Communicatienetwerken

Antwoorden en vragen 2010-2011

Met dank aan:

Jeroen Van de Sande, Nicolas De Smyter, Sofie Van Gassen, Quentin Braet, Robrecht Cannoodt, Pieter De Baets

Hoofdstuk 1 : Inleiding

1. Bespreek de structuur van het Internet als "network of networks"

Het internet is een netwerk. Dit netwerk (tier 1) bevat allemaal sub netwerken (tier 2). En deze netwerken kunnen nog eens andere netwerken bevatten. Zo bestaat er nog een tier 3 en een toegangs ISP. De uiteindelijke hosts of eindgebruikers bevinden zich pas op deze allerlaatste laag. De toegangs ISP (lokale ISP) vormt dan een netwerk, en alle bovenliggende ISP lagen zijn netwerken van netwerken.
Grof genomen is het internet dus een aaneenkoppeling van netwerken

2. Leg uit wat een protocol is

Een protocol is een afgesproken manier van communiceren tussen 2 computers. Hierbij wordt er vastgelegd wat er verstuurd wordt, en op welke manier. Tevens worden eventuele volgordes en/of antwoordregels vastgelegd.
Een menselijk protocol is bv 'hey', 'hey', 'hoe laat is het?', '5 voor 12'.

3. Geef de verschillende lagen van het TCP/IP referentiemodel. Geef bij elke laag een voorbeeld (leg kort uit)

Mogelijke ezelbruggetjes
Applicatie
(FTP (filezilla, proftpd), SMTP(sendmail), HTTP (apache etc))
Toepassingen
Transport
(TCP, UDP)
Host to host
Netwerk
(IP)
Host to host, routering
Datalink
(Ethernet, PPP)
Point to point
Fysieke link
(twisted pair, koper)
Point to point, bits "on the wire"

4. Bespreek de algemene werking van FTP

Werkt via TCP. Er is een controle connectie (poort 21 op server) voor commando's en een data connectie (poort 20) voor overdracht van gegevens. Eerst worden via de controle connectie de gegevens (login, password..) gecontroleerd alsook de commando's. De effectieve data wordt dan verzonden via de dataconnectie.

5. Bespreek algemeen de eigenschappen van TCP en van UDP. Leg uit en vergelijk

UDP (User Datagram Protocol)

- Geen aankomst garantie
- Geen volgorde garantie
- Enkele richting (unidirectioneel)
- Connectionless (geen staat)
- Snel
- DNS, streaming, IM (in zeker zin)

- Geen flow of congestion control
- Data-eenheid: datagrampacket
- bv: DNS of RIP

TCP (Transmission Control Protocol)

- Bevat staat
- Garantie op afleveren en volgorde
- Bi directioneel (full duplex)
- 3 way handshake
- Trager, maar robuuster
- Congestion en flow control
- Data-eenheid: segment
- bv: HTTP, SMTP, FTP of telnet

6. Bespreek de algemene werking van het internetprotocol (IP) en de typische eigenschappen

Is een routering van boodschappen over de verschillende netwerken en intern in een netwerk op basis van een unieke cijfercombinatie ter identificatie van de bestemming van het pakket. In router wordt er voor gebruik gemaakt van routeringstabel.

- Best effort
- Enkele richting
- Connectionless
- Data-eenheid: IP datagram
- adres bestaat uit 4 * 1 byte, voorgesteld met een 'dotted decimal' notering

7. Bespreek het principe van encapsulatie

Een pakket opgesteld door een bovenliggende laag wordt door de onderliggende laag integraal in zijn payload opgenomen. Het is als het ware een soort matroesjka met een schil boven een andere schil (binnenste is data van applicatielaag, buitenste van datalinklaag). Anders gezien kan gesteld worden dat iedere laag zijn eigen header voor deze van de vorige laag plaatst.

Bv: data => {data} | TCP => {data | TCP} | IP (TCP = header TCP, IP = header IP)

8. Leg het verschil uit tussen een host en een router

Een host is een definitieve eindbestemming, de gastheer voor de toepassingen. Een router (=packet switch) zal enkel de pakketten verder doorheen het netwerk routeren. Tevens zal een router maar tot de netwerklaag kunnen lezen en de host tot de applicatielaag.

9. Leg uit : client en server laag. Geef een voorbeeld. Vergelijk met client/server bij applicaties

De client laag geeft gegevens door aan de server laag voor verzending, en zal van de server laag antwoorden ontvangen. De server laag geeft services aan de client laag. Net zoals een client-server.

Bv: client-server : applicatie - transport, transport - netwerk..

10. Bespreek identificatie in de applicatie-, transport- en netwerklaag

- Applicatie: username en/of wachtwoord
- Transport: poort
- Netwerk: ip adres

11. Hoe noemt men de informatieblokken in de applicatie-, transport-, netwerk- en datalinklaag ?

- Applicatie: message
- Transport: segment (TCP) of datagram (UDP)
- Netwerk: IP-datagram
- Datalink: frame

12. Wat is : IETF, RFC, ISP ? Geef kort uitleg

- IETF: Internet Engineering Task Force, ontwikkelt web standaarden
- RFC: Request For Comment, indienen van een standaard (voor een protocol) omschrijving waarop mensen dan commentaar kunnen geven. (of aanpassingen)
- ISP: Internet Service Provider, verleent toegang tot het internet aan klanten.

Hoofdstuk 2 : Applicatielaag

1. Bespreek het client-server principe, op applicatieniveau.

De client vraagt een dienst aan de server, de server beantwoordt dan aan deze dienst. Beiden dienen hiervoor (dezelfde) applicatie te draaien en zijn een host in een netwerk.

- Server:
  - Vaste toegangspoort
  - Meerdere connecties via deze poort tegelijk
  - Luistert (passief) en zendt dan pas
  - High end machines
- Client:
  - Willekeurige poort
  - 1 connectie/poort
  - Zend (actief)
  - Bv. browser

2. Bespreek het concept van “threads”. Geef een voorbeeld.

Draaien op de server. Laat toe om meerdere clients die via de standaard poort verbinden simultaan te beantwoorden. Voorbeeld (webpagina): 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 gebruikers toekomen. Deze krijgen dan ook telkens een (nieuwe) thread toegewezen. Als de thread klaar is, verstuurt deze dan het antwoord naar de desbetreffende client die op het antwoord wacht.

3. Welke transportdiensten kan een applicatie vereisen? Geef enkele voorbeelden.

- Zekerheid van aankomst
- Tijdsgevoelig
- Bepaalde (minimum) bandbreedte vereist
- Beveiliging
- Volgorde van segmenten
- Bv:
  - bestand doorsturen: geen verlies, volgorde belangrijk
  - mail: geen verlies, volgorde belangrijk
  - streaming : snel, verlies wordt getolereerd, minimale bandbreedte vereist

4. Bespreek het HTTP protocol en de belangrijkste protocolboodschappen.

Wordt gebruikt om gegevens van een webserver te ontvangen. Kan persistent zijn (HTTP/1.1) of niet(HTTP/1.0). Persistent en niet-persistent zie vraag 6.

Standaard volgende volgorde boodschappen (1.0):
=> **Vraag TCP verbinding**

<= **Accepteer TCP verbinding**
=> **Stuur vraag naar object**
<= **Ontvang vraag, stuur object, sluit tcp**
=> **Ontvang object en sluit tcp** (herhaal voor andere objecten)

Tevens bevat het een aantal standaard commando's:

- GET
- POST
- PUT (1.1) upload een file
- HEAD (filtering, if modified since etc)
- DELE (1.1) delete een file

5. Waarvoor staat : HTTP, URL, HTML? Geef kort uitleg.

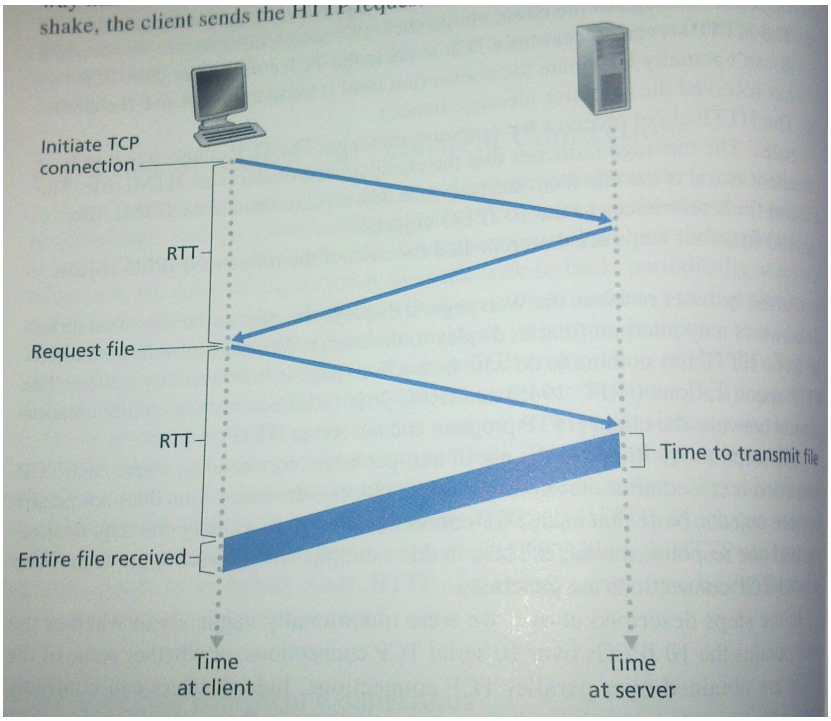
- HTTP: Hyper Text Transfer Protocol, protocol om gegevens over internet te verzenden
- URL: Uniform Resource Locator, uniek id om document op internet te vinden. Bestaat uit hostname (bv ugent.be) en pad (bv. /users)
- HTML: Hyper Text Markup Language: programmeertaal waarmee webpagina's gemaakt worden, wordt gebruikt om koppelingen naar andere documenten weer te geven (bv afbeeldingen)

6. Bespreek de verschillende HTTP connectiemogelijkheden.

- Persistent
  - Connectie blijft behouden
  - Met of zonder pipelining
    - Met pipelining: opvragen verschillende opvragen tegelijk versturen
    - Zonder pipelining: aanvragen sequentieel achter elkaar
  - Standaard voor HTTP/1.1
- Niet persistent
  - Voor ieder object nieuwe verbinding
  - Trager
  - Standaard voor HTTP/1.0

7. Bespreek een eenvoudig model voor responstijd bij HTTP.

Om de TCP-connectie op te zetten is er eerst RTT nodig. Eenmaal de verbinding opgezet is kan de request doorgestuurd worden. Het duurt dus RTT voor het antwoord hierop ontvangen wordt. Dan is er nog de transmit time nodig om alles te ontvangen.



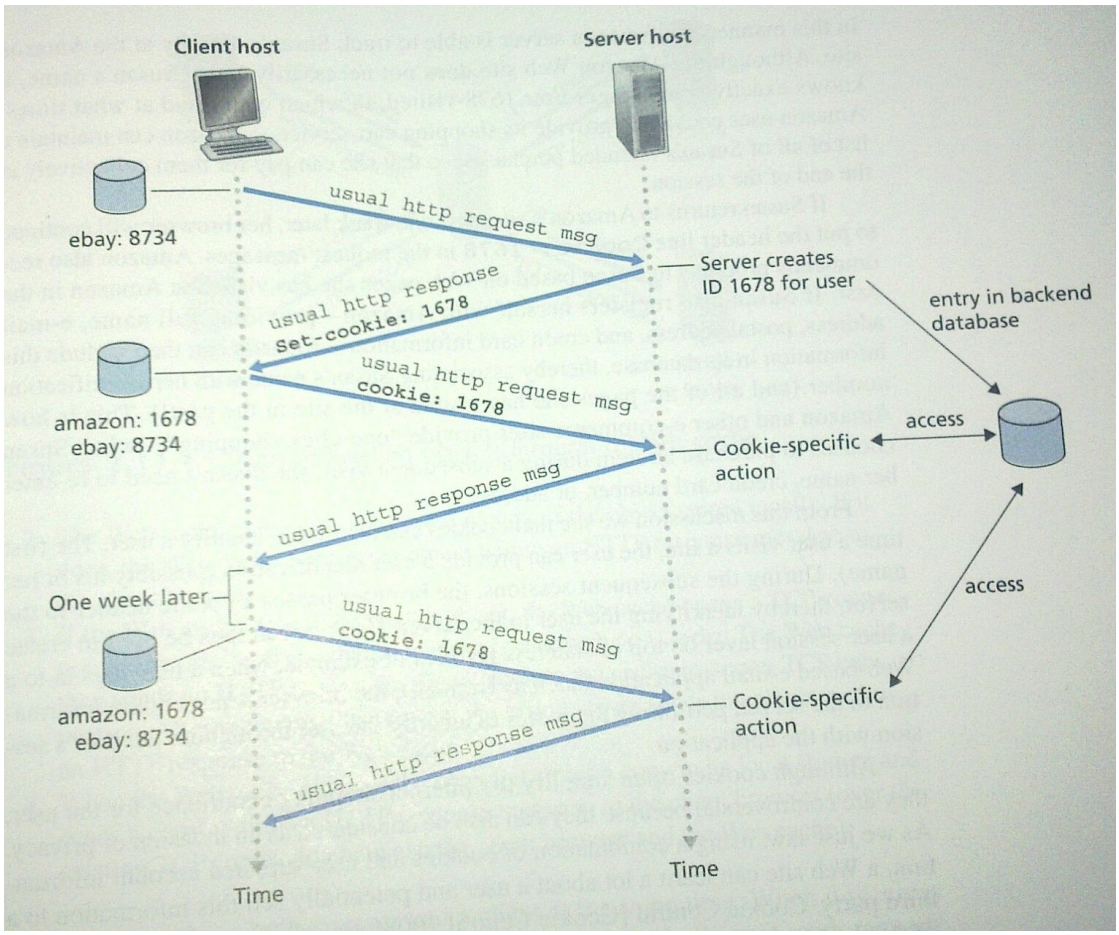
Stel X := Aantal objecten
RTT := Round-trip time
TTF := Time to transmit file

Totale tijd om X objecten door te sturen:

Persistent zonder pipelining	RTT + X * (RTT + TTF)
Persistent met pipelining	2 * RTT + X * TTF
Niet-persistent	X * (2 * RTT + TTF)

8. Bespreek het principe van cookies.

- Houden info over gebruiker bij, voor opvang van statelessness van HTTP
- Back end database op server (cookie id bv verbonden met login en password in database)
- Cookie managing door web browser op client



9. Bespreek conditional GET en waarvoor is dat nuttig?

- Handig voor proxy servers => verkort ook opvraag tijd
- Filteren van de aanvraag
- Beperken bandbreedte, omdat het enkel doorgestuurd wordt als het nieuw is
- If not modified since

10. Geef een overzicht van de verschillende e-mail protocols (afkorting + korte uitleg wat het doet).

- SMTP: Simple Mail Transfer Protocol: push email naar eigen/andere mail servers
- POP3: Post Office Protocol 3: haal email van de eigen mail server op naar host/user agent (pull).
    - Authenticatie
    - Opvragen mails
    - Update (laat bericht op server of delete)
- IMAP: Internet Message Access Protocol : soortgelijk aan pop maar dan inclusief verschillende mappen structuur etc
- HTTP: verzenden en lezen van email via webbrowser
- RFC 822: message format: hoe enkel tekst versturen
- MIME: Multipurpose Internet Mail Extensions: formaat en codering van niet tekst gebaseerde bestanden (multimedia)

11. Bespreek het gebruik van naam en adres. Geef een voorbeeld.

- Adres: ip, cijfer combinatie van 4 keer 8 bit (of 4 bytes), bv. 157.193.128.10
- Naam: hostnaam, makkelijk te onthouden voor mensen, bv. www.ugent.be

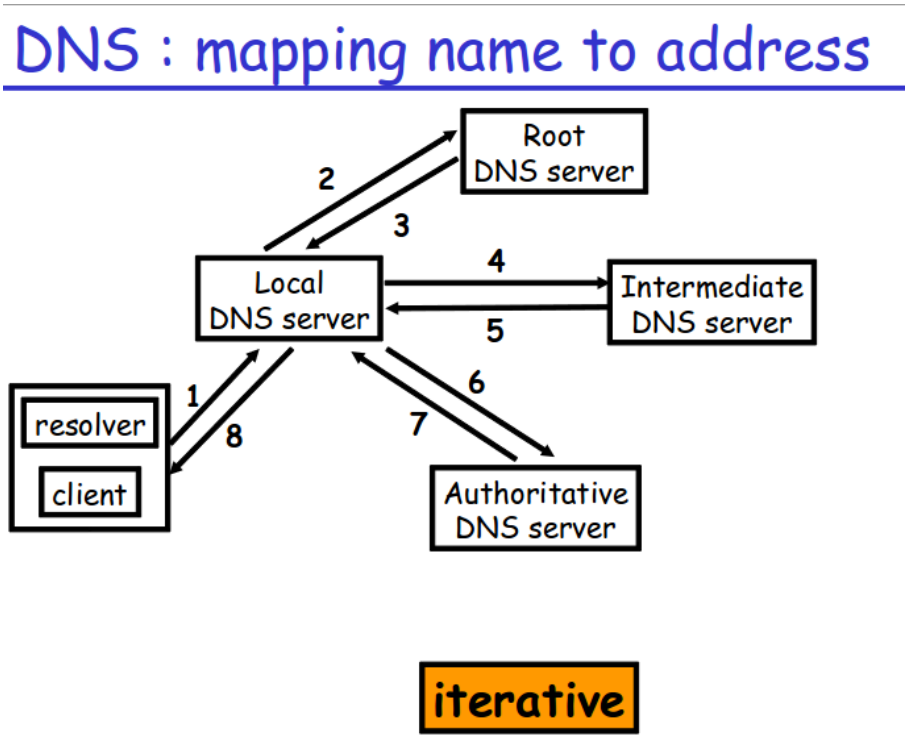
12. Bespreek de DNS hierarchie. Geef een voorbeeld.

- Root DNS: top level DNS server (13 in de wereld), verwijst door naar lokale DNS registrars
- Top-level domain servers (TLD): verantwoordelijk voor top-level domeinen (bv com, be..)
- Authorative DNS server: houdt de records van publiek beschikbare hosts bij
- Lokale DNS server: DNS van ISP (zowel lokaal als niet-lokaal)
- Intermediate DNS server: tussenliggende server (vb TLD server) server met records eventueel in cache

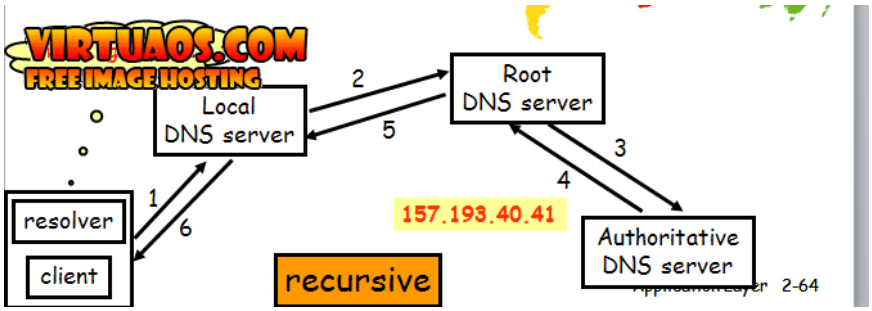
DNS is een applicatielaagprotocol dat gebruik maakt van een gedistribueerde database om een hostname te vertalen naar zijn adres.
Een webbrowser zal altijd eerst de lokale DNS opvragen. Indien deze de gevraagde record niet bezit zal de lokale DNS server via de root DNS opvragen wie er verantwoordelijk is voor de top level van het gevraagde domein. Deze kan eventueel de records nog in cache hebben (intermediate) en ze direct doorsturen. Indien hij de records niet meer in cache heeft dan zal de webbrowser aan deze top level kunnen vragen wie de authorative DNS is en daar dan zijn gegevens aan vragen.
Opvragen via DNS kan recursief of iteratief.

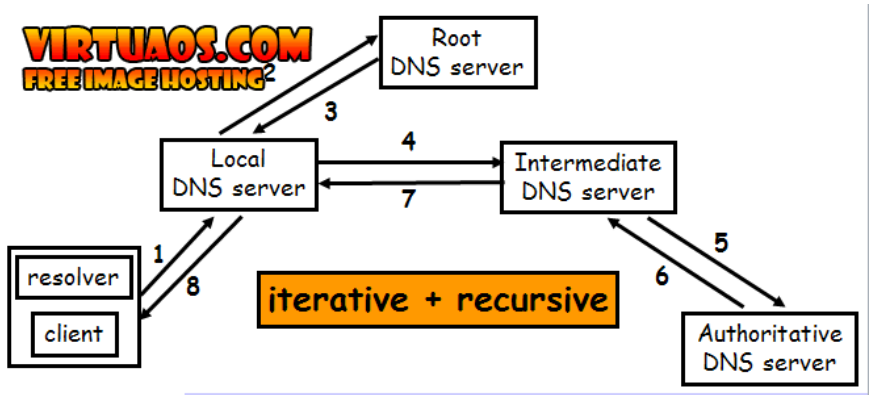
13. Bespreek het iteratief en recursief mappen in DNS.

Iteratief wil zeggen dat als een host een aanvraag naar een DNS richt, en deze daar niet voor verantwoordelijk is, deze dan het adres zal teruggeven aan de host van een andere DNS die waarschijnlijk het antwoord zal weten.



Recursief wil zeggen dat als een host een aanvraag naar een DNS richt, en deze daar niet verantwoordelijk voor is, de root DNS dan zelf de verantwoordelijke server zal connecteren. De connectie tussen de host en de root DNS zal verbonden blijven tot de root DNS het antwoord heeft gekregen van de verantwoordelijke server en dat teruggeeft aan de host.





caching of name/address translation pairs
- caching in intermediate name servers
- improve delay performance of name/address translation
- reduce number of DNS queries on the network
- cached record is valid limited in time (few days)
- very limited number of requests towards root servers

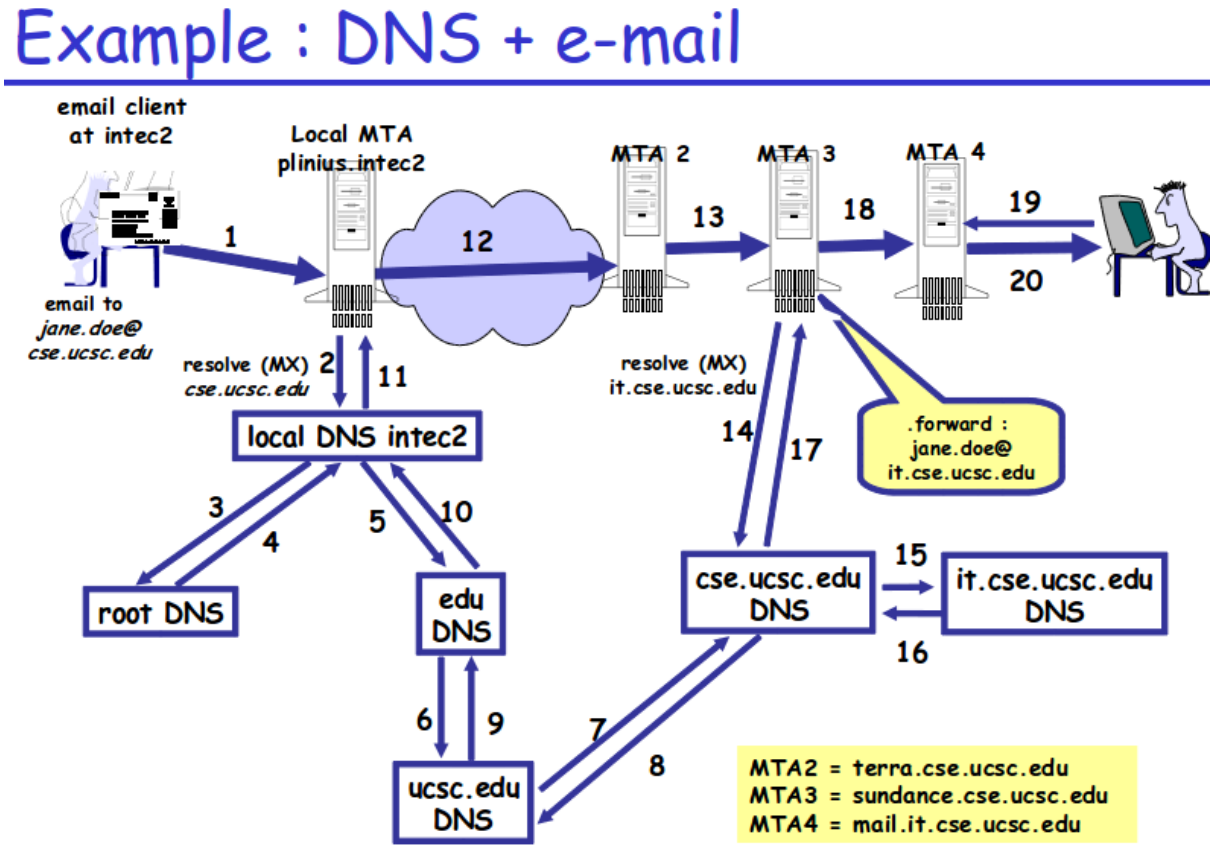
14. Wat is (in de context van DNS) : RR, A, NS, CNAME, MX (geef ook een voorbeeld)

- RR: resource records, hoe DNS records op te slaan
  - naam die vertaald moet worden
  - TTL, hoe lang wordt de record gecached
  - Class (IN voor internet)
  - Record type (A,NS, CNAME of MX)
  - Record data: waarde, bv een ip adres
- A: address record; waar de server te vinden is (www.ugent.be is te vinden op 157.192....)
  - Name: host name, data: ip adres
  - plinius.intec.ugent.be IN A 157.193.214.4
  - Effectieve vertaling van hostname naar IP adres
- NS: waar de name server te vinden is voor dat domein
  - Name: host name, data: hostname van machine die de gegevens kent
  - ugent.be IN NS ugdns1.ugent.be (authoritative name server for ugent.be)
  - Doorverwijzing naar andere DNS server die verantwoordelijk is voor hostname
- CNAME: canonical hostname
  - mail2.intec.ugent.be IN CNAME plinius.intec.ugent.be
  - alias voor een host name. Name en data beide een hostname
- MX: mail record: waar de mailserver te vinden is te vinden op een hostnaam, en met welke

voorkeur als er meerdere zijn.

- intec.ugent.be IN MX preference=10 mail-tech.intec.ugent.be
  preference=30 cedar.ugent.be

15. **Op het examen wordt een voorbeeld gegeven (b.v. MIME header, HTML file, DNS request) en er wordt gevraagd dat te bespreken.**
    - MIME: (Multipurpose Internet Mail Extension) zie Hoofdstuk 2, slides 57-58
    - HTML: (Hyper Text Markup Language) zie Hoofdstuk 2, slides 33-35
    - DNS: (Domain Name System) zie Hoofdstuk 2, slides 71-72
16. **Bespreek het voorbeeld DNS + e-mail (op het einde van de paragraaf over DNS). De figuur wordt opgegeven.**



Als een mail verstuurd moet worden, zal de email client van de gebruiker de mail pushen op de lokale MTA (Message Transfer Agent). Deze MTA (plinius.intec2) zal dan de lokale DNS contacteren om de MTA van de ontvanger te bepalen. De lokale DNS (intec2) zal eerst de root DNS contacteren, waarna hij ook iteratief de DNS server zal contacteren die voor het top-level domain “edu” verantwoordelijk is. We kunnen zien dat het **iteratief+recursief** is, want de connectie tussen de root DNS en intec2 stopt nadat het adres van het top-level domain teruggeven is. De DNS aanvraag gaat dan daarna verder vanuit de lokale DNS. Deze zal recursief connecteren met ucsc.edu en deze op z’n beurt ook recursief met cse.ucsc.edu. Deze zal antwoorden met het adres van MTA3. We keren nu op onze stappen terug tot in de lokale MTA1 (stappen 8, 9, 10 en 11). Nu de lokale MTA1 het juiste adres van MTA3 weet, zal deze de mail doorsturen naar MTA3 (via MTA2).

Eenmaal de mail toegekomen is op MTA3 moet deze doorgestuurd worden naar john.doe@it.cse.ucsc.edu (volgens de forward-regel). Om dit te doen moet eerst de DNS

gecontacteerd worden om te weten waar mail.it.cse.ucsc.edu zich bevindt. Dit gebeurt door de cse.ucsc.edu-DNS-server te contacteren (wat voor MTA3 de lokale DNS zal zijn). Deze zal het recursief opvragen aan it.cse.ucsc.edu die antwoord kan geven waar de mail-server zich bevindt. Dan keren we op onze stappen terug (stap 16 en 17) tot in MTA3. MTA3 weet nu naar waar de mail gestuurd moet worden en zendt het door naar MTA4.

Als de ontvanger nu zijn clientapplicatie opent, zal deze applicatie met MTA4 verbinden om via POP3 of IMAP zijn mails op te vragen. Als MTA4 antwoordt met de mails die voor hem bestemd zijn is de mail ontvangen en eindigt dit verhaal ^^

17. Leg de figuur uit die de prestatie van C/S en P2P vergelijkt (de figuur wordt opgegeven).

- Client server
  - server zendt sequentieel N kopieën (server: N keer F bytes uploaden)
  - Client i heeft F/di tijd nodig om te downloaden
  - $$D_{cs} = max\left\{\frac{N*F}{u_s}, \frac{F}{d_{min}}\right\}$$
- P2P
  - Server moet maar 1 keer uploaden (tijd = F/us)
  - Client i heeft tijd F/di nodig om te downloaden (blijft zelfde)
  - $$D_{P2P} = max\left\{\frac{F}{u_s}, \frac{F}{d_{min}}, \frac{N*F}{u_s + \sum_{i=1}^{N} u_i}\right\}$$

N = aantal clients, F = grootte file, u_s = server upload, u_i = client i upload, d_i = client i download

18. Bespreek het principe van DHT.

- Ditributed hash table
- Database heeft key en value, de key kan een bestandsnaam zijn, en de value het ip waar het op te halen is
- Gebruikers kunnen de DHT databank dan opvragen. Dit is zeer eenvoudig in een client-server model. Echter iets moeilijker in een p2p omgeving.
  - Circulair (zie afbeeldingen hoofdstuk 2 slides 89-90)
  - Circulair met shortcuts (zie afbeelding hoofdstuk 2 slide 91)
- Aanvullend: peer churn
  - Peers pingen hun opvolger(s)
  - Wat als 4 wegvalt?
  - 3 maakt 5 z'n opvolger en vraagt aan 5 wie zijn directe opvolger is (6). 3 maakt dan 6 z'n 2e opvolger

Hoofdstuk 3 : Transportlaag

1. Bespreek multiplexering (connection-oriented and connectionless)

Multiplexen: Verpakken van data-segmenten en toevoegen van header-informatie (voor demultiplexering) zodat het kan doorgegeven worden aan netwerklaag om te versturen.
Demultiplexen: uitpakken van de verpakte data-segmenten.

- Connectionless (bv DNS)
    - Datagrams (UDP)
    - Sockets aangemaakt met unieke poort
    - Datagram bevat src port en dest port
- Connection oriented (bv HTTP)
    - Sockets krijgen elk hun poort
    - Pakketten krijgen src en dest port èn ip
    - Sorteren per ip en op volgorde van verzenden bij aankomst (demulitplexeren)
    - Server kan meerdere sockets hebben per gebruiker/verbinding (ieder verschillend door ook opnemen van een ip adres)

2. Bespreek de verschillende velden van een TCP segment (het segment zelf wordt gegeven op het examen).

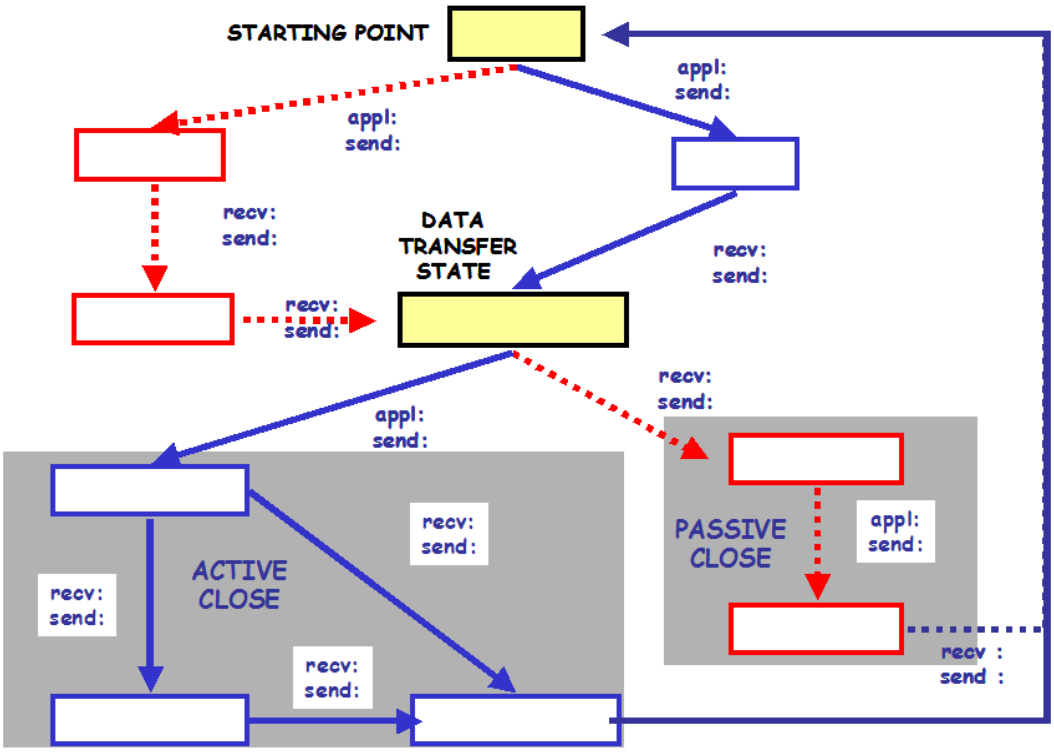
<table>
<tr><td colspan="8">16 bit source port nr</td><td colspan="8">16 bit destination port nr</td></tr>
<tr><td colspan="16">32 bit sequence nr</td></tr>
<tr><td colspan="16">32 bit ACK nr</td></tr>
<tr><td colspan="4">4 bit header</td><td colspan="4">not used (6 bits)</td><td>U
R
G</td><td>A
C
K</td><td>P
S
H</td><td>R
S
T</td><td>S
Y
N</td><td>F
I
N</td><td colspan="8">16 bit window size</td></tr>
<tr><td colspan="8">16 bit checksum</td><td colspan="8">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 dest poort: 16 bit elk, gebruikt ter identificatie
- 32 bit sequentie nummer: nodig voor volgorde
- 32 bit ack nummer: geeft nr van volgende segment (dat eventueel ng verstuurd moet worden)
- 4 bit lengteveld: lengte van de TCP-header in 32-bit woorden
- 16 bit window size: hoeveel bytes de ontvanger nog kan ontvangen.
- 16 bit checksum: om te zien of bestand correct is.
- 16 bit urgent pointer: waar in het segment er eventueel belangrijke data staat
- Options: maximum segment size etc
- Vlaggen
    - ACK-bit: waarde in bevestigingsveld is geldig
    - RST- , SYN- en FIN-bit: gebruikt bij het tot stand brengen/afsluiten van de verbinding
    - PSH-bit: ontvanger moet de gegevens direct bezorgen bij de bovenliggende laag
    - URG-bit: segment bevat gegevens die door de verzender als 'urgent' zijn aangemerkt

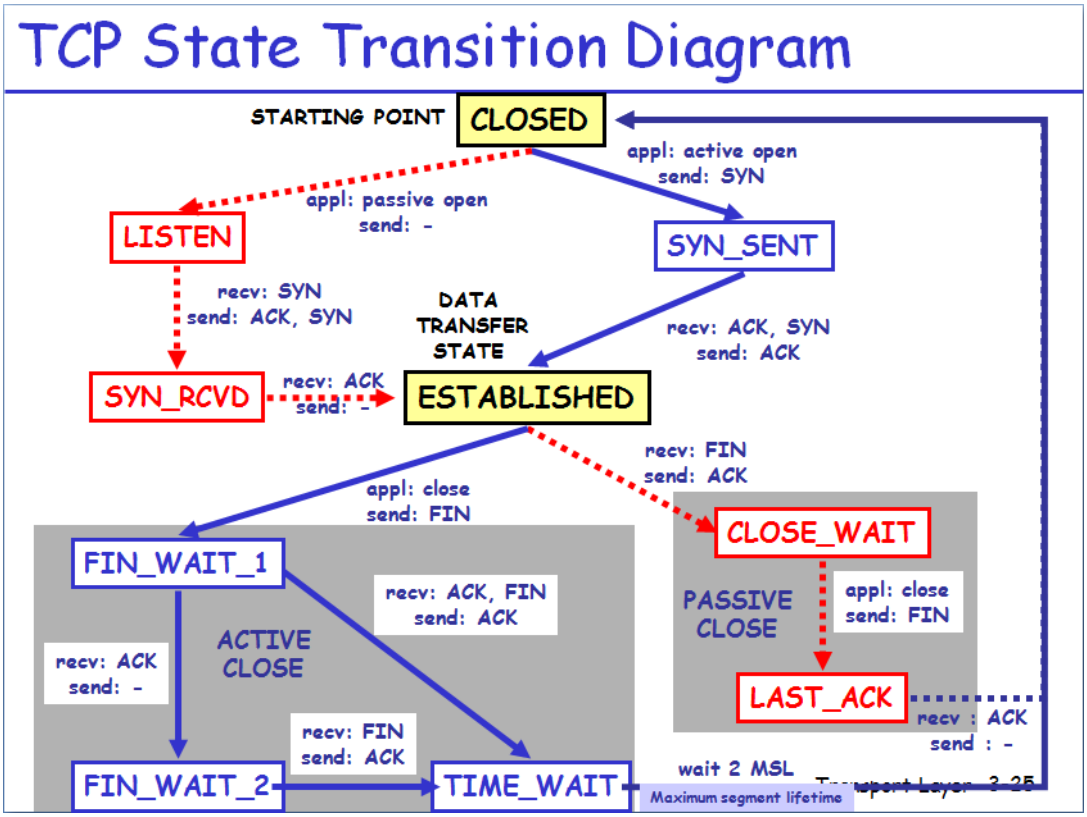
3. Bespreek het TCP toestandsdiagramma (een skeletfiguur wordt opgegeven, zonder enige tekst). Maak bij de uitleg eveneens gebruik van een tijdsverloop (tijdsas client- en serverzijde aangeven en welke boodschappen er uitgewisseld worden).

Opgegeven skeletfiguur:

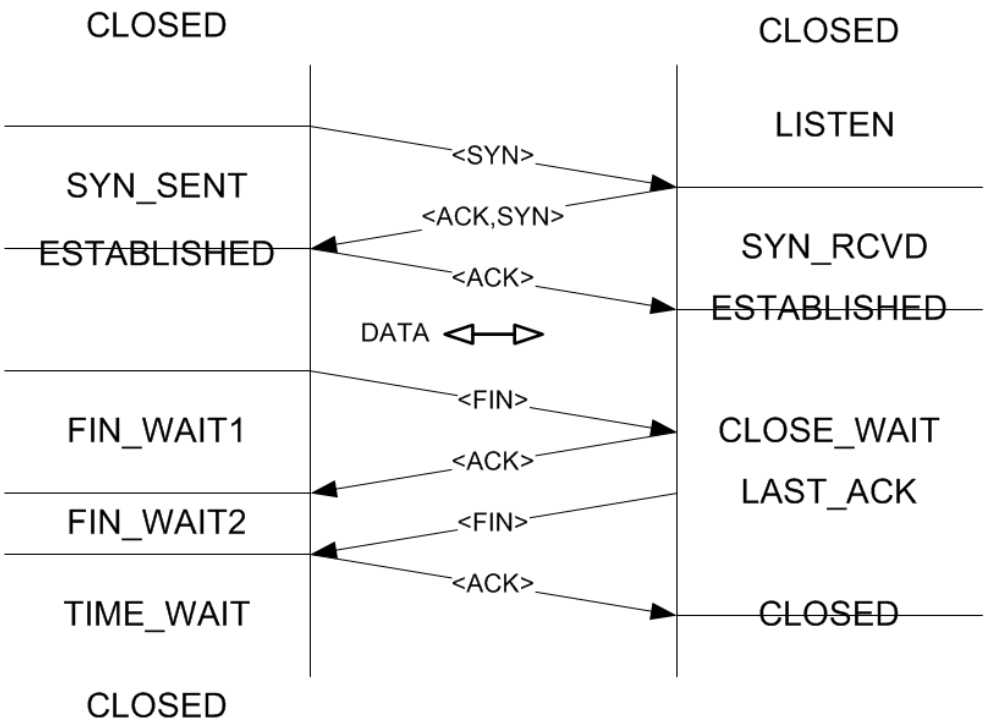
TCP State Transition Diagram



Ingevuld:



Tijdsas:



In het begin zijn zowel de server als client CLOSED. De server gaat dan beginnen luisteren naar inkomende berichten en gaat dus in de LISTEN status. Als de client een <SYN> stuurt naar de server zal de client zelf in SYN_SENT status gaan. Als de <SYN> ontvangen is bij de server zal deze in de SYN_RCVD status gaan van zodra hij de <ACK,SYN> verstuurd heeft. Als dit toekomt bij de client zal deze in de ESTABLISHED status gaan, want dus betekent dat deze aanneemt dat hij verbonden is met de server. Hij zal ook nog een <ACK> naar de server sturen, die dan ook in de ESTABLISHED status gaat gaan.
Vervolgens wordt er data verstuurd tussen de client en de server.
Als de connectie afgesloten mag worden, zal de client een <FIN> sturen naar de server, client gaat na het verzenden in FIN_WAIT1 status. Als de server de <FIN> ontvangt zal deze in de CLOSE_WAIT status gaan. Hij zal antwoorden met een ACK zodat de client in FIN_WAIT2 kan gaan. Als de server ook klaar is om af te sluiten, zal deze zelf ook een <FIN> sturen de client. De server gaat na het verzenden in LAST_ACK. Als de client de <FIN> ontvangt gaat deze in TIME_WAIT, want hij moet nog even wachten voor als er nog iets gestuurd zou worden van de server en daar nog op moet antwoorden. Als de server de laatste <ACK> ontvangen heeft gaat deze in CLOSED status. Als de timer aan de clientzijde uiteindelijk afgelopen zal zijn, zal deze ook in CLOSED status gaan.

4. Bespreek: acknowledgement, timeout retransmit, duplicate reception, piggybacking, delayed ack, accumulated ack, selective retransmit, fast retransmit, retransmission timer, retransmission time-out, measured round trip time.

- ACK: bevestigingsbit in de TCP-header, geeft aan dat de vorige segmenten correct ontvangen zijn
- timeout retransmit: de zender houdt een timer bij per verzonden segment, als die afloopt en er is op dat moment nog geen ACK ontvangen, zal de zender het segment opnieuw versturen
- duplicate reception: het kan gebeuren dat de ontvanger 2x eenzelfde segment ontvangt (bvb door een ACK die verloren gegaan is of te laat aankwam). De tweede keer wordt het segment gewoon genegeerd
- piggybacking: een ACK die gestuurd wordt samen met een ander segment dat naar dezelfde

ontvanger moet

- delayed ACK: wanneer de ontvanger de ACK niet onmiddellijk stuurt maar nog even wacht
- accumulated ACK: de ontvanger stuurt 1 ACK voor meerdere segmenten
- selective retransmit: wanneer een reeks segmenten doorgestuurd werd en de ontvangen ACK geeft aan dat er ergens een segment verloren ging, dan zal enkel dat segment opnieuw verzonden worden (men gaat er dus van uit dat alle volgende segmenten wel aangekomen zijn)
- fast retransmit: wanneer de ontvanger merkt dat het doorgestuurde segment een volgnummer heeft dat hoger is dan verwacht (niet in volgorde), dan gaat hij er van uit dat de tussenliggende segmenten verloren gingen en zal hij onmiddellijk vragen om die opnieuw te versturen => na triple duplicate ACK.
- retransmission timer: timer bijgehouden door de zender per verzonden segment, wanneer voor dat segment niet binnen een vastgelegde tijd een ack ontvangen wordt, zal de zender het segment opnieuw verzenden
- retransmission time-out: de time-out die optreedt bij bovenstaand puntje
- measured round trip time: de RTT van het laatst verzonden segment (vanaf verzenden van de data tot ontvangen van de ack)

5. Hoe berekent men de RTO? En hoe meet men de round trip time M?

- RTO = Retransmission Time-Out
- M = gemeten RTT (tijd tussen verzenden segment en ontvangen van ack)
- RTT = $\alpha\, RTT + (1-\alpha)\, M$ (=EWMA =Exponential Weighted Moving Average)
- $D = \beta\, D + (1-\beta)\,|RTT - M|$ (= afwijking in RTT)
- RTO = RTT + 4 D

6. Bespreek het principe van flow control in TCP. Leg in detail uit a.d.h.v. diverse “windows”. Waarom wordt flow controle gebruikt ?

Receive window (ontvanger): het aantal die aan ontvangerszijde ontvangen kunnen worden. Als applicatielaag bijvoorbeeld niet op tijd kan verwerken worden segmenten gebufferd.

Send window (zender): receive window MIN het aantal segmenten dat al verstuurd zijn, maar waarvoor nog geen ACK ontvangen is.

Flow controle wordt gebruikt om er zeker van te zijn dat de segmenten ontvangen zullen worden aan de ontvangerszijde en ze verwerkt kunnen worden. Als we bijvoorbeeld te veel segmenten tegelijk zouden doorsturen, zodat de ontvanger deze niet op tijd kan verwerken/opslaan, zullen er pakketten verloren gaan (wat we dus willen vermijden met flow control).

Bij flow controle komt het er op neer dat er maar evenveel segmenten zullen verzonden worden als er vrije plaatsen zijn in het send window. Het receive window wordt per ACK van de ontvanger aangepast naar de waarde die bij de ACK zat. Om op die manier te kunnen bepalen hoeveel segmenten er doorgestuurd moeten worden. Bij flow controle wordt er rekening gehouden met de buffers van de transportlaag naar de applicatielaag.

7. Bespreek het principe van congestion control in TCP. Waarom wordt dit gebruikt ?

Congestion window (zender): verhoogt met 1 voor elke ACK dat ontvangen wordt

Congestion controle houdt rekening met hoeveel verkeer het netwerk aan kan, deze kijkt namelijk naar de buffers van de netwerklaag.

8. Bespreek het verband tussen : send window, receiver window, congestion window

Het send window wordt bepaald door het minimum van het receive window en het congestion window te

nemen en daar de reeds verstuurde elementen die nog geen ACK teruggekregen hebben af te trekken.

9. Hoe wordt congestion gedetecteerd en wat is de reactie (geen details)?

Als er segmenten verloren gaan door congestion zullen er duplicate ACK's sturen. Als er duplicate ACK's toekomen kan het congestion window verlaagd worden om er zeker van te zijn dat er niet te veel verstuurd wordt.

10. Bespreek het principe van slow start en congestion avoidance.

Bij slow start komt het er op neer dat het congestion window begint bij 1 en telkens verhoogd wordt met elke ACK. Op die manier zal er geleidelijk aan meer verstuurd kunnen worden. Als het congestion window groter wordt dan een bepaalde treshold, zal de status naar congestion avoidance veranderd worden. In de praktijk komt het bij slow start dus neer op het verdubbelen van het congestion window bij elke ACK.
Bij congestion avoidance komt het neer op verhogen van het congestion window met 1 per RTT. Dit om te zorgen dat het congestion niet zo stijl meer stijgt zoals bij slow start.

11. Bespreek het principe van fast retransmit en fast recovery

Als er een segment toekomt bij de ontvanger dat niet in volgorde is, dus een vroeger element nog niet is toegekomen. Dan zal er onmiddellijk een ACK gestuurd worden dat het segment opnieuw gestuurd moet worden. Ook al moet er normaal gezien een delayed ACK gestuurd worden op een later tijdstip. Er moeten minstens 3 duplicate ACK's ontvangen worden voor hetzelfde segment voor er fast retransmit uitgevoerd zal worden. Als er nog geen 3 zijn, zal de duplicate ACK gewoon genegeerd worden en behandeld worden als een gewone ACK.
Als er 3 duplicate ACK's van hetzelfde segment ontvangen zijn, zal het congestion window gehalveerd worden (blijft wel minstens 1). Op die manier zal het versturen van segmenten gehalveerd worden en zal het terug even duren voor er nog duplicate ACK's gekregen worden.

12. Leg uit : AIMD

Manier van congestion control:
AIMD: Additive Increase Multiplicative Decrease
Het congestion window wordt additief vergroot, dus telkens verhogen met 1 per RTT
Als er een pakket verloren gaat, wordt het congestion window 'multiplicatief' verlaagd: de grootte van het congestion window wordt gedeeld door twee.

13. Leg uit hoe TCP "fairness" ondersteunt. Geef een voorbeeld hoe men dat kan omzeilen

TCP ondersteunt "Fairness" door ervoor te zorgen dat als er K TCP-sessies zijn die langs dezelfde bottleneck moeten, ze elk ongeveer evenveel (R/K) van de bandbreedte R krijgen. Dit valt te omzeilen door meerdere TCP-sessies te starten, of UDP te gebruiken.
zie ook voorbeeld slides hoe beide lijnen elkaar naderen.

14. Waarvoor worden TCP en UDP gebruikt ? Geef enkele voorbeelden

UDP wordt gebruikt als de snelheid belangrijker is dan de kwaliteit, bijvoorbeeld bij multimedia streams. UDP wordt ook gebruikt bij DNS, voor de meeste andere internet protocollen wordt TCP gebruikt, zoals HTTP, FTP, mail, … UDP is niet fair en wordt gebruikt door applicaties die niet afgeremd willen worden door congestion windows.

Hoofdstuk 4 : Netwerklaag

1. Bespreek de verschillende IP-adresklassen. Geef enkele speciale adressen

(Source)

Klasse	Range van eerste octet	#netwerken	#hosts/netwerk	Doel
A	0-127	126 (2^7)	16M-2 (2^24-2)	Voor netwerken met een groot aantal hosts *
B	128-191	16K (2^14)	64K-2 (2^16-2)	Voor netwerken met een gemiddeld aantal hosts
C	192-223	2M (2^21)	256-2 (2^8-2)	Voor kleine LANs
D	224-247			Voor IP multicasting
E	248-255			Reserved for “experimental use”

Speciale adressen:

- Notatie van een netwerk: Het host-gedeelte allemaal nullen
- 0.0.0.0: Deze host op dit netwerk, wordt gebruikt voor booting, enkel toegestaan als source
- 0.0.X.Y: Een host op dit netwerk, wordt gebruikt voor booting, enkel toegestaan als source
- 127.X.Y.Z: loopback interface, gebruikt om te debuggen
- 255.255.255.255: Broadcast naar alle hosts op dit netwerk, enkel toegestaan als destination
- X.Y.255.255: Broadcast naar alle hosts op remote netwerk, enkel toegestaan als destination
- Private ranges: 10.0.0.0-10.255.255.255; 172.16.0.0-172.31.255.255; 192.168.0.0-192.168.255.255; gebruikt door netwerken die niet verbonden zijn met het internet (bvb private netwerken)

2. Bespreek het principe van direct connected networks en subnetworks

Hosts die verbonden zijn zonder dat er een router tussen hen in zit, vormen een direct connected network. Als voor een netwerk de IP adressen worden opgesplitst in [Network | Subnet | Host] spreekt men van een subnetwerk (daar kunnen dus wel routers in zitten).

3. Bespreek subnet adressering. Geef een voorbeeld

Als we bijvoorbeeld 157.193.100.10 hebben als IP-adres. Dan zal 157.193 wijzen op het netwerk van de UGent, de 100 wijst op het subnetwerk van de UGent (bijvoorbeeld telin) en dan 10 wijst op de exacte pc (host) binnen dat subnetwerk. Om de host te weten uit het subnetwerk hebben we de mask nodig, dat bestaat uit een aantal enen vooraan (hier 24), gevolgd door allemaal nullen daarachter (hier 8 nullen). Het aantal enen wijst op het aantal bits genomen moeten worden om het (sub)netwerk te identificeren. Alle bits die er achter nog komen identificeren dan de host. De mask zou hier zijn: FF.FF.FF.00

4. Bespreek CIDR. Geef een voorbeeld

CIDR (Classless Inter-Domain Routing) maakt geen gebruik meer van klassen zoals er vroeger wel gedaan werd. CIDR geeft achter het IP adres nog op hoeveel bits er nodig zijn om het netwerk adres aan te duiden. Bv: 157.193.16.0/24 wil zeggen dat de eerste 24 bits van het adres 157.193.16.0 aanduiden welk netwerk het is, de andere overige bits kunnen gebruikt worden om de verschillende hosts aan te duiden. CIDR gaat de mogelijke ranges altijd in 2 delen opdelen en deze eventueel ook nog eens verder onderverdelen zodat de range zo goed mogelijk aansluit.

5. Bespreek het verschil tussen routing en forwarding

Bij routing wordt de route effectief nog bepaald, terwijl bij forwarding gewoon doorgestuurd wordt zoals het al eerder werd vastgelegd. Routing gebeurt aan de hand van de routeringstabel bv van een router, terwijl

een switch ook al zal forwarden.

6. Bespreek de verschillende velden van een IP-datagram (het datagram zelf wordt gegeven op het examen)

<table>
<tr><td>4 bit version</td><td>4 bit header length</td><td>8 bit type of service</td><td colspan="2">datagram length in bytes</td></tr>
<tr><td colspan="3">16 bit identifier</td><td>3 bit flags</td><td>13 bit fragment offset</td></tr>
<tr><td colspan="2">8 bit TTL</td><td>8 bit protocol</td><td colspan="2">16 bit header checksum</td></tr>
<tr><td colspan="5">32 bit source IP address</td></tr>
<tr><td colspan="5">32 bit destination IP address</td></tr>
<tr><td colspan="5">options (if any)</td></tr>
<tr><td colspan="5">data</td></tr>
</table>

- Versie: IPv4 of IPv6
- header lengte: grootte van header in 32 bit woorden
- Type of Service (ToS): 3 bits prioriteit, 4 bits delay, bandwidth.. en 1 ongebruikt bit
- datagram lengte: grootte van volledige IP datagram in bytes (inclusief header)
- identifier : unieke identificatie van het fragment
- vlaggen en fragmentatie offset: gebruikt bij fragmenteren van IP datagrammen
- TTL (time to live): Geven een maximale houdbaarheid aan het datagram (elke keer het datagram door een router passeert wordt het met 1 verminderd, TTL=0: pakket wordt gedropped)
- Protocol: geeft soort protocol (6=TCP, 17=UDP, 1=ICMP)
- header checksum: foutdetectie datagram (header opdelen in 16 bit getallen, bitsgewijs optellen en resultaat vergelijken met checksum)
- source en destination IP: van waar komt het pakket en waar moet het naartoe
- opties (te volgen route etc) en data: spreken voor zich

7. Bespreek fragmentatie

Netwerk links hebben een maximum grootte (MTU) voor IP datagrammen, wanneer die grootte overschreden wordt, moet het datagram worden opgesplitst. Pas op het eindpunt worden al deze datagrammen weer samengevoegd. Het opsplitsen gebeurt door elk onderdeel een eigen header te geven waar de fragflag op 1 staat. Ook is er een offset waardoor je de volgorde van de pakketten kan achterhalen.

8. Wat is ICMP ? Geef een voorbeeld bij het gebruik in een redirect en traceroute

ICMP: Internet Control Message Protocol, zorgt voor controle berichten en foutmeldingen.

Redirect: Als een pakket een kortere route kan volgen dan de huidige route, dan zal de router het pakket correct doorsturen, en ondertussen ook een ICMP redirect message versturen naar de zender om die duidelijk te maken dat er een kortere router is. Als bvb A een pakket naar C stuurt via router B, en B merkt dat A en C op hetzelfde subnet zitten, dan zal B een redirect message naar A sturen. Dit is een type 5 ICMP bericht.

Bij een traceroute wordt de route naar een bepaalde bestemming gezocht. Dit gebeurt door pakketten te verzenden met eerst TTL=1, dan TTL=2, enzovoort tot de bestemming wordt bereikt. Wanneer een host een pakket ontvangt met TTL == 1, dan wordt het pakket gedropped en een ICMP Time Exceeded terugestuurd. Aan de hand van al deze ICMP pakketten die teruggestuurd worden kan de zender de route

bepalen.

9. Bespreek NAT. Geef een voorbeeld. Wat is large scale NAT ?

NAT = Network Address Translation

Dit wordt gebruikt om meerdere hosts in een lokaal netwerk toegankelijk te maken met slechts 1 extern IP adres. De IP's die je toekent in het netwerk spelen geen rol voor de buitenwereld, waardoor je van ISP kan veranderen zonder IP's te moeten aanpassen. Dit is ook een beveiligingsvoordeel.

De NAT router zal elke uitgaande connectie mappen op een andere poort. Wanneer de (interne) host met IP 192.168.0.2 via poort 1234 een connectie wil opzetten met een webserver, zal de router dit vertalen naar de buitenwereld als het externe ip dat een connectie opzet met een willekeurige andere poort. Dit wordt bijgehouden in de NAT tabel, waardoor antwoorden op deze verbinding terug naar de juiste host kunnen gestuurd worden.

Een large scale NAT krijg je wanneer de ISP dit systeem ook toepast, zo krijg je de lokale NAT en de ISP die beide vertalingen gaan doen. Dit is vooral handig voor landen waar maar weinig IP adressen voor handen zijn. (wordt ook wel NAT444 genoemd)

Nadelen:

- inbreuk tegen het lagensysteem: routers mogen in principe maar tot de netwerklaag gaan.
- beperkt aantal poorten (kan problemen opleveren bij large scale)
- mogelijk dubbel gebruik private IP range bij meervoudige NAT

10. Bespreek DHCP. Geef een voorbeeld

DHCP = Dynamic Host Configuration Protocol

Doel: Zorgen dat een host dynamisch een IP-adres toegewezen krijgt op het moment dat hij toegevoegd wordt, waardoor adressen hergebruikt kunnen worden en mobiele gebruikers makkelijker aan netwerken toegevoegd kunnen worden.

Dit gebeurt door een 'DHCP discover' message van de nieuwe host, waarop de DHCP server antwoordt met een 'DHCP offer'. De host kan het IP-adres dan aanvragen met een 'DHCP request', waarna de DHCP-server een 'DHCP ACK' stuurt en het IP-adres is toegekend.

11. Wat is een AS ? Geef 3 types (waarom is het belangrijk een onderscheid te maken)

Een AS is een autonomous system. Het internet bestaat uit verschillende ASs. Zo'n AS is dus een netwerk dat op zich functioneert, en verbonden is met andere ASs

- Stub AS: small corperation, klein AS in een bedrijf. Verbonden met andere AS via 1 connectie
- Multihomed AS: large corporation, geen transit, verbonden met meerdere andere AS's
- Transit AS: provider, verbind verschillende AS met elkaar.

Het nut van de verschillende types ligt in het feit welk routeringsprotocol er gevolgd moet worden (klein netwerk, groot netwerk, BGP,...)

12. Bespreek het verschil tussen intra- en inter-AS routering

- Intra-AS: Binnen 1 AS: de administrator is zelf verantwoordelijk voor het protocol voor de routering, dit kan per AS verschillend zijn, bvb RIP of OSPF.
- Inter-AS: Tussen de verschillende AS, dus 1 algemeen protocol nodig: alles gebeurt via BGP (Border Gateway Protocol). Soort lijmmiddel tussen verschillende AS.

13. Bespreek het principe van distance vector en link-state routering. Geef een voorbeeld voor beide strategieën

Distance vector routering: bv. RIP (= Routing Information Protocol)

=> Buren wisselen distance vectors uit, aan de hand waarvan ze hun afstand tot de andere routers kunnen bepalen. Dit gebeurt om de 30 seconden a.d.h.v. advertisements. Om te voorkomen dat er tot

oneindig geteld kan worden, ligt het maximum aantal hops op 15 (16 is oneindig). Dit betekent dat dit protocol enkel geschikt is voor kleine netwerken. Hoewel het vrij traag is, wordt het vaak gebruikt.

Link-state routering: bv. OSPF (= Open Shortest Path First)
=> Elke router houdt een overzicht van het netwerk bij in een Link-state database, met behulp van Link-state pakketten die uitgewisseld worden. Hierbij wordt in de router zelf dan bv. Dijkstra toegepast op deze database om de kortste paden te bepalen. De voordelen van OSPF zijn onder andere dat er authenticatie gebeurt, er meerdere paden met dezelfde kost toegestaan zijn en de kost-metriek verschillend kan zijn naargelang de toepassing.

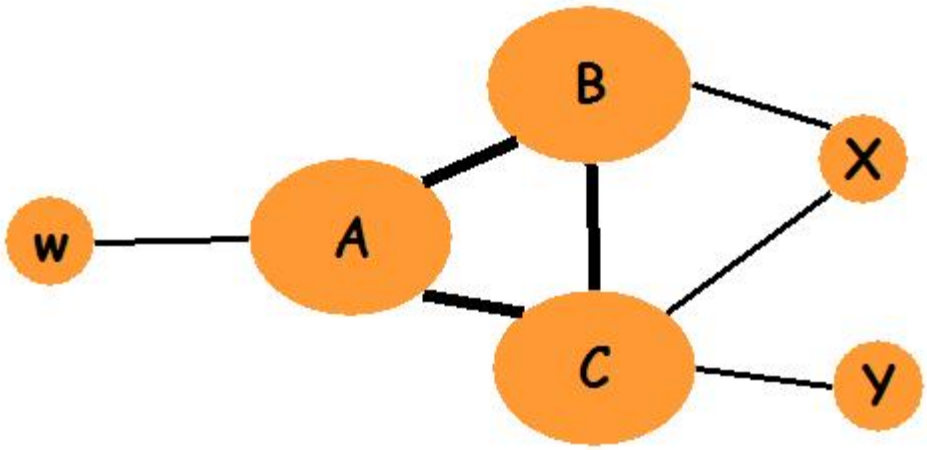
14. Bespreek hierarchical OSPF. Waarom is dat nuttig ?

Bij hierarchical OSPF deel je het netwerk op in areas, elke area voert intern OSPF uit, 1 backbone area bepaalt dan het pad tussen deze verschillende subnetwerken.
Er zijn 4 soorten OSPF-routers in het totale netwerk:

- interne routers: binnen een area
- area border routers: horen zowel bij een area als bij de backbone
- backbone routers: horen enkel bij de backbone
- boundary routers: routers die ook extern verbonden zijn (met andere AS'en) voeren dus BGP uit

Voordeel: minder rekenwerk voor het algoritme (minder nodes), maar ook beter voor beveiliging: buiten een area kan niemand zien hoe de indeling van de area eruit ziet

15. Bespreek een voorbeeld van BGP. Waarom heeft men I-BGP en E-BGP ?



- A, B en C zijn **transit AS's** (ISP)
- W, X en Y zijn **stub AS's** of **multihomed AS's**

- X meldt aan B en C dat het geen paden kent naar andere bestemmingen dan zichzelf (doet zich voor als een stub AS)
- B kent pad AW van A
    - meldt dit pad aan zijn klant X
    - meld dit pad niet aan ISP C
      reden: C zou dataverkeer aan W via B kunnen verzenden (CBAW) --> werk en kosten voor B terwijl de weg CAW gebruikt kan worden

BGP: de standaard om te routeren tussen verschillen Autonome Systemen
I-BGP : Interior-Border Gateway Protocol, werkt enkel met AS binnen hetzelfde subnetwerk

E-BGP: Exterior-BGP, werkt tussen verschillende subnetwerken.
Men werkt met een interne en een externe BGP om het werk te verdelen.

16. Wat is een AS-PATH ? Wat is een NEXT-HOP ?

Een AS-PATH is een pad van de ene AS naar de andere, waarbij alle tussenliggende ASs meegerekend worden. Dit zijn alle ASs waarlangs een advertisement gepasseerd is.
De NEXT-HOP is de router interface aan het begin van een AS-PATH.

17. Wat is policy based routing in BGP ? Geef een voorbeeld

Door policy based routing kan een router beslissen welke pakketten hij wil forwarden.

Hoofdstuk 5 : Datalinklaag

1. Bespreek de verschillende velden van een Ethernet frame (het frame zelf wordt gegeven op het examen)

preamble	frame delim	destination address	source address	type	data	padding	checksum

- 7 bytes, Preamble: gebruikt voor synchronisatie tussen ontvanger en verzender: 7 bytes : 10101010
- 1 byte, Frame delimiter: geeft start van frame weer: 10101011
- 6 bytes, Destination address: wereldwijd uniek adres (tenzij met mac spoof). De eerste 3 bytes zijn gefixed op een firma, de laatste 3 zijn vrij en worden per bedrijf ingevuld.
- 6 bytes, Source address: idem als Destination address
- 2 bytes, Type: het type data (vb ip frame)
- 0-1500bytes, Data
- 0-46 bytes, Pad: om de frame minimum lengte van 64 byte te bekomen
- 4 bytes, Checksum: spreekt voor zich

2. Bespreek het CSMA/CD principe

CSMA/CD = Carrier Sense Multiple Access / Collision Detection
Dit is een protocol dat bepaalt wanneer er pakketten verzonden mogen worden.
Als de bus vrij is, kan een host beginnen zenden, als de bus bezet is wacht de host. Eenmaal de bus terug vrij komt kan de host dan beginnen zenden.
Als twee hosts beginnen te zenden op een moment dat ze van elkaar nog niet weten dat de andere ook aan het zenden is, kan er een collision optreden. Op het moment dat ze deze collision opmerken stoppen ze met zenden, en wachten ze even. Hoe lang ze wachten is een random keuze, en het aantal keuzemogelijkheden vergroot iedere keer als terug een collision zou optreden, met een maximum van 1023 time slots.
Om te zorgen dat er geen collision ongedetecteerd voorbij kan gaan, moet de frame lengte groter zijn dan 2 maal de propagatietijd. Hierdoor wordt collision detection gegarandeerd.

3. Waarom gebruikt men bij Ethernet een minimale framelengte van 64 bytes ?

Hierdoor werkt collision detection nog goed bij een RTT van 51,2 µsec, wat aan 10 Mbit/s de aanvaardbare lengte van maximum 5 km kabel geeft. Moest de framelengte korter zijn, zou de propagatietijd ook kleiner moeten zijn om zeker te zijn dat collisions gedetecteerd worden, waardoor de afstand waarover het signaal verzonden kan worden ook alsmaar kleiner wordt.

4. Wat is exponential back-off (bespreek)

Dit betekent dat het aantal keuzemogelijkheden voor de tijd die ze wachten na een collision steeds vergroot. Na de eerste collision wordt random gekozen tussen 0 of 1 tijdslot (= 51,2 µsec), als dit opnieuw een collision oplevert wordt het aantal mogelijkheden exponentieel vergroot naar 0, 1, 2 of 3 tijdslots, enzoverder, tot een maximum van 1023 tijdslots.

5. Bespreek ARP bij Ethernet

Als host A wil verbinden met een andere host B. Host A weet nog niet waar host B zich bevindt, dus gaat host A een ARP broadcast doen om dit te bepalen. De broadcast bevat het IP adres van host B dat gekend is door host A. De broadcast zal over de routers heen uiteindelijk bij host B terechtkomen die zal antwoorden (met zijn MAC-adres) naar host A dat hij host B is.

6. Wat is een Ethernet hub ? En een Ethernet switch ?

Een ethernet-hub zit op de fysische laag en zendt alles wat hij binnenkrijgt voort via al zijn andere links.
Er gebeurt geen frame buffering of CSMA/CD. “Shared broadcast”
Een ethernet-switch zit op de datalink laag en kan ethernet frames bijhouden en verder zenden. Aan de hand van het MAC adres kan de switch selecteren naar welke link(s) het frame moet verder gezonden worden. Een switch is self-learning en moet niet geconfigureerd worden, en wordt niet gedetecteerd door hosts. “Point to point link”

7. Hoe worden de switchtabellen ingevuld ? En hoe worden ze gebruikt ?

Als de switch een frame ontvangt, kan hij de locatie van de zender invullen in zijn switchtabel. Als de switch de locatie van de bestemming nog niet kent, broadcast hij het bericht, anders kan hij het direct in de juiste richting sturen. Na verloop van tijd worden oude entries uit de tabel verwijderd.

8. Bespreek STP en geef een voorbeeld

STP = Spanning Tree Protocol
Om te voorkomen dat switches in een oneindige lus frames zouden blijven forwarden, wordt het Spanning Tree Protocol toegepast. Dit zorgt ervoor dat interfaces geblokkeerd worden, waardoor geen lussen in het netwerk voorkomen. Deze configuratie gebeurt automatisch: eerst worden alle poorten geblokkeerd, waarna een root switch gekozen wordt en met Kruskal een opspannende boom wordt gezocht, en aan de hand hiervan worden de poorten dan juist ingesteld. De root switch wordt gekozen aan de hand van het laagste Bridge ID.

9. Wat is een VLAN ? Bespreek twee types VLAN

VLAN = Virtual Local Area Network
Port Based VLAN (static): De poorten van een router worden gebruikt als VLAN (geen tags)
VLAN met Tags: de frames bevatten tag header met identificatie van VLAN (=VLAN-tagged frame) of de tag header bevat prioriteitsinformatie maar geen VLAN identificatie (=priority-tagged frame, VID=0)

frame filtered VLAN (filteren op basis van de inhoud van het frame bv: aparte VLAN voor ICMP)
ethernet adres gebaseerde vlan (dynamisch)

10. Geef een aantal voor- en nadelen van switches (versus routers)

+ Operaties in een switch zijn eenvoudiger en vragen minder processing
+ Switches zijn zelf-lerend en moeten niet geconfigureerd worden. Plug-and-play
- Al het verkeer wordt door de spanning tree geleid, ook als er eigenlijk meer bandbreedte beschikbaar is.
- Switches hebben geen bescherming tegen broadcast storms.

11. Bespreek PPP

PPP: Point-to-Point Protocol

Eén zender, één ontvanger, één connectie, op datalinklaag. Geen nood aan Media Access Control, expliciete MAC addressering. Er is foutdetectie (geen -correctie), geen flow controle, geen volgordewaarborg.
Mogelijkheid tot configureren van link + doorzenden status link

Hoofdstuk 7 : Beveiliging

1. Bespreek een aantal mogelijke aanvallen op het internet (en de bijhorende verdedigingen)

Mapping: port scan om via actieve poorten de actieve services te vinden. Door traffic op netwerk te registreren kan ongewenste activiteit bemerkt worden.

Packet sniffing: Trudy kan alle niet-geëncrypteerde data lezen. Door na te zien of poorten in luistermodus staan, hubs vermijden

IP Spoofing: Verkeerde informatie als source of destination adres meegeven aan IP pakket. Routers zouden moeten controleren of source en destination adres wel mogelijk zijn.

DoS: Denial of Service. Overspoel ontvanger met TCP pakketten. **DDoS** (Distributed DoS) als er meerdere zenders een ontvanger aanvallen . firewall, ids,...

2. Bespreek het principe van cryptografie met symmetrische sleutels

Alice en Bob hebben elk dezelfde symmetrische sleutels die ze gebruiken om gegevens te encrypteren/ decrypteren. Als Alice een bericht naar Bob wil sturen, zal ze het encrypteren met haar sleutel. Het versleutelde bericht wordt dan verstuurd naar Bob die het zal decrypteren met zijn sleutel. Trudy kan dan enkel de versleutelde informatie zien, en zonder de juiste sleutel is ze hier niets mee.

3. Bespreek het verschil tussen blok- en stroomversleuteling. Wat is CBC ?

Bij blokcodering zal een bepaald bericht opgedeeld worden in blokken die dan per blok versleuteld en doorgezonden zullen worden. Aangekomen bij de ontvanger zal het dan gedecodeerd worden per blok en terug samen gebracht worden.
Stroomcodering zal elke bit apart coderen.
Cipher-Block Chaining (CBC): elk blok dat nog niet gecodeerd is, zal eerst bitgewijs opgeteld worden met het vorige gecodeerde blok. Pas daarna zal het blok gecodeerd worden.

4. Bespreek het principe van cryptografie met openbare sleutels

Alice zal een bericht coderen met de publieke sleutel van Bob, het bericht doorsturen naar Bob die het dan zal decoderen met zijn private sleutel.

5. Bespreek het principe van digitale handtekening

Digitale variant van de handtekening, maakt het mogelijk om de eigenaar van documenten te verifiëren en is onvervalsbaar. Hier wordt de publieke sleutel gebruikt om te decoderen in plaats van te coderen. Op die manier moet diegene die het document echt gemaakt heeft minstens de sleutel van de zogezegde eigenaar hebben. Als Alice bijvoorbeeld een document wil digitaal tekenen met haar private sleutel. Dan weet Bob zeker dat als hij het bericht ontvangt en het kan decoderen met Alice haar publieke sleutel dat het bericht van Alice komt.

6. Bespreek KDC en CA

Key Distribution Center en Certification Authorities
Key Distribution Center zorgt er voor dat beide partijen aan geheime symmetrische sleutel geraken. Elke gebruiker van KDC krijgt een unieke sleutel die gebruikt wordt om te interageren met KDC. Als Alice bv een bericht aan Bob wil zenden, dan gaat Alice eerst vragen aan KDC om de geheime sleutel van Bob. Deze sleutel is gecodeerd met de unieke sleutel van Alice. Alice kan de sleutel van Bob dus decoderen en

vervolgens het bericht coderen met de sleutel van Bob en dan naar Bob sturen.
Bob moet zijn publieke sleutel bij CA registreren om zijn authenticiteit te bewijzen.

7. Bespreek principe e-mail encryptie

Als Alice een mail wil versturen naar Bob, dan zal ze een random sleutel aanmaken. Ze zal met deze sleutel het bericht coderen. De sleutel wordt dan ook nog eens gecodeerd met de publieke sleutel van Bob. Zowel het gecodeerde bericht als de versleutelde sleutel zal doorgestuurd worden naar Bob. Bob kan met zijn sleutel de random sleutel decoderen en daarmee dan het gecodeerde bericht van Alice decoderen.

8. Bespreek principe SSL ("toy example")

Client A wil beveiligde connectie met server starten. Client A stuurt aanvraag naar server B die antwoordt met de publieke sleutel van die server. Client A genereert een willekeurige sleutel die gecodeerd met de publieke sleutel van B doorgestuurd wordt. B kan deze sleutel decoderen met zijn eigen private sleutel. Van op dat moment kan data ge(de)codeerd worden met deze random sleutel.
SSL zit tussen transport- en applicatielaag.

9. Bespreek principe IPSec: twee modes, SA

De twee modes zijn: Transport mode en Tunneling mode.
Bij transport mode zijn de hosts op de hoogte van de IPsec bescherming, de tussenliggende routers niet.
Bij tunneling mode zijn de (end) routers op de hoogte van de IPsec bescherming, hosts niet.
SA: Security Association: virtuele connectie tussen de 2 hosts (of routers in tunneling mode)
IPsec zit boven de datalinklaag.

10. Bespreek pakketfiltering "packet firewall: stateless en stateful" en toepassingsgateway "application gateway"

Stateless kijkt niet of het klopt wat er in de TCP segmenten verstuurd wordt (je kan dus een ACK sturen zonder dat er TCP connectie bestaat). Bij Stateful zal er opgevolgd worden of er een connectie is en of het wel nut heeft om bepaald segment door te laten. Stateful vraagt dus meer werk, maar is veiliger.
Application gateway wordt ook wel soms "proxy server" genoemd. Het is een manier om voor de server geheim te houden van waar de eigenlijke data komt.

Hoofdstuk 10 : IPv6

1. Bespreek de verschillende types adressen

Unicast (one-to-one)

- Link-local address (1 per interface, aangemaakt bij booten met Interface ID): prefix FE80::/10
- Global Unicast address: prefix 2000::/3 + global routing prefix = /48
- Unique Local Address (ULA, for local communications): prefix FC00::/7

Multicast (one-to-many): prefix FF00::/8 + 4 flag bits + 4 scope bits
bvb: FF02::1 = alle nodes op dezelfde link
FF02::2 = alle routers op zelfde link

Anycast (one-to-nearest = one-to-one-of-many):
bvb. dichtsbijzijnde DNS-server op netwerk

2. Wat zijn de belangrijkste verschillen tussen een IPv4 en IPv6 header ? Waarom heeft men

die verschillen ingevoerd ?
De grootte is verdubbeld naar 40 bytes (aangezien zender/ontvanger elk al 16 bytes zijn) maar is nu vast van grootte (makkelijker te verwerken). Het versie veld bevat nu 6 ipv 4, ToS is hernoemd naar Traffic Class, ipv. Total Length wordt enkel nog Payload Length aangeduid (aangezien de header toch vast is van grootte). Protocol-veld is hernoemd naar Next Header en TTL naar Hop Limit.

Verdwenen velden: header length (staat toch vast), checksum (was duur om telkens te herberekenen en bij moderne fiber connecties komt corruptie nog maar zelden voor + opvangbaar door TCP en UDP), ID, flags en fragmentatie-offset (routers fragementeren geen pakketten meer!).
Toegevoegde velden: flow label om paketten te groeperen.

Er kunnen dus geen opties meer toegevoegd worden aan het IP pakket, in plaats daarvan dient men extensie headers toe te voegen waarnaar dan verwezen wordt vanuit Next Header. Voorbeelden van zo'n extensie headers zijn Routing headers, Fragment Headers. Afhankelijk van het type header worden deze enkel geïnterpreteerd door de bestemmeling of door sommige/alle tussenliggende routers (Hop-by-Hop en Routing headers).

3. Geef een voorbeeld van adresresolutie

Tegenhanger van ARP in IPv4 is Neighbour Discovery Protocol (NDP) dat gebruik maakt van ICMPv6. Maakt gebruik van lokale multicast adressen, link-local adressen en sollicited-node multicast adressen (multicast adres afgeleid van laagste 24-bits van het gezochte adres, prefix FF02::1:FF00:/104).

Wanneer host A het link-layer adres van host B wil te weten komen stuurt hij een Neighbour Sollicitation naar het sollicited-node-multicast-adres gebaseerd op het adres van B (hierbij kan ook gebruik gemaakt worden van het Ethernet multicast-adres dat begint met 33-33-...). Er wordt geantwoord met een Neighbour Advertisement in unicast.

4. Bespreek de verschillende autoconfiguratiestappen

1. Genereer een link-local address op basis van EUI-64 (MAC-adres) en wordt lid van de all-nodes en sollicited-node multicast groepen. Voer Duplicate Address Detection (DAD) uit en leg vervolgens het link-local address vast als voorkeursadres.
2. (stateless) Router sollicitation (via router multicast: FF02::2), elke router antwoordt met Router Advertisement met netwerk prefixes. Voor elke prefix die via autoconfiguratie ingesteld dient te worden genereert de host een nieuw adres en controleert of dit nog vrij is via DAD.
3. (stateful) Wanneer netwerk prefixes uit de Router Advertisement de Managed/Other flag hebben dient een adres via DHCPv6 verkregen te worden of kunnen er extra configuratiedetails daar opgevraagd worden.

5. Geef de basisprincipes die kunnen gebruikt worden bij een overgang van IPv4 naar IPv6

- **Dual-stack**: afhankelijk van de andere partij 'praat' de host IPv4 of IPv6. Dit vereist echter wel dat ook bvb. de ISP IPv6 ondersteunt en route. Een ander nadeel is dat alle configuraties dubbel moeten onderhouden worden (wat kan leiden tot conflicten/beveilingsproblemen), niet alle software is hiermee compatibel.
- **Tunneling**: gebruik maken van de bestaande IPv4 infrastructuur om IPv6 pakketjes ter versturen. Vereist een tunnel die verbonden is met zowel IPv4 als IPv6 internet.
- **Translation**: vertaal IPv4 adressen naar IPv6 adressen (en omgekeerd) door embedding/NAT-technieken, dit dient zo transparent mogelijk te gebeuren maar niet alles kan 1 op 1 met elkaar

gemapped worden.