

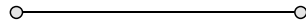
Tentamenstof deel B (hoofdstukken 5 tot en met 7) versie 10/12/02

gebaseerd op Tanenbaum "Computer Networks", 4e druk

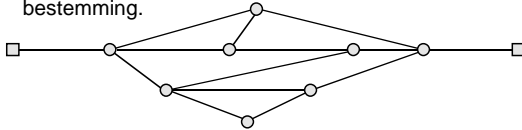
5. THE NETWORK LAYER

De functie van de netwerklaag

- De datalinklaag brengt *frames* van het ene punt van een transmissiekanaal naar een ander.



- De netwerklaag brengt *pakketten* van bron naar bestemming.

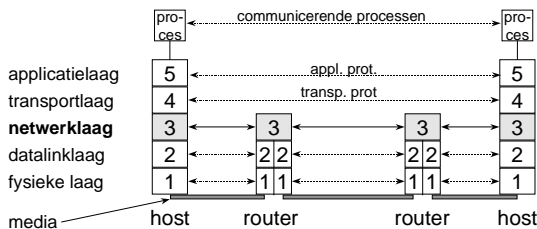


□ Station (*host*) ○ Knooppunt (*router*)

De netwerklaag:

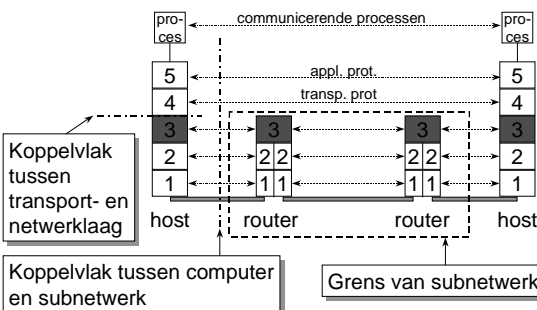
- heeft (direct of indirect) kennis van de topologie van het netwerk
- kiest geschikte paden door het netwerk
- kan rekening houden met belasting in het netwerk bij de keuze van de route
- moet ook een pad zoeken indien bron en bestemming in verschillende (gekoppelde) netwerken liggen.
- moet in geval van verschillende gekoppelde netwerken de problemen die ontstaan door die verschillen oplossen.

Plaats van de netwerklaag



- Netwerklaag de laagste laag die zich bezighoudt met transmissie van eind tot eind, zij doet dat echter stapsgewijs.
- Transportlaag en applicatielaag doen dat direct van eind tot eind.

Koppelvlakken (*interfaces*)



5.1. Network Layer design issues

5.1.1. Store and Forward Packet Switching

5.1.2. Services provided to the Transport layer

- De geleverde diensten moeten onafhankelijk zijn van de gebruikte technieken in het subnet
- De transportlaag moet afgeschermd zijn voor:
 - het aantal sub-netten tussen de eindstations
 - type van de subnetten
 - topologie van de subnetten
- De netwerkadressen moeten volgens een uniform nummerplan zijn dat zowel LAN's als WAN's beslaat.

Twee soorten diensten:

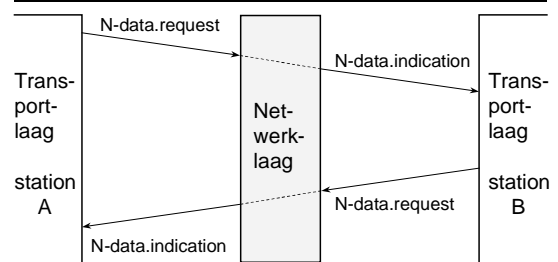
5.1.3 Verbindingsloze diensten:

- Geen foutbeheersing, stroombeheersing en volgordebewaking
- De complexiteit van de communicatie zit in de stations (in de transportlaag)
- Dienstprimitieven slechts:
 - N-DATA.request (opdracht van transportlaag in de zender, aan de netwerklaag data te verzenden)

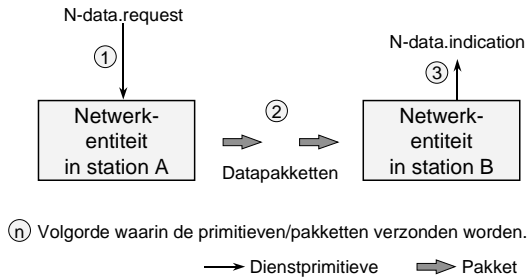
N-DATA.indication (melding aan de transportlaag in de ontvanger dat data ontvangen

is)

Verbindingsloze primitieven



Datapakketten en dienstprimitieven



5.1.4 Verbindingsgerichte diensten:

- Opzetten van een verbinding nodig
- Onderhandelen over de kwaliteit (en de kosten) van de dienst (b.v. snelle of nauwkeurige bezorging)
- Full-duplex communicatie
- Pakketten worden in volgorde afgeleverd
- Stroombeheersing tussen bron en bestemming
- soms: gegarandeerde bezorging, bevestiging daarvan door het subnet, mogelijkheid tot prioriteiten
- Verbreken van de verbinding na afloop van de communicatie
- De complexiteit van de communicatie zit in het subnet

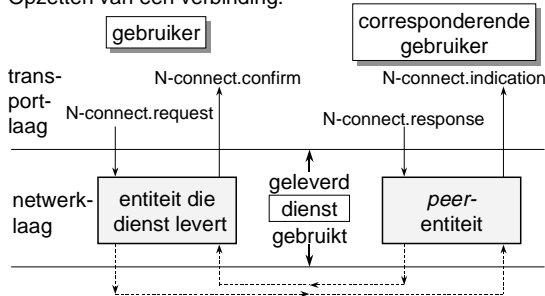
Behalve de dienstprimitieven N-DATA.request en N-DATA.indication voor het eigenlijke datatransport, zijn de dienstprimitieven voor het opzetten van een verbinding:

- N-connect.request
- N-connect.indication
- N-connect.response
- N-connect.confirm

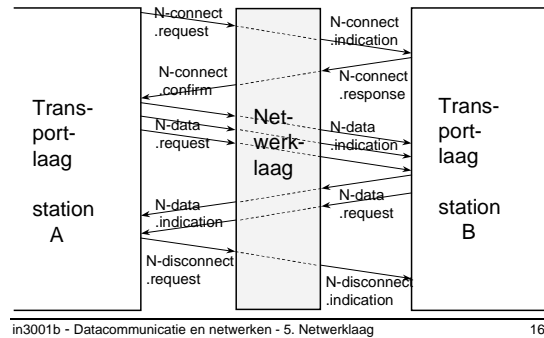
Zie de volgende transparant voor de uitgewisselde primitieven tussen transportlaag en netwerklaag (zie ook paragraaf 1.3.5.).

Dienstprimitieven verbindingsgericht

Opzetten van een verbinding:



Dienstprimitieven verbindingsgericht (2)



In de transparant direct hierboven zijn de uitgewisselde dienstprimitieven bij het opzetten van een verbinding nogmaals, maar dan op een andere manier, geschetst, bovendien worden de dienstprimitieven voor het uitwisselen van data en opheffen van de verbinding getoond:

- Voor het verbreken van de verbinding
- N-disconnect.request
- N-disconnect.indication

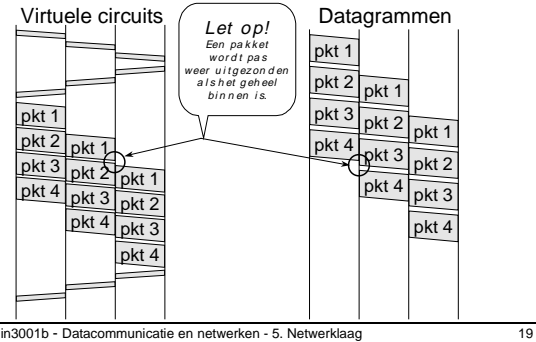
Internal organization of the Network layer

Er zijn twee manieren waarop het subnet intern kan werken:

- Virtuele circuits (het subnet werkt verbindingsgericht)
 - Route (full duplex) door het netwerk wordt bepaald, bufferruimte in de knooppunten en transmissiecapaciteit worden gereserveerd. Alle pakketten volgen dezelfde route. Vooral gebruikt voor verbindingsgerichte diensten.
- Datagrammen (het sub-net werkt verbindingsloos)
 - Pakketten worden van een volledig bestemmingsadres voorzien en zonder meer verzonden. Pakketten behorende tot een bepaald bericht kunnen in principe over verschillende routes gestuurd worden. Vooral gebruikt voor verbindingsloze diensten.

De twee soorten pakschakelen

De twee soorten pakketschakelen



LET OP

Bij pakketschakelen wordt een pakket eerst in zijn geheel ingelezen in een knooppunt en dan kan het pas weer verzonden worden naar het volgende. Elk knooppunt in de keten levert dus minstens een tijlvertraging op die gelijk is aan de tijd om het in te lezen. Zie bovenstaande figuur. Vergelijk nog even hoe het bij circuitschakelen gaat (fig. 2-35, blz. 132)

5.1.5. Comparison of virtual circuit and datagram subnets

- Fig. 5-2 kennen
- afweging lange adressen in de pakketten tegen tabelruimte in de knooppunten.
- afweging tijd voor opzetten van een verbinding tegen tijd nodig voor routing van elk pakket afzonderlijk.
- Bij de virtueel-circuit-techniek is congestiebeheersing gemakkelijker.
- Voor korte transacties kost het opzetten van een virtueel circuit relatief veel tijd
- Virtueel circuit kwetsbaarder voor storingen (de verbinding is dan verbroken en een nieuw virtueel circuit moet worden opgezet.)

Voor- en nadelen van de combinaties van geleverde diensten en werking van het subnet; zie tabel van figuur 5-3

5.2. Routing algorithms

5.2. Routing

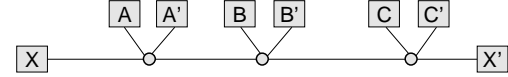
Eisen:

- Eenvoudig
 - Correct
 - Robuust
 - Stabiël
 - Rechtvaardig
 - Optimaal
- ← Kan tegenstrijdig zijn
- ← Kan tegenstrijdig zijn

Rechtvaardig <-> optimaal

Optimaliseren naar bijvoorbeeld

- Totale doorvoer
- Gemiddelde vertraging
- Optimale totale doorvoer kan onrechtvaardig zijn voor sommige gebruikers:



Voorrang geven aan A-A', B-B' en C-C' is goed voor de prestatie van het netwerk als geheel, maar niet voor X-X'.

Robuust tegen stabiël.

Robuustheid vraagt snel ingrijpen, dat kan instabiliteit in het netwerk veroorzaken, bijvoorbeeld doorlopend verleggen van routes. (pakketten kunnen in het ergste geval gaan zwerven door het netwerk of zelfs rondjes afleggen.)

Soorten routeringsalgorithmen

- Niet adaptief (Vast ingestelde routes)
- Adaptief
 - Gecentraliseerd (Een centrale computer beslist, vaak op grond van informatie van de knooppunten)
 - Geïsoleerd (Knooppunten beslissen op grond van eigen informatie, bijvoorbeeld wachrijlengten)
 - Gedistributeerd (Knooppunten beslissen mede op grond van informatie van naburige knooppunten)

5.2.1. The Optimality Principle

Begrijpen: het optimaliteitsprincipe en de sink tree van een bestemming

5.2.2 Shortest path routing

De werking van het algoritme kennen

5.2.3 Flooding

het begrip flooding en het nut hiervan kennen

5.2.4 Distance Vector Routing

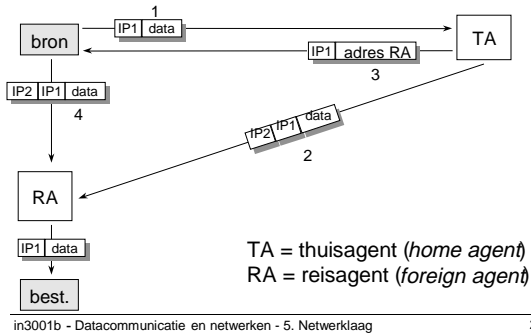
Het concept kennen. Aan de hand van ontvangen distance vectors een routing tabel kunnen construeren. Begrijpen waarom goed nieuws zich sneller verspreidt dan slecht nieuws.

5.2.5 Link State Routing

kennen tot en met einde p.363

5.2.6 t/m 5.2.8 behoren niet tot de tentamenstof

5.2.9. Routeren bij mobiele stations



5.2.9. Routing for mobile hosts

Een van de mogelijke routeringsmethoden is:

- Elke mobiel is geregistreerd daar waar zijn thuisbasis is in een database (TA);
- Indien een mobiel buiten het gebied van zijn TA komt, meldt het zich op een of andere manier aan bij de database voor bezoekers (RA) van dat gebied. Deze RA geeft zijn eigen adres door aan de TA.
- Wordt er een pakket gezonden (1) aan een mobiel dat niet in het gebied van zijn TA is, dan geeft zijn TA het door (2) aan de betreffende RA die het aflevert aan het mobiel. (hierbij wordt het oorspronkelijk pakket met header IP1 ingepakt in een pakket met header IP2)
- De verzender van het pakket krijgt het adres van de RA waar het mobiel op dat moment onder valt toegezonden van de TA (3).

De volgende pakketten worden direct naar de betreffende RA gezonden (4).

5.2.10 en 5.2.11 behoren niet tot de tentamenstof.

5.3. Congestion control algorithms

Congestie is verstopping van het netwerk, zodanig dat bij een toenemend aantal verzonden pakketten per tijdseenheid het aantal afgeleverde pakketten per tijdseenheid afneemt.

Oorzaken:

- Aanbod van pakketten bij een router is groter dan
 - de verwerkingscapaciteit van de router
 - de capaciteit van de uitgaande lijnen van de router.

Men kan onderscheid maken naar:

- Lokale congestie
 - aantal pakketten naar router groter dan aantal verzonden pakketten door de router
- Subnetcongestie
 - aantal pakketten naar subnet groter dan aantal pakketten vanuit het subnet

Congestie heeft de eigenschap zichzelf te versterken:

- Door congestie lange wachttijden => onnodige heruitzendingen door de stations
- weggooien van pakketten in overbelaste router => heruitzending door de stations
- Stroombeheersing is afstemmen van het aantal pakketten van de zender op het aantal dat de ontvanger kan verdragen. Soms wordt stroombeheersing gebruikt om congestie te bestrijden. Knooppunten in het netwerk zenden berichten naar andere knooppunten om ze te laten stoppen met zenden. Echter een knooppunt dat door stroombeheersing niet voldoende pakketten kwijt kan, raakt zelf vol.

5.3.1. General principles of congestion control
Congestiebeheersing kan ingedeeld worden naar de manier waarop het gebeurt: met of zonder terugkoppeling.

Congestiebeheersing zonder terugkoppeling
Beslissen zonder de toestand in de rest van het netwerk in acht te nemen dus op grond van lokale problemen. De gereedschappen zijn:

- "ruim" ontwerpen
- toegang van het verkeer beperken
- pakketten weggooien
- lokaal regelen van het verkeer op diverse plaatsen in het net

Congestiebeheersing met terugkoppeling kan onderverdeeld worden in drie stappen:

1. Bewaken van het net om te meten waar en wanneer congestie optreedt;

2. Informatie doorgeven naar de aangrijpingspunten voor de regeling van het verkeer in het net;
 3. Correctie van de werking van het netwerk om het probleem te verhelpen
- Problemen bij congestiebeheersing met behulp van terugkoppeling:

- Het stabiel houden van het net is vaak niet eenvoudig.
 - De signaleringspakketten zorgen voor extra verkeer juist als het net al zwaar belast is.
- De terugkoppeling kan op twee manieren:
- **Expliciet**
Pakketten die congestie melden naar de punten sturen waar ingegrepen moet worden.
 - **Impliciet**
Het ingrijpen wordt afgeleid uit lokale waarneming van langskomende pakketten

5.3.2. tot en met 5.4 behoort niet tot de tentamenstof

5.5 Internetworking

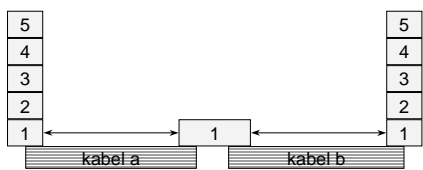
5.5.1 How Networks Differ

Van de 12 verschillen van fig 5-43 moet je er toch wel 9 kunnen noemen op het tentamen.

5.5.2 How networks can be connected

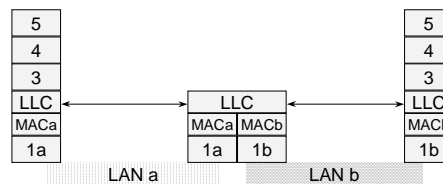
Vijf soorten van koppelingen bestaan ingedeeld naar de laag waarop gekoppeld wordt:

Repeater (koppelen op laag 1)



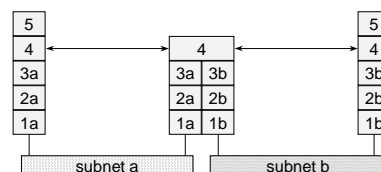
Koppelt kabelsegmenten.
Leest signalen in, reconstrueert de bits en zendt ze door.
Compenseert verzwakking van het signaal bij lange lijn.

Brug (koppelen op laag 2)



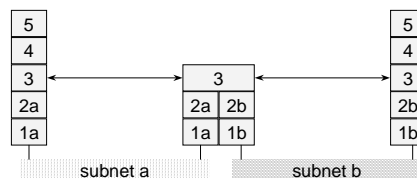
Koppelt LAN's.
Leest frames (DL-PDU's) in, slaat ze op en zendt ze door.

Transport gateway (koppelen op laag 4)



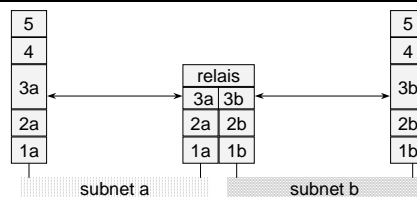
Koppelt netwerken die verschillend transportprotocol hebben.
Leest T-PDU's in, slaat ze op en zendt ze door.

Routers (koppelen op laag 3)



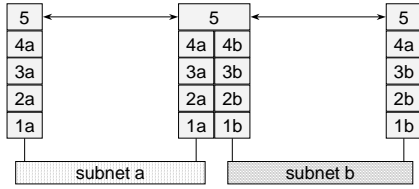
Koppelt netwerken met een zelfde netwerkprotocol en gedeelde adresruimte.
Leest pakketten (N-PDU's) in, slaat ze op en zendt ze door.

Multiprotocol-routers (koppelen op laag 3)



Koppelt netwerken met een verschillend netwerkprot.
Leest pakketten (N-PDU's) in, slaat ze op, zet ze om en zendt ze door.
(relais: omzetten, adressen vertalen, doorzenden en routeren)

Applicatie gateway (koppelen op laag 5)



Leest A-PDU's in, slaat ze op en zendt ze door.
Bijv. doorgeven van E-post tussen netwerken die niet een gezamenlijk netwerk- of transportprotocol hebben.

5.5.3. Concatenated Virtual Circuits

Principe kennen

5.5.4. Connectionless Internetworking

Principe kennen

5.5.5. Tunneling

Principe kennen zie fig 5-47.

IP-pakket wordt eerst verpakt in een ethernet frame, wordt ontdaan van het frame en verpakt in een pakket van de WAN. Het wordt vervolgens ontdaan van de *header* van het pakket van het WAN en weer verpakt in een ethernet frame.

5.5.6 Internetwork Routing

Vershillende niveaus van routerin in een internet kennen.

5.5.7. Fragmentation

Weten waarom fragmentatie nodig kan zijn.
Verschil weten tussen transparante en niet transparante fragmentatie.
De voor en nadelen van de beide methoden kennen.

5.6. The network layer in the Internet

Het netwerkprotocol van het Internet is IP (Internet Protocol)

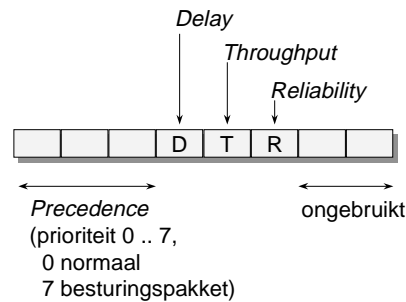
5.6.1. The IP protocol

Het boek is verwarrend voor wat betreft de benaming van de IP-pakketten. In dit hoofdstuk wordt het een datagram genoemd maar in hoofdstuk 6 een pakket. We zullen aannemen dat IP-PDU (IP-Protocol Data Unit), datagram en pakket synoniemen zijn dus:

- pakket = datagram =
= IP-PDU (*Internet Protocol Data Unit*)

De *header* van een IP-PDU (fig. 5-45) goed kennen, echter passief, dat wil zeggen gegeven de figuur, kunnen uitleggen waarvoor de verschillende velden dienen.

- *Version*
Versie van het IP-protocol waarbij het pakket hoort.
- *IHL*
Lengte van de *header* van het pakket in 32-bit woorden
- *Type of service* (wordt genegeerd)



-
- *Total length*
Lengte van het pakket (*header* + data)
- *Identification*
Identificatienummer van het pakket waartoe een pakketfragment behoort.
- *DF (Don't Fragment)*
Pakket mag niet gefragmenteerd worden. (Fragmenten zijn pakketten die oorspronkelijk samen een groter pakket vormden.)
- *MF (More Fragments)*
Meer fragmenten volgen; slechts het laatste fragment heeft dit bit niet "op" staan.
- *Fragment Offset*
Plaats van het fragment in het oorspronkelijke pakket
- *Time to live*
Teller die de levensduur van het pakket bijhoudt. In de praktijk telt hij af van een door de verzender ingesteld getal tot nul, waarbij elk knooppunt het getal met één verlaagt; stappenteller (*hop counter*).
- *Protocol*
Geeft aan met welk transportprotocol samengewerkt wordt (TCP, UDP,...)
- *Header checksum*
Controlesom over de gehele *header*.
- *Source address* en *destination address*.
Adressen van de verzender en de bestemming

- *Options*
 - *Security*
Geeft aan hoe geheim het bericht is.
Wordt in de praktijk genegeerd
 - *Strict source routing*
Geeft het complete te volgen pad, dat wil zeggen de complete lijst van adressen van routers.
 - *Loose source routing*
Geeft een aantal *routers* die in die volgorde aangedaan moeten worden.
 - *Record route*
Het adres van de *routers* die aangedaan zijn, wordt aan het *Options* veld toegevoegd.
 - *Timestamp*
Adres + tijdstip van de routers die aangedaan zijn; wordt toegevoegd

5.6.2. IP addresses

IP-adressen zijn 4 byte lang

- Klasse A (grote netwerken):
stationsadres 3 byte
netwerkadres 1byte - 1 bit
- Klasse B (middelgrote netwerken):
stationsadres 2byte,
netwerkadres 2byte - 2bit
- Klasse C (kleine netwerken):
stationsadres 1 byte,
netwerkadres 3byte - 3bit
- Klasse D (multicast): 4 byte - 4bit
- *Dotted decimal notation*
- Mogelijkheden om aan te geven:
 - Dit netwerk of dit station:
nullen in het betreffende veld
 - Omroepbericht (*broadcast message*):
enen in het betreffende veld
 - *Loopback* (127)
Bericht blijft binnen station maar wordt behandeld als binnengekomen bericht.
- Subnets
Principe kennen
- *subnet mask*
- CIDR (Classless InterDomain Routing)
principe kennen
- NAT (Network Address Translation)
principe kennen

5.6.3. Internet Control Protocols

The Internet Control Message Protocol

Het protocol wordt door stations en routers gebruikt. De meldingen van het protocol kunnen in vijf groepen verdeeld worden:

Groep	Bericht
Foutmeldingen	Destination unreachable Time exceeded Parameter problem
Congestiebeheersing	Source quench
Routeverandering	Redirect
Toetsen van bereikbaarheid	Echo request Echo reply
Meten van prestatie	Timestamp request Timestamp reply

De bovenstaande vijf groepen kennen. Van de erbij behorende berichten de aard kunnen noemen maar de precieze naam wordt niet gevraagd.

- The Address Resolution Protocol
Principe kennen.
- The Reverse Address Resolution Protocol
Principe kennen.

5.6.4. - 5.6.6

Behoort niet tot de tentamenstof.

5.6.7. Mobile IP

Doorlezen

5.6.8. IPv6

In de toekomst

- gebrek aan adressen
- veel draadloze mobile stations

Veranderingen t.o.v. IPv4

1. Langere adressen
(meer stations mogelijk)
2. eenvoudigere *header* (7 i.p.v. 13 velden)
(snellere verwerking in de routers)
3. Betere ondersteuning van *options* (enkele oude standaardvelden zijn nu opties zodat de standaard *header* korter kon worden)
4. Beveiliging beter
5. Meer mogelijkheid voor specificeren van de soort dienst (*type of service*)

Velden passief kennen:

- *Version*
- *Priority*
0 .. 7 verzendingen die vertraagd mogen worden in geval van congestie
- 8 .. 15 tijdgebonden verkeer (audio en video)

- *Flow Label*
Soort pseudo-verbinding om bijvoorbeeld bandbreedte te reserveren
- *Payload length*
- *Next header*
geeft aan welk optieveld volgt
- *Hop limit*
- *Source address*
- *Destination address*

Controversies (p471 - 473) doorlezen

5.7 Summary
doorlezen

6. TRANSPORTLAAG

6.1 Transport Service

6.1.1 Services aan bovenste lagen
Taak van de transportlaag
begrippen: Transport Entiteit, Transport adres, Netwerk adres, TPDU

6.1.2 Transport Service Primitieven
Eenvoudiger dan Netwerk Service Primitieven, redenen kennen.

6.1.3 Berkeley Sockets
Socket mechanisme begrijpen

6.1.4 Socket Programming
voorbeeld doorlezen

6.2 Elementen van transportprotocollen

6.2.1 adressering, relatie TSAP, NSAP

6.2.2 verbinding maken
Start volnummers volgens Tomlinson's algoritme.
overeenstemming door drievoudige handdruk.

6.2.3 vrijgeven van een verbinding
Het twee-leger probleem

6.2.4 flow control en buffering
Betekenis van flow control.
Flow control in de transportlaag is gebaseerd op een sliding window protocol (net als in de datalinklaag), maar de Ack is nu losgekoppeld van de toestemming om de volgende TPDU te zenden. Deadlock risico.

Relatie window size en netwerk capaciteit

6.2.5 multiplexing
Upward en downward multiplexing en voorbeelden van toepassingen

6.2.6 crash recovery
Omdat een ontvangende Transport Entiteit zowel iets naar zijn hogere laag als naar zijn peer moet zenden, en deze acties niet als één atomaire actie kunnen worden uitgevoerd, bestaat er geen methode van herstarten die nooit een pakket dubbel aflevert of verloren laat gaan.

6.3 Voorbeeld: een simpel transport protocol
Bekijken; met name de verschillende representatievormen kennen en begrijpen (zoals figuur 6.21 en 6.22)

6.4 Internet transport protocollen: UDP

6.4.1 principes van UDP kennen

6.4.2 RPC principes kennen.
begrippen: stub, marshalling

6.4.3 RTP principe kennen

6.5 Internet Transport protocollen: TCP

6.5.1 Introduction

6.5.2 TCP service model
Sockets,
Transport Adres = IP adres + port number
well known port numbers
verbingsgericht, full duplex, point-to-point
bytestream (iedere byte zijn eigen nummer)
PUSH flag, urgent data

6.5.3 TCP protocol
Segmenten, segment header, maximum transfer unit (MTU)

6.5.4 TCP Segment header
Betekenis van de verschillende velden en opties kunnen uitleggen.

6.5.5. TCP connection establishment
Verbinding maken met 3-voudige handdruk en SYN en ACK bits

6.5.6 TCP connection release

verbinding beëindigen door FIN bits

6.5.7 TCP connection management modeling

6.5.8 TCP transmission policy

Nagles algoritme kennen en begrijpen

Silly Window Syndrome kennen en begrijpen

6.4.6 TCP congestie beheersing

Naast een window voor flow control ook een congestie window. De kleinste van de twee geldt.

Het congestie window begint klein, als er geen timeout optreedt wordt het vergroot, totdat een drempel wordt overschreden of er een time-out optreedt. window wordt dan weer kleiner gemaakt etc.

6.5.10 TCP timer management

Belangrijke timers:

T_R retransmissie timer: dynamisch aan te passen aan wisselende Round Trip Times (RTT's) m.b.v Jacobson's algoritme (kennen!)

T_P Persistentie timer

T_A Keep-Alive Timer

T_C Timed-Wait-State Timer

functie van de laatste 3 timers kennen.

6.5.11 Draadloos TCP en UDP

Problemen bij draadloos TCP :

- Congestion control werkt averechts bij fouten
- Inhomogene paden: Oplossing: indirect TCP of snooping agent

6.5.12

behoort niet tot de tentamenstof

6.6 Performance Issues

6.6.1 performance problemen

- tijdelijk, b.v. als gevolg van congestie
- permanent b.v. als gevolg van een onbalans of verkeerde parameterwaarden.

6.6.2 het meten van performance

Interessante eenvoudig te meten gegevens:

- Round Trip Time (RTT)
- aantal retransmissies (als percentage)
- effectieve bandbreedte (bit/s)

Punten van aandacht:

- Begrijpen wat je meet

- representativiteit
- controleerbare omgeving
- sample size
- effect van caching en buffering
- gevaren van extrapolatie

6.6.3 ontwerpen met het oog op performance

- Processing capaciteit vaak belangrijker dan bandbreedte
- Beperk het aantal PDU's
- Beperk het aantal context switches
- Buffers zo min mogelijk kopiëren
- Congestie voorkomen is beter dan genezen.
- Kies time-out waarden conservatief
- Optimaliseer op de meest voorkomende situatie

6.6.4 Fast TPDU processing

alleen het principe kennen

6.6.5 Protocols for Gigabit netwerken

Essentieel: bij grote bandbreedte is propagatietijd de beperkende factor
Vermijd daarom terugkoppeling zoveel mogelijk.

6.7 Summary doorlezen

7. APPLICATIELAAG

7.1 DNS

7.1.1 DNS Name Space

kennen

7.1.2 Resource Records

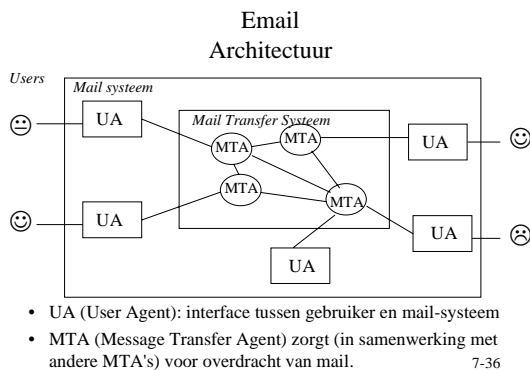
principe kennen

7.1.3 Name Servers

zones, primaire name server, authoritative informatie, secundaire servers, caching. name resolution, recursieve navigatie, iteratieve navigatie

7.2 Email:

7.2.1 Architecture and services



7.2.2 User Agent

7.2.3 Message Formats

RFC822, MIME, base64 encoding, quoted printable encoding

7.2.4 Message Transfer

SMTP protocol

7.2.5 Final Delivery

POP3 en IMAP: principes kennen en de belangrijkste verschillen

7.3 World Wide Web

7.3.1 Architectural overview kennen

7.3.2 Static Web Documents

principes van HTML en FORMS kennen (niet alle tags van buiten leren)
de functie van XML, XSL en XHTML kennen

7.3.3 Dynamic Web Documents

Vershil tussen client-side en sever-side Web
Page generation kennen

werkwijze van CGI interface, PHP, Javascript kunnen beschrijven. Geschiktheid van de verschillende methoden voor verschillende soorten toepassingen

7.3.4 HTTP

functie, structuur en werkwijze kennen

7.3.5 Performance Enhancements

Principes en belangrijkste eigenschappen kennen van Caching, Server replication, Content Delivery Networks.

Het materiaal vanaf 7.3.6 behoort niet tot de tentamenstof. (Maar is het bestuderen zeker waard)