen en poorten binnen het privénetwerk.

Voorbeeld: host in netwerk vraagt een webpagina (IP-adres + poort waar pagina ontvangen zal worden), deze aanvraag komt via de switch toe bij de NAT-router. Deze maakt een nieuwe poort vrij om de aanvraag te doen, koppelt deze aan de door de host opgegeven poort en zijn lokaal IP-adres en doet dan de aanvraag aan de webserver met het publiek IP-adres. De webserver verzendt naar de NAT-router, op zijn poort en het publieke IP-adres. De NAT-router ontvangt de webpagina, kijkt welke host verbonden is met die poort en zendt dan de webpagina door naar lokaal IP-adres en poort van host.

Het systeem is bedoeld om het tekort aan IPv4-adressen op te vangen, een probleem dat in de toekomst in principe opvangen zal worden door IPv6. Het heeft als voordeel dat de lokale IP-adressen mogen

veranderen, zonder dat de buitenwereld daar iets van ondervindt. Het betekent eveneens dat het veranderen van provider geen invloed heeft op het thuisnetwerk en ook dat de individuele hosts niet meer rechtstreeks te adresseren zijn van buitenuit, wat een betere beveiliging toelaat.

De nadelen van het systeem zijn dat poortnummers in principe niet bedoeld zijn om hosts te adresseren, maar wel processen. Daarnaast horen routers enkel een implementatie te voorzien tot op de netwerklaag, terwijl hiervoor ook de transportlaag (poortnummers!) gebruikt wordt.

Een large scale NAT is hetzelfde principe, maar dan toegepast door een ISP op zijn klanten.

4.10 BESPREEK DHCP. GEEF EEN VOORBEELD.

Dynamic Host Configuration Protocol, protocol om IP-adressen dynamisch toe te wijzen aan hosts. Dit gebeurt door een DHCP-server, die al dan niet ingebouwd is in een router. Dit heeft als voordelen een plug and play-connectiviteit voor de gebruikers, IP-adressen die hergebruikt kunnen worden wanneer ze niet langer gebruikt worden en de lease die vernieuwd kan worden voor een IP-adres.

Voorbeeld:

- *Host zendt een DHCP-discoverbericht uit via UDP*
  - *source=0.0.0.0, destination=255.255.255.255, port=67, ID=x*
- *Iedereen in netwerk ontvangt de oproep, de DHCP-servers reageren met een DHCP-offerberichten*
  - *source=server-IP, destination=255.255.255.255, port=68, ID=x+1, data= aanboden IP-adres + lifetime van het aangeboden adres*
- *Host ontvangt offers en kiest één door een DHCP-request uit te sturen*
  - *source=0.0.0.0, destination=255.255.255.255, port=67, ID=x, data=IP-adres van de uitgekozen server + lifetime van het aangeboden adres*
- *DHCP-server herkent zijn eigen IP-adres en zendt een DHCP-ACK-bericht uit om de procedure te beëindigen*
  - *source=server-IP, destination=255.255.255.255, port=68, ID=x+1, data= aangeboden IP-adres + lifetime van het aangeboden adres*

4.11 WAT IS EEN AS ? GEEF 3 TYPES (WAAROM IS HET BELANGRIJK EEN ONDERSCHEID TE MAKEN).

Autonomous System, het internet bestaat uit verschillende AS's die onderling verbonden zijn. Types:

- stub AS: klein bedrijf, maakt 1 connectie met andere AS's
- multihomed AS: groter bedrijf, meerdere verbindingen met andere AS's
- transit AS: provider, verbindt meerdere AS's

Het is belangrijk een onderscheid te maken omdat elk type AS een ander routeringsalgoritme gebruikt, alle routers binnen een AS gebruiken hetzelfde routeringsalgoritme.

4.12 BESPREEK HET VERSCHIL TUSSEN INTRA- EN INTER-AS ROUTERING.

- intra-AS routering: wordt gebruikt om de routes te bepalen tussen hosts binnen een AS en wordt gekozen door het AS zelf. Twee protocollen worden veel gebruikt: RIP (Routing Information Protocol) en OSPF (Open Shortest Path First). Naar deze protocollen wordt ook soms verwezen als IGRP (Interior Gateway Routing Protocol).
- inter-AS routering: wordt gebruikt om de routes te bepalen tussen hosts in verschillende AS's. Dit is vastgelegd op BGP (Border Gateway Protocol)

4.13 BESPREEK HET PRINCIPE VAN DISTANCE VECTOR EN LINK-STATE ROUTERING. GEEF EEN VOORBEELD VOOR BEIDE STRATEGIEËN.

- Distance Vector Routering: de afstand tot bestemmingen wordt bijgehouden in een distance tabel. De distance vectors worden onderling uitgewisseld door routers, wat het een gedecentraliseerd padbepalingsalgoritme maakt. Als metriek voor afstand wordt de hop count genomen: het aantal subnetten op een pad (max. 15). RIP is eenvoudig en wordt veel gebruikt, maar convergeert traag, is enkel goed voor kleine netwerken en de hop count stijgt tot oneindig bij het wegvallen van een router, vandaar de hop count beperking. *voorbeeld: RIP*
- Link-state routering: elke router beschikt over de volledige topologie van het netwerk, wat het een gecentraliseerd padbepalingsalgoritme maakt. Het kortste pad wordt berekend met behulp van het algoritme van Dijkstra, met het eigen subnet als root. De netwerktopologie wordt verspreid met behulp van periodieke broadcasts en een broadcast als de topologie zou wijzigen. De broadcasts worden link state broadcasts genoemd; *voorbeeld: OSPF*

4.14 BESPREEK HIERARCHICAL OSPF. WAAROM IS DAT NUTTIG ?

Voor hierarchical OSPF wordt een AS opgedeeld in verschillende zones. Elke router stuurt dan de link state broadcasts door naar de routers in zijn eigen zone. Om verbindingen tussen de verschillende zones te verwezenlijken zijn er Area Border routers. Centraal in de AS staat één Backbone Area, die verantwoordelijk is om het verkeer tussen de verschillende zones te routeren. In deze Backbone Area zitten alle Area Border routers.

De verschillende types routers bij hierarchical OSPF zijn

- interne routers: horen bij een area, zorgen voor de padbepaling in die area
- area border routers: horen zowel bij de area als de backbone, zorgen voor de padbepaling van pakketten met een bestemming buiten de area
- backbone routers: horen bij de backbone, zorgen voor padbepaling binnen de backbone
- boundary routers: uitwisseling van padinformatie met andere AS's, gebruiken BGP.

Het nut van hierarchical OSPF is dat het één groot rekenprobleem opsplitst in kleinere problemen, die makkelijker op te lossen zijn.

4.15 BESPREEK EEN VOORBEELD VAN BGP. WAAROM HEEFT MEN I-BGP EN E-BGP ?

Border Gateway Protocol, wordt gebruikt voor inter-AS routering. BGP biedt aan elke AS:

- bereikbaarheidsinformatie over de buren van de AS
- bereikbaarheidsinformatie aan alle routers in het AS
- het kan de goede routes naar subnetten bepalen

Voorbeeld: X meldt aan B en C dat het geen paden kent naar andere bestemmingen. Dus weet B dat het nooit verkeer via X naar Y kan doorsturen.

Voorbeeld 2: B kent pad AW van A. B meldt dit aan X, maar niet aan C. De reden hiervoor is dat C dan het pad CBAW kan gebruiken, terwijl ook CAW beschikbaar is.

I-BGP: intern-BGP, gebruik van BGP om padbepalingsinformatie te verspreiden naar routers binnen AS
E-BGP: extern-BGP, gebruik van BGP tussen routers in verschillende AS's (voorbeelden)

4.16 WAT IS EEN AS-PATH ? WAT IS EEN NEXT-HOP ?

Een AS-PATH is een sequentie van tussenliggende AS's tussen source en destination routers. Samen vormen ze een gerichte route voor pakketten om langs te reizen. Het AS-PATH bevat voor elke AS het AS Number (ASN) waar de advertisement voor deze route is gepasseerd.

NEXT-HOP is een BGP-attribuut dat aangeeft wat het IP-adres is van de volgende hop die gebruikt wordt om een bepaalde bestemming te bereiken.

4.17 WAT IS POLICY BASED ROUTING IN BGP ? GEEF EEN VOORBEELD.

Bij policy based routing worden bepaalde politieke en economische regels gebruikt die ervoor zorgen dat een router bepaalde routers niet zal adverteren, ook al bestaan deze paden.

Voorbeeld: een multihomed network zal bijvoorbeeld geen transitverkeer willen en dus enkel de prefixen van zijn eigen netwerken adverteren.

5 DATALINKLAAG

5.1 BESPREEK DE VERSCHILLENDE VELDEN VAN EEN ETHERNET FRAME (FRAME WORDT GEGEVEN).

- preamble: 7 bytes, laten toe dat de adapter zijn klok synchroniseert met die van de verzender
- frame delimiter: 1 byte, aankondiging van eind van synchronisatie
- destination / source adress: 6 bytes, MAC-adres van de ontvanger/verzender
- type: 2 bytes, duidt bovenliggend netwerkprotocol aan
- data: 0-1500 bytes, payload
- padding: 0-46 bytes, extra nullen om aan de minimale frame length van 64 bytes te komen
- checksum: 4 bytes, foutdetectie aan de hand van CRC

5.2 BESPREEK HET CSMA/CD PRINCIPE.

Carrier Sense Multiple Access / Collision Detection: opdat iets verzonden mag worden moet je vooraf luisteren (carrier sensing), begint er toch iemand tegelijk te sturen op de lijn, stop dan (collision detection)

CSMA luistert of het kanaal vrij is, indien het in gebruik is wordt gewacht tot het kanaal vrij is, samen met een bijkomende 96 extra bits. Indien het niet in gebruik is, wordt het frame verzonden

CD detecteert botsingen. Wanneer geen botsing wordt ontdekt, wordt het frame beschouwd als verzonden. Treedt een botsing op, dan stopt men het verzenden, worden 48 jambits verzonden (=alle verzendende adapters op de hoogte brengen van botsing) en betreedt men de exponentional backoff fase (zie vraag 4).

5.3 WAAROM GEBRUIKT MEN BIJ ETHERNET EEN MIN. FRAMELENGTE VAN 64 BYTES ?

Indien een frame kleiner zou zijn dan 64 bytes, zou het kunnen gebeuren dat een collision optreedt, maar omdat het segment zo klein is, zal het frame geheel verzonden zijn vooraleer het jamsignaal de verzender kan bereiken. Daardoor lijkt het alsof het frame correct werd verzonden, maar in werkelijkheid is de data verloren gegaan.

5.4 WAT IS EXPONENTIAL BACK-OFF (BESPREEK).

Wanneer een collision optreedt, dan wacht de adapter steeds $K \times 512$ bitintervallen om opnieuw te proberen, met $K \in \{0,1,2,\ldots 2^m\text{-}1\}$. m is het aantal keer dat een collision is opgetreden en m heeft een bovengrens van 10. De K wordt willekeurig gekozen uit de aangegeven mogelijke waarden en alle waarden voor K hebben een even grote kans om gekozen te worden.

Men gebruikt systeem omdat een adapter geen idee heeft hoeveel adapters betrokken zijn bij een botsing. Door K steeds (mogelijk) te laten vergroten, kan men zich aanpassen aan meerdere scenario's.

5.5 BESPREEK ARP BIJ ETHERNET.

Address Resolution Protocol: protocol dat zorgt voor een vertaling tussen de netwerklaagadressen (IP-adres) en de linklaagadressen (MAC-adres). Hiervoor heeft elke node zijn eigen ARP-tabel die IP-adressen op MAC-addressen mapt. Elke mapping heeft een TTL die aangeeft wanneer ze vervalt.

Voorbeeld: een node A (10.11.12.13) wil iets verzenden naar node B (10.11.12.14), daarvoor heeft hij het MAC-adres nodig van B. Indien dit MAC-adres reeds aanwezig is in zijn ARP-tabel, dan gebruikt hij dat. Zit het er niet in, dan gebruik A het ARP-protocol om het MAC-adres te bepalen. A doet dit als volgt wanneer B zich binnen hetzelfde LAN-netwerk bevindt:

1. *A construeert ARP-pakket met ARP-query*
2. *A broadcast dit pakket naar het hele netwerk (destination FF.FF.FF.FF.FF.FF)*
3. *B ontvangt dit pakket en herkent daarin zijn eigen IP-adres. Hij antwoordt met een nieuw ARP-pakket waarin de juiste mapping staat*
4. *A ontvangt het pakket van B en slaat de mapping op in zijn ARP-tabel*

Wanneer B zich niet in hetzelfde LAN-netwerk bevindt, zal A dit herkennen in zijn routing tabel en daarom als bestemmingsadres het MAC-adres nemen van de router, die de rest afhandelt.

5.6 WAT IS EEN ETHERNET HUB ? EN EEN ETHERNET SWITCH ?

- Hub: een hub werkt op de individuele binnenkomende bits, de hub werkt dus op de fysieke laag, in plaats van op de linklaag. Hij versterkt ze en stuurt ze door naar alle andere interfaces van de hub. Het is eigenlijk een repeater met wat extra netwerkbeheerfunctionaliteit.

  Het voordeel is de simpliciteit van de hub en het feit dat de maximale afstand uitgebreid wordt. De nadelen zijn dat iedereen elkaar kan horen, waardoor de collision domains versmelten tot één groot collision domain, wat ook betekent dat de bandbreedte gedeeld wordt. Bovendien kunnen 10BaseT en 100BaseT hubs niet in éénzelfde netwerk gebruikt worden.

- Switch: een switch werkt op de binnenkomende frames, de hub verzendt ze verder en filter ze aan de hand van hun MAC-adres: ontvangen frames worden niet doorgezonden naar alle nodes, maar enkel naar de interface die leidt naar de bestemming. Werkt dus op de linklaag.

  Het gebruik van een switch vermijdt de nadelen van een hub: elk LAN-segment behoudt zijn collision domain, zonder dat ze samen smelten. 10/100 kunnen tegelijk gebruikt worden.

5.7 HOE WORDEN DE SWITCHTABELLEN INGEVULD ? EN HOE WORDEN ZE GEBRUIKT ?

Switchtabellen worden gebruikt voor filtering en forwarding. Entries in een switchtabel bevatten een MAC-adres van een node, de interface naar die node en een timestamp.

Switchtabellen worden automatisch gevuld en zijn zelflerend. Het vullen gebeurt als volgt:

1. Switchtabel is leeg
2. Voor elk inkomend frame slaat de switch het MAC-adres van de afzender op, de interface waarop dit frame is binnengekomen, alsook de huidige tijd.
3. Als er lange tijd geen frames binnenkomen van bepaald MAC-adres uit de tabel, dan wordt de entry uit de tabel gehaald

De switchtabellen worden als volgt gebruikt: stel dat een frame met destinationadres D binnenkomt via interface i. Dan zijn er drie mogelijkheden:

- Geen entry voor D in de tabel, de switch forwardt het frame naar alle interfaces, behalve i
- Entry voor D die op i mapt: frame wordt gedropt (=filtering)
- Entry die D associeert met j ($j \neq i$): switch stuurt frame naar j (=forwarding)

5.8 BESPREEK STP EN GEEF EEN VOORBEELD.

Spanning Tree Protocol: dit protocol voorkomt dat een frame eindeloos rondgestuurd wordt door een opspannende boom van het netwerk op te bouwen. De configuratie ervan werkt als volgt:

- Blokkeer alle poorten
- Kies een root switch, aan de hand van het laagste Bridge ID dat bestaat uit 2 bytes bridge priority en 6 bytes MAC-adres
- Maak een maximum spanning tree via Kruskal
- Open de poorten om een netwerkboom te bekomen zoals de opspannende boom

STP is enkel bruikbaar in kleine LAN-omgeving omdat de recoverytijd ervan hoog is (30-60s). Rapid STP is een uitbreiding van STP die een lagere recoverytijd heeft.

5.9 WAT IS EEN VLAN ? BESPREEK TWEE TYPES VLAN.

Virtual Area Network: hosts is een VLAN kunnen met elkaar communiceren alsof zij (en geen enkele andere hosts) verbonden waren met de switch.

- port-based VLAN: de poorten van de switch worden in groepen verdeeld door de netwerkbeheerder. In de praktijk krijgen poorten een VLAN ID toegewezen en kunnen ze enkel communiceren met poorten met hetzelfde VLAN ID. Verkeer tussen verschillende VLAN's gebeurt via een aparte router.
- tagged VLAN: frames krijgen een extra tagheader. In de header zit informatie over de VLAN waar het frame vandaan komt. Switches en hosts zijn 'VLAN-aware' als ze tagged frames ondersteunen. Frames zonder tag zijn zgn. 'untagged frames'. Er zijn twee soorten UF's:
    - VLAN tagged frame: tagged frame met VLAN-identificatie en prioriteitsinformatie in de header
    - Priority-tagged frame: tagged frame met enkel priority-informatie (VLAN-ID = 0)

5.10 GEEF EEN AANTAL VOOR- EN NADELEN VAN SWITCHES (VERSUS ROUTERS).

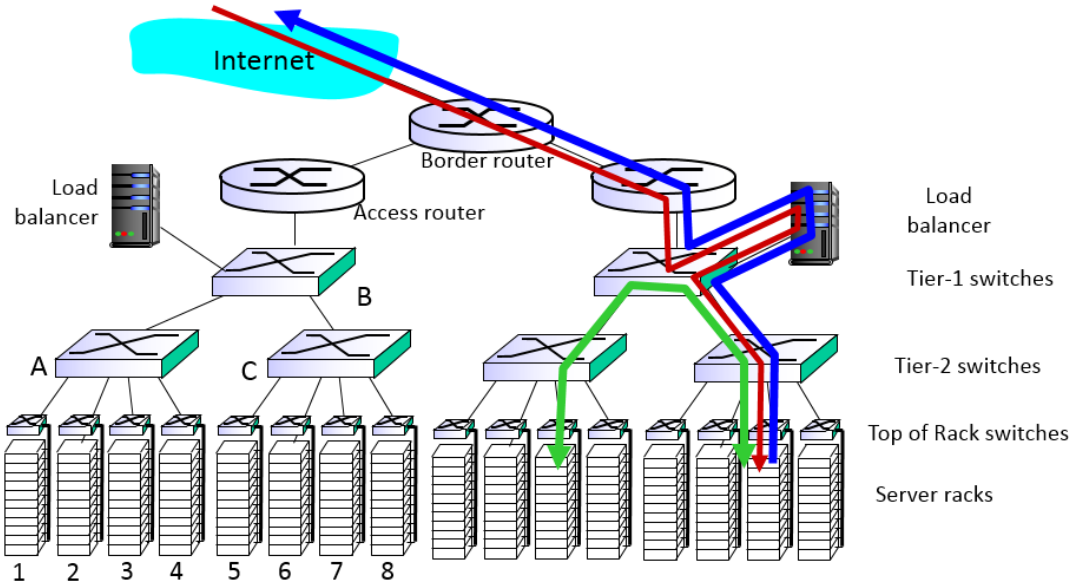
- Switches
  - Voordelen
    - plug & play, de switchtables zijn self-learning
    - simpeler, dus minder packet processing (implementatie tot linklaag)
  - Nadelen
    - het verkeer wordt beperkt tot een spanning tree, ook wanneer alternatieve bandbreedte aanwezig zou zijn
    - geen bescherming tegen broadcast storms
- Routers
  - Voordelen
    - biedt ondersteuning voor meerde netwerktopologieën en om cycling te voorkomen gebruikt men TTL-counters en goede routingprotocollen
    - firewallbescherming tegen linklaag broadcast storms
  - Nadelen
    - heeft IP-configuratie nodig en is dus niet plug & play
    - geavanceerder,dus meer packet processing (implementatie tot netwerklaag)

5.11 BESPREEK DE ARCHITECTUUR EN WERKING VAN EEN "DATA CENTER NETWORK".

De uitdaging voor een data center network is het hosten van de verschillende applicaties, die elk een gigantisch aantal clients bedient. Om dit goed te kunnen doen moet ze de load goed beheren/balanceren en processing-, networking- en data- bottlenecks ontwijken.

Belangrijk voor een data center network is de load-balancer. De load balancer werkt op applicatieniveau. Hij ontvangt het binnenkomende verkeer en verdeelt de workload over de verschillende servers en verbergt daarbij de complexe structuur van het datacenter voor de clients.

In het datacenter is er een sterke interconnectie (tot zelfs fully connected → alle switches en racks met elkaar verbonden) tussen de verschillende switches en racks. Dat zorgt voor een verhoogde throughput tussen de verschillende racks (door de aanwezigheid van meerdere routing paths), alsook voor een verhoogde betrouwbaarheid door redundantie.



HOOFDSTUK 8 : BEVEILIGING

8.1 BESPREEK EEN AANTAL MOGELIJKE AANVALLEN OP HET INTERNET (EN DE BIJHORENDE VERDEDIGINGEN)

- Mapping: via port scans, waarbij men een TCP-connectie probeert op te zetten op verschillende poorten, probeert men te weten te komen welke services actief zijn en welke poorten open staan.
    - Oplossing: netwerkverkeer monitoren en uitkijken naar verdachte activiteit
- Sniffing: pakketten die gebroadcast worden of over een broadcastmedium verstuurd worden, kunnen gelezen worden door iedereen in het netwerk
    - Oplossing: software installeren die de beheerder waarschuwt als een NIC in promiscue mode werkt, belangrijke gegevens versleutelen, broadcastmedia (zoals hubs) vermijden, echter is sniffing ook een belangrijke tool voor de administrators!
- IP-spoofing: het bronadres van een IP-pakket wordt vervalst
    - Oplossing: pakketten filteren die een onmogelijke origine hebben, authentiseren met met IPsec.
- (Distributed) Denial of Service: de werking van een host wordt verstoord of zelfs platgelegd door een grote hoeveelheid pakketten op deze host af te sturen, de host moet namelijk elk van deze pakketjes verwerken. Bij distributed DoS komt de aanval van een groot aantal verschillende hosts (al dan niet met hun medeweten, cq. botnets)
    - Oplossing: deze pakketjes proberen weg te filteren en/of de bron te zoeken van de pakketten en het probleem daar 'op te lossen'.

8.2 BESPREEK HET PRINCIPE VAN CRYPTOGRAFIE MET SYMMETRISCHE SLEUTELS.

Beide gebruiken dezelfde geheime sleutel om hun boodschappen te versleutelen. Beiden moeten deze symmetrische sleutel op voorhand kennen.

Voorbeeld: A encrypteert boodschap m met sleutel K_{A-B} en verstuurt de geëncrypteerde boodschap $K_{A-B}(m)$ naar B. B ontvangt deze boodschap en decrypteert ze: $m = K_{A-B}(K_{A-B}(m))$

8.3 BESPREEK HET VERSCHIL TUSSEN BLOK- EN STROOMVERSLEUTELING. WAT IS CBC ?

- Stroomversleuteling: de stroom wordt bitsgewijs versleuteld
- Blokversleuteling: de volledige boodschap wordt opgedeeld in k-bit (bv. 64-bit) blokken, die elk apart versleuteld worden. Voor grote blokken worden de mappings echter te groot, dus deelt men de k-bit blokken nog eens op in kleinere blokken (bv. 8-bit). Die worden elk apart versleuteld en daarna door elkaar aaneengezet. Dit wordt een aantal keer herhaald.
- CBC: Cipher-Block-Chaining, om te voorkomen dat dezelfde data dezelfde versleutelde data oplevert (wat het breken van de code zou vergemakkelijken), gebruikt men een Random Number Generator om een willekeurige R te berekenen. Dan wordt een blok B als volgt versleuteld naar een blok E: $E = K_s(B\ xor\ R)$. Omdat deze R nu telkens zou moeten worden meegegeven, genereert men enkel voor het eerste blok zo'n willekeurige R. Voor elk volgend blok wordt het vorige blok gebruikt als k-bit getal. Dit is Cipher Block Chaining.

8.4 BESPREEK HET PRINCIPE VAN CRYPTOGRAFIE MET OPENBARE SLEUTELS.

Nu worden twee verschillende sleutels gebruikt: één publiek gekende sleutel en één sleutel die enkel door de ontvanger is gekend. A gebruikt de publieke sleutel van B om boodschap te encrypteren tot $K_B^+(m)$ en A verzendt deze geëncrypteerde boodschap naar B. B decrypteert deze boodschap met zijn private sleutel: $m = K_B^-(K_B^+(m))$. Het is belangrijk dat K_B^- niet berekend kan worden uit K_B^+.

RSA is een encryptieprotocol dat gebruikt maakt van het principe van openbare/private sleutels.

8.5 BESPREEK HET PRINCIPE VAN DIGITALE HANDTEKENING.

De digitale handtekening is een cryptografische techniek die het mogelijk maakt om een handtekening te realiseren in de digitale wereld. Eén manier is om het bericht met een private sleutel te encrypteren en het als digitale handtekening mee te sturen met het oorspronkelijke bericht. De ontvanger kan de handtekening dan decrypteren met de publieke sleutel van de verzender en indien de gedecrypteerde handtekening en het bericht overeenstemmen, is de ontvanger zeker van de integriteit van het bericht.

Een andere manier is om te werken met een message digest, waarbij de digitale handtekening een hash is van het oorspronkelijke bericht. Daarvoor hasht de verzender het bericht, waarna het geëncrypteerd wordt met een private sleutel. De ontvanger decrypteert de hash en maakt een hash van het bericht. Indien beide overeenstemmen, is de integriteit verzekerd. Deze methode is vooral voor lange berichten sneller dan het volledige bericht nog eens te versleutelen zoals hierboven.

8.6 BESPREEK KDC EN CA.

- KDC: Key Distribution Centre, laat twee partijen toe veilig aan de geheime symmetrische sleutel te raken voor onderlinge communicatie. Gebruikers communiceren met dit KDC aan de hand van een geheime sleutel die ze kregen tijdens de registratie bij dit KDC. Wanneer A aan het KDC de geheime sleutel van B opvraagt, zal het KDC deze sleutel terugsturen, maar versleuteld met de unieke sleutel van A. A kan nu die sleutel decrypteren met zijn geheime sleutel en een communicatie opzetten met B. *Voorbeeld: Kerberos*

- CA: Certification Authority, een belangrijk aspect van een digitale handtekening is public key certification: men moet kunnen controleren of een publieke sleutel wel degelijk tot een specifieke entiteit behoort.

  Een publieke sleutel wordt aan een entiteit gebonden door een Certification Authority (CA)(*voorbeeld: VeriSign)*. Een CA controleert eerst of een entiteit is wie hij zegt dat hij is. Indien de CA de identiteit van de entiteit kan verifiëren, creëert hij een certificaat dat de publieke sleutel van de entiteit koppelt aan zijn identiteit (een naam, een IP-adres). Deze certificaten worden geëncrypteerd met de private sleutel van de CA en kunnen worden gedecrypteerd met de publieke sleutel van het CA.

  Als B nu een bericht wil sturen naar A, zal hij zijn certificaat meesturen. A kan het certificaat controleren door het te decrypteren met de publieke sleutel van de CA en komt zo ook B's publieke te sleutel te weten (waarmee hij dan bv. de digitale handtekening kan decrypteren).

8.7 BESPREEK PRINCIPE E-MAIL ENCRYPTIE.

Bij het versturen van e-mails vereist men, indien mogelijk: confidentiality, sender/receiver authentication en message integrity.

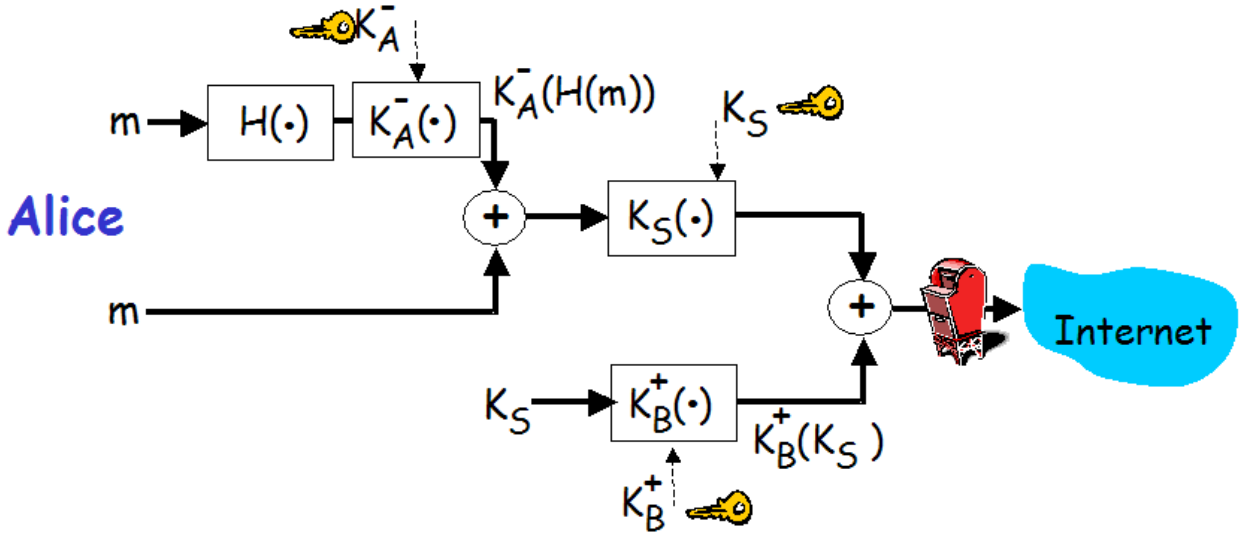
Het PGP-protocol (Pretty Good Privacy) bereikt dit allemaal. Het gebruikt een symmetrische sleutel om de geheimhouding te bewaren, een private sleutel om de identiteit van de verzender te garanderen en een hashfunctie om de berichtintegriteit te verzekeren.

PGP werkt als volgt voor zender A die een bericht 'm' wil verzenden naar B:

1. A hasht m → H(m)
2. A encrypteert H(m) met eigen private sleutel → $K_A^-(H(m))$
3. A voegt H(m) samen met m
4. A maakt een symmetrische sleutel aan, K_s
5. A voegt $K_A^-(H(m))$ en m samen en encrypteert ze met K_s → $K_s(K_A^-(H(m))+m)$
6. A encrypteert de symmetrische sleutel K_s met de publieke sleutel van B → $K_B^+(K_s)$
7. A voegt $K_s(K_A^-(H(m))+m)$ en $K_B^+(K_s)$ samen en verstuurt ze over het internet naar B

PGP werkt dan als volgt voor ontvanger B die 'm' wil lezen

1. B decrypteert de symmetrische sleutel met zijn eigen private sleutel
2. B decrypteert e-mail en gehashte mail
3. B kan al de mail lezen als hij wil, maar moet nu nog de integriteit controleren
4. B decrypteert de hash met de publieke sleutel van A
5. B hasht de e-mail om te controleren dat de hash overeenstemt



8.8 BESPREEK PRINCIPE SSL (“TOY EXAMPLE”).

Secure Socket Layer, is een beveiligingslaag tussen de applicatielaag en de transportlaag. SSL voegt data integrity, server/client authentication en confidentiality toe aan TCP. SSL bestaat uit drie stappen: handshake, key derivation en data transfer.

SSL werkt als volgt:

- client surft met een browser naar een veilige pagina op een SSL-server (https://…)
- SSL-handshaking procedure treedt op:
    - browser zendt SSL-versienummer en cryptografische voorkeuren door
    - server zendt SSL-versienummer en cryptografische voorkeuren door + server zendt zijn certificaat door (= publieke sleutel, geëncrypteerd met de private sleutel van een trusted CA)
    - browser past publieke sleutel CA toe en verkrijgt publieke sleutel server
    - browser maakt symmetrische sessiesleutel aan en encrypteert die met publieke sleutel server
    - browser verzendt de geëncrypteerde sessiesleutel
- SSL tunnel is gerealiseerd, gegevens kunnen nu met sessiesleutel beveiligd verzonden worden

8.9 BESPREEK PRINCIPE IPSEC: TWEE MODES, SA

Via IPSec zal de gehele payload van het IP-pakket geëncrypteerd en/of geauthenticeerd worden, waardoor alle data uit de bovenliggende protocollen onzichtbaar zal worden (“blanket coverage”). Wordt gebruikt voor VPN.

Twee modes:

- transport mode: bij transport mode wordt het IPSec-datagram verstuurd en ontvangen door de twee hosts. De gegevens van het datagram worden omhuld met header- en trailervelden en een authenticatiegegevensveld.
- tunnel mode: twee routers zijn op de hoogte van de IPSec verbinding en voeren dit volledig transparant uit voor de hosts. De IPSec-transformatie wordt uitgevoerd op het volledige datagram en een nieuw datagram wordt verstuurd

SA: vooraleer informatie via IPSec verstuurd kan worden, dient er een virtuele simplexconnectie opgezet te worden door de twee eindpunten waarin alle beveiligingsinstellingen worden doorgegeven. IPSec is dus connection oriented. Dit wordt de Security Agreement (SA) genoemd.

8.10 BESPREEK PAKKETFILTERING “PACKET FIREWALL: STATELESS EN STATEFUL” EN TOEPASSINGSGATEWAY “APPLICATION GATEWAY”

Als een firewall rekening houdt met de openstaande TCP connecties om te beslissen of hij een pakket zal tegenhouden of niet, dan noemen we hem stateful. Dan wordt er ook gebruik gemaakt van een connection table, naast de lijst met algemene rules. Stateless firewalls werken pakket per pakket.

Application gateway: Hiermee kunnen we kan traffic gefilterd worden aan de hand van applicatieregels. De application gateway wordt als een tussen gebruikt langs waar alle traffic moet passeren en enkel de pakketten die van deze source komen worden doorgelaten door de firewall. Hiervoor is het wel belangrijk dat de applicaties weten hoe ze deze application gateway moeten contacteren.

HOOFDSTUK 10 : IPV6

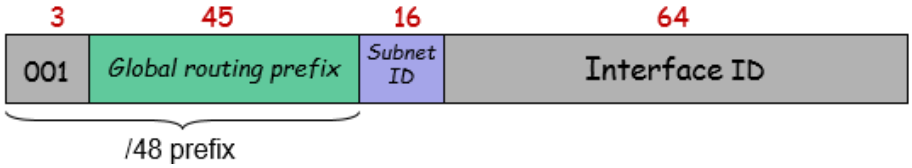
10.1 BESPREEK DE VERSCHILLENDE TYPES ADRESSEN BIJ IPV6.

- Unicast: one-to-one, gebruikt om een enkele netwerkinterface te identificeren, het pakket met dit adres als bestemming wordt afgeleverd bij die specifieke computer
    - link-local addres
    - global address
    - unique local address
- Multicast: one-to-many, gebruikt om een verzameling van interfaces te identificeren die normaal gezien tot verschillende nodes behoren. Pakketten wordt geleverd aan en verwerkt door alle adressen die geïdentificeerd worden door dat adres
- Anycast: one-to-nearest = one-to-one-of-many, wordt net zoals multicast toegewezen aan een groep interfaces. Een packet dat verzonden wordt naar een anycast-adres wordt geleverd naar juist één van de interfaces die behoren tot die groep. Typisch is dit de 'dichtste', afhankelijk van hoe de router het begrip afstand benadert.

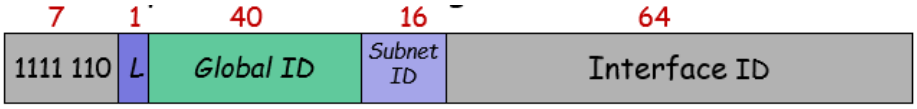
10.2 LEG UIT: FE80::/10, 2000::/3, FC00::/7. WAARVOOR WORDEN DEZE VERSCHILLENDE SCOPES GEBRUIKT?

- FE80::/10: link-local address
    - automatisch gegenereerd door node voor al zijn interfaces

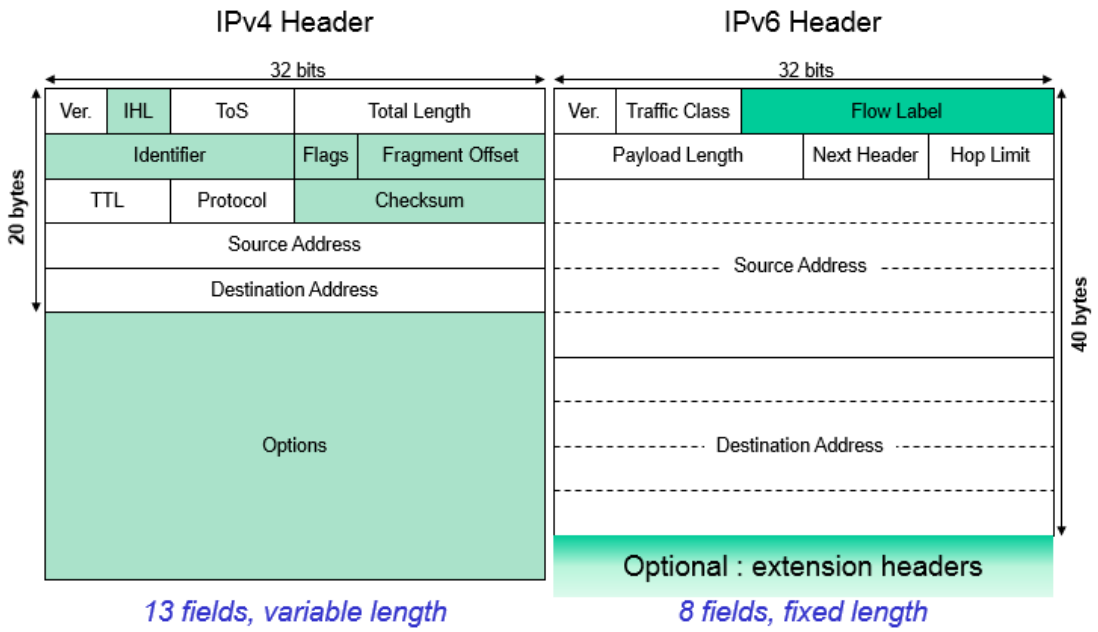
- 2000::/3: global unicast address
    - benaderbaar over heel de wereld, functionaliteit van klassiek IPv4-adres
    - wordt gealloceerd in /48-ranges en laat subnetting toe dankzij 16-bit subnet ID



- FC00::/7: unique local address
    - vervangen het deprecated 'site-local address'
    - enkel benaderbaar in privaat netwerk of een gelimiteerde verzameling ULA-sites, functie van privaat IPv4-adres



10.3 WAT ZIJN DE BELANGRIJKSTE VERSCHILLEN TUSSEN EEN IPV4 EN IPV6 HEADER ? WAAROM HEEFT MEN DIE VERSCHILLEN INGEVOERD ?



De IPv4-header is 13 velden groot en heeft een variabele grootte van minstens 20 bytes. De IPv6-header is 8 velden groot en heeft een vaste grootte van 40 bytes, verzender- en ontvangeradres elk al 16 bytes!

Wijzigingen:

- Version: 6 → 4
- ToS → Traffic Class
- Total Length → Payload Length: header heeft een vaste grootte
- Protocolveld → Next Header: geeft aan welk type de header heeft die volgt op de IPv6 header, dient eveneens ter vervanging van het optieveld door indien nodig aan te geven welke extensieheaders volgen
- TTL → Hop Limit: in principe wordt geen tijd gemeten, dus hop limit is betere benaming

Verdwenen velden:

- Header Length: headerlengte is vast
- Checksum: duur en reeds geïmplementeerd op de datalink- en transportlaag
- ID/flags/offset: routers fragmenteren pakketten niet langer dankzij Path MTU discovery
- Options: zie Next Header

Toegevoegde velden:

- Flow Label: gebruikt om pakketten te groeperen

10.4 GEEF EEN VOORBEELD VAN IPV6 ADRESRESOLUTIE.

De tegenhanger van ARP in IPv4 is het Neighbour Discovery Protocol (NDP), dat gebruikt maakt van ICMPv6. Het doel van NDP is net als bij ARP een vertaling te maken tussen netwerklaag en transportlaagadressen en is er gekomen vanwege de slechte schaalbaarheid van ARP, dat slecht schaalbaar is omdat het zorgt voor een interrupt op elke aangesproken host. Het NDP-protocol maakt. IPv6 Address Resolution maakt gebruikt van pseudo unicast: multicast zonder alle nodes te interrupten.

Voorbeeld: wanneer host A het link-layeradres van host B te weten wil komen, stuurt hij een Neighbour Sollicitation naar het sollicited-node multicastadres dat gebaseerd is op het adres van B (prefix FF02::1:FF00:0/104+laatste 24 bits van unicastadres van B). Er wordt dan geantwoord in een Neighbour Advertisement in unicast.

10.5 LEG UIT: IPV6 DAD.

In IPv6 hebben nodes de mogelijkheid en verantwoordelijkheid hun eigen IID te generen, gebaseerd op hun MAC-adres. Dit biedt het voordeel dat de nodes kunnen communiceren zonder dat een centrale server de adressen moet beheren of een beheerder statische entries moet gaan maken.

Vooraleer een IPv6-adres gebruikt mag worden, moet er gecontroleerd worden dat het uniek is. Hiervoor kunnen dezelfde ICMPv6-boodschappen worden gebruikt als bij Address Translation.

Een functie die Duplicate Address Detection heet, wordt hiervoor ingezet. Ze werkt als volgt:

1. De IPv6-node wijst het IPv6-adres toe aan haar interface, maar markeert het als een voorlopig adres (tentative): het mag nog niet gebruikt worden. De node joint de overeenkomstige sollicited-node multicast groep, alsook de allnodes groep ff02::1
2. Zoals in address resolution, stuurt de node een Neighbour Sollicition naar het sollicited-node multicast address. Het eigen adres is nog tijdelijk, dus is de bron ‘unspecified’.
3. Als er geen NA komt na een zekere time-out, adres uniek en tentative → preferred

10.6BESPREEK DE VERSCHILLENDE AUTOCONFIGURATIESTAPPEN VAN IPV6.

1. Link Local in orde: genereer een tentative link-local adres op basis van EUI64 (MAC-adres) en word lid van de all-nodes en sollicited-node multicast groepen. Voer duplicate address detection (DAD) uit en leg vervolgens het link-local adres vast als voorkeursadres.
2. Stateless, Global Address ok: voer een Router Sollicitation uit via het all-routers multicast adres (FF02::2). De router antwoordt door een Router Advertisement te sturen naar all-nodes multicast adres (FF02::1) met daarin de prefix en parameters maar zonder DNS-informatie. De host zal nu een IPv6-genereren op basis van de gegeven prefix en zijn IID.
2. Stateful, DNS ok: dit is een optionele stap na de stateless autoconfiguration. Als de Router Advertisement de managed/other-flag aanzetten, dan is er een DHCPv6-server. De host kan nu de DHCPv6-server aanspreken met behulp van het multicast FF02::1:2-adres

10.7 GEEF DE BASISPRINCIPES DIE KUNNEN GEBRUIKT WORDEN BIJ EEN OVERGANG VAN IPV4 NAAR IPV6.

- Dual Stack: laat toe om incrementeel IPv6 te introduceren in bestaande netwerken. Afhankelijk van de andere partij zal de host IPv4- of IPv6-pakketten versturen. Hiervoor werd DNS een extra type record geïntroduceerd, het AAAA-record. Een Dual Stack server zal dan zowel een A- als AAAA-entry record hebben. Het probleem met Dual Stack is dat alle applicaties IPv4- en

IPv6-compatibel moeten zijn. Ook de ISP moet IPv6 ondersteunen. En bovendien is er dubbel onderhoud: de routing tables, firewall,…

- Tunneling: wordt gebruikt als de ISP geen IPv6 ondersteunt. Er wordt een tunnel gebruikt op het bestaande IPv4-netwerk en de IPv6-pakketten zijn de payload van de IPv4-pakketten. Voorbeelden van tunneling zijn 6to4 tunneling en Teredo.
- Translation: wordt gebruikt als je LAN IPv6 only is. De oplossing doet denken aan NAT. Er vindt een vertaling plaats van IPv6-addressen naar IPv4-addressen: het bronadres wordt vertaald naar een IPv4-adres, hetzelfde gebeurt voor het doeladres. De payload van het IPv6-pakket wordt de payload van het IPv4-pakket. De nadelen zijn de slechte schaalbaarheid van deze oplossing en het feit dat niet alle IPv6-features vertaald kunnen worden naar IPv4.