Naam	Initialen	Studentnummer	Geboortedatum	Studierichting

Tentamen Telematica Systemen en Toepassingen (261000)
8 november 2006
9.00 – 12.30

Opmerkingen:

- Alleen 1 dubbelzijdig blad met aantekeningen / samenvatting (ongeacht lettergrootte / dichtheid) en een woordenboek zijn toegestaan als hulpmateriaal. Het gebruik van het boek van Kurose en Ross of enig ander materiaal is niet toegestaan.
- Gebruik van PDA, laptop computer, mobiele telefoon, enz., is niet toegestaan. Schakel je mobiele telefoon uit en berg hem op.
- Geef je antwoorden op deze bladen.
- Aanduidingen zoals "[10]" bij vragen betekenen dat je 10 punten voor die vraag kunt verdienen.
- Vul je naam, studentnummer, enz., bovenaan deze bladzijde in.

Alleen voor de docent:

Vraag	1	2	3	4	5	6	7	Totaal
Punten								
Maximum	10	12	17	16	16	7	8	86

Lijst van afkortingen:

ARP	Address Resolution Protocol
DNS	Domain Name System
FSM	Finite State Machine
FTP	File Transfer Protocol
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IP	Internet Protocol
MAC	Medium Access Control
NAT	Network Address Translator
POP3	Post Office Protocol version 3
RTT	Round Trip Time
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

1) Algemene vragen [10]

Geef van ieder van de volgende stellingen aan of deze waar of niet waar is. Omcirkel het juiste antwoord. [**Let goed op:** goed: +1 punt; fout: -1 punt; geen antwoord: 0 punten]

a) Gedurende de *slow-start* fase vergroot de TCP zender zijn *congestion window* met één segment wanneer een nieuwe *acknowledgement* wordt ontvangen.

waar / niet waar

b) Een TCP ontvanger stuurt bij ieder pakket dat hij terug stuurt naar de zender het *receive window header* veld mee.

waar / niet waar

c) Een gewone router laat bij het doorsturen van een pakket de IP header ongewijzigd.

waar / niet waar

d) Door het gebruik van *limited scope query flooding* is het in Gnutella mogelijk dat bepaalde *content* niet gevonden wordt terwijl het wel in het netwerk aanwezig is.

waar / niet waar

e) Met ARP kan een IP adres omgezet worden naar een domein naam.

waar / niet waar

f) Connectie-georienteerde communicatie vindt altijd plaats via circuit-geschakelde netwerken.

waar / niet waar

g) *Soft-state* protocollen zijn altijd *pull*-protocollen.

waar / niet waar

h) Routers in het internet hebben geen notie van *end-to-end* applicaties.

waar / niet waar

i) Hoewel TCP een bidirectionele transportservice biedt aan de beiden eindpunten, is de waarde van de hertransmissie *time-out* niet hetzelfde voor de beide richtingen.

waar / niet waar

j) Sattelietcommunicatie tussen Europa en de VS leidt tot lagere vertragingstijden dan communicatie via een zeekabel.

waar / niet waar

2) Applicatieprotollen [12]

a) Waarom worden naast IP adressen ook nog poortnummers gebruik binnen applicatieprotocollen; de IP adressen zijn toch wereldwijd uniek? [1]

b) Waarom maakt DNS gebruik van UDP en niet van TCP? Wat is hiervan het gevaar? [1]

c) Waarom is FTP niet geschikt voor grootschalige "*on-line publishing*"? [1]

Een webpagina (op *webserver S*) bestaat uit een basis HTML frame en 3 ingebedde plaatjes. Vanaf *client* machine *C* willen we deze webpagina downloaden. De RTT tussen *C* en *S* is constant. Ga er van uit dat het IP adres van de *webserver* bij de *client* bekend is.

d) Hoeveel TCP verbindingen worden gebruikt alvorens bovenstaande webpagina in z'n geheel te zien is via de *browser* van *client* machine *C* in geval dat:

d1) HTTP/1.0 wordt gebruikt? [1]

d2) HTTP/1.1 zonder *pipelining* wordt gebruikt? [1]

d3) HTTP/1.1 met *pipelining* wordt gebruikt? [1]

e) Geef expressies (en motiveer die) voor de downloadtijd *DLT* voor de drie in deelopgave (d) onderscheiden gevallen. Ga er van uit dat de TCP *flow*- en *congestion-control* mechanismen geen vertragende/beperkende factor zijn, dat de HTML basispagina B_1 bytes omvat, en de drie objecten respectievelijk B_2, B_3 en B_4 bytes. De transmissiesnelheid tussen *C* en *S* bedraagt *R* bytes per seconde. Voor de transmissietijden van de TCP *SYN*, TCP *SYN ACK* en HTTP *REQUEST* berichten mag je aannemen dat ze verwaarloosbaar klein zijn.

e1) Bij gebruik van HTTP/1.0 [1]

e2) Bij gebruik van HTTP/1.1 zonder *pipelining* [1]

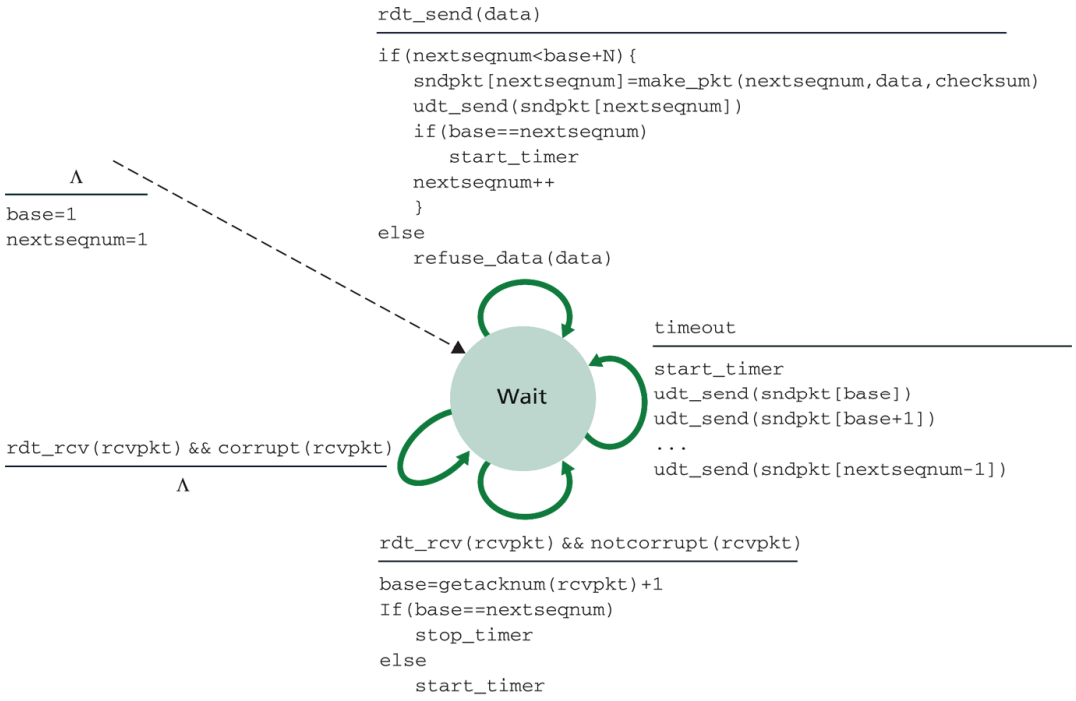
e3) Bij gebruik van HTTP/1.1 met *pipelining* [1]

f) Het doel van web *caching* is *download*tijden in het internet te verkleinen. Daartoe worden locaal objecten bewaard in zgn. *proxy-servers*. Deze *proxy-servers* hebben echter een eindig geheugen, zodat op een gegeven moment eerder opgeslagen objecten moeten worden weggegooid om het bewaren van nieuwe objecten mogelijk te maken. Bedenk en beschrijf en motiveer een efficiente strategie om te beslissen welke objecten weggegooid moeten worden. Efficient betekent hierbij dat we er naar moet streven zo vaak mogelijk een HTTP GET *request* vanuit de cache te kunnen afhandelen. [2]

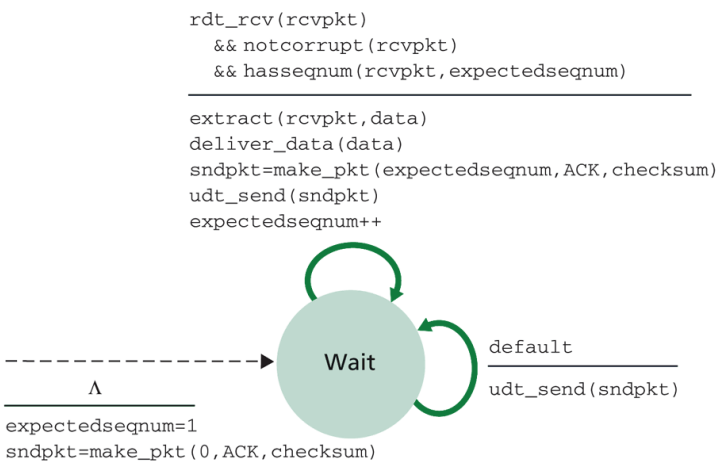
g) Leg kort uit waarom DNS *caching* (in locale DNS servers) de *load* balanceringsfunctionaliteit van DNS ondergraaft. [1]

3) Go-Back-N [17]

Hieronder vind je de *extended finite state machines* voor zender en ontvanger van het Go-Back-N protocol, zoals die in het boek van Kurose en Ross te vinden zijn.



Extended FSM description of GBN sender

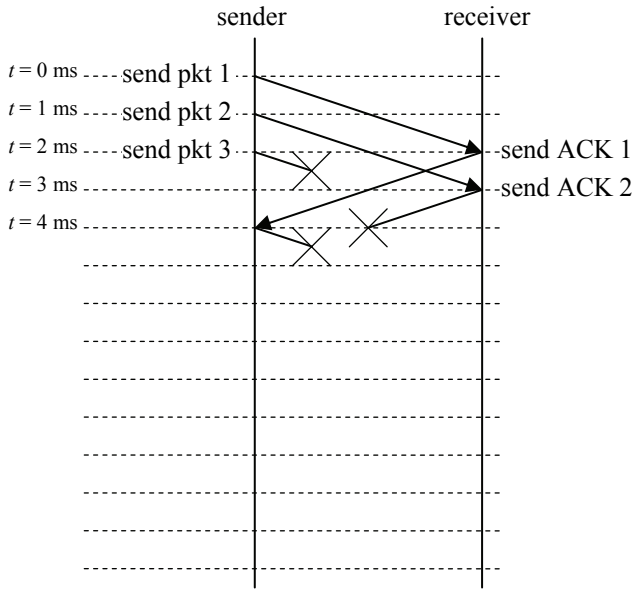


Extended FSM description of GBN receiver

a) De belangrijkste functie van *Go-Back-N* is het verzorgen van betrouwbare dataoverdracht. Geef twee andere belangrijke functies die naast betrouwbare dataoverdracht in de transportlaag geimplementeerd kunnen zijn. [2]

Neem in de rest van deze opgave aan dat zender en ontvanger operen volgens het *Go-Back-N* protocol zoals in de bovenstaande FSMs beschreven, met **N = 3**, een onbeperkt aantal beschikbare *sequence* nummers en een **timerwaarde** van **5 ms**. De propagatiedelay tussen zender en ontvanger (en vice versa) is constant: 2 ms. De tijd nodig voor het verwerken van een pakket is verwaarloosbaar. Tenslotte wordt aan de zendkant voortdurend nieuwe data aangeboden voor verzending.

Hieronder vind je een *Time-Sequence Diagram* van het initiële verloop van de operatie. Op tijdstip $t = 0$ ms verstuurt de zender een pakket met *sequence* nummer 1. Dit komt binnen bij de ontvanger op tijdstip $t = 2$ ms waarna de ontvanger direct een *acknowledgement* met *sequence* nummer 1 terug stuurt, welke op tijdstip $t = 4$ ms door de zender ontvangen wordt. Op tijdstip $t = 1$ ms wordt door de zender een pakket met *sequence* nummer 2 verstuurd. Na ontvangst op tijdstip $t = 3$ ms stuurt de ontvanger meteen een *acknowledgement* met hetzelfde *sequence* nummer terug, dat verloren gaat. Tenslotte stuurt de zender op tijdstip $t = 2$ ms een pakket met *sequence* nummer 3 dat ook verloren gaat.

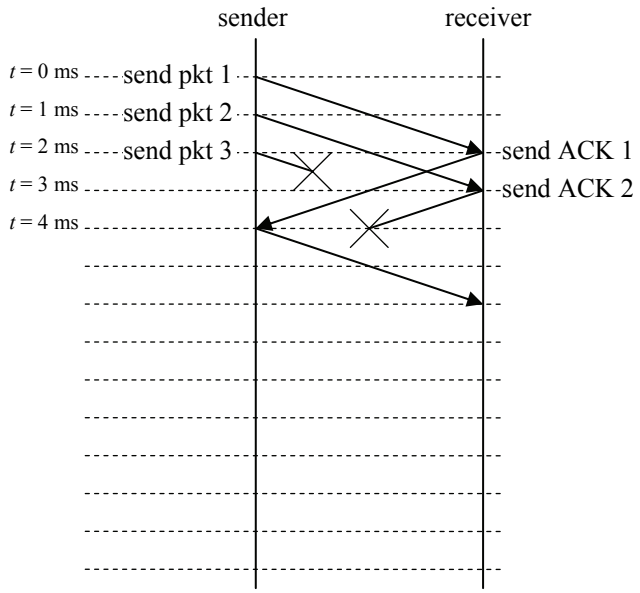


b) Op tijdstip $t = 4$ ms zal de zender opnieuw een pakket versturen. Wat is het *sequence* nummer van dit pakket? [1]

c) Veronderstel dat dit pakket ook verloren gaat. Wanneer zal bij de zender de timer aflopen? [1]

d) Welk pakket (met welk *sequence* nummer) zal de zender op dat tijdstip (als eerste) opnieuw versturen? [2]

Veronderstel nu dat het pakket dat bij b) verzonden is (op tijdstip $t = 4$ ms), en alle volgende, wel correct bij de ontvanger aankomen. Zie onderstaand *Time-Sequence Diagram*. Maak voor onderstaande opgaven opnieuw gebruik van de FSM aan het begin van deze opgave.



e) Welke *acknowledgement* (met welk *sequence* nummer) zal de ontvanger sturen na ontvangst van het bij b) (op tijdstip $t = 4$ ms) verzonden pakket? [1]

f) De zender mag na ontvangst van deze *acknowledgement* nog een pakket sturen. Waarom is dat zo? Welk *sequence* nummer heeft dit pakket? [2]

Waarom?:

Sequence nummer:

g) Wanneer dit pakket bij de ontvanger binnenkomt, welk *sequence* nummer zal de *acknowledgement* hebben die de ontvanger vervolgens verstuurt? [1]

h) Zal de zender na ontvangst van deze *acknowledgement* nog een pakket sturen? Zo ja, met welk *sequence* nummer? Indien nee, waarom niet? [1]

i) Wanneer zal bij de zender de *timer* aflopen en zal de zender een pakket opnieuw versturen? [1]

j) Welk pakket (met welk *sequence* nummer) zal de zender op dat tijdstip (als eerste) opnieuw versturen? [1]

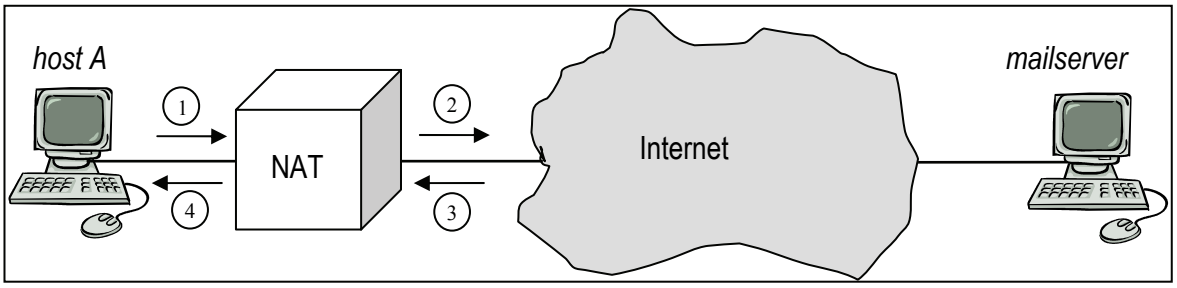
k) Doe een voorstel hoe dit *Go-Back-N* protocol efficienter kan worden gemaakt door op een andere manier met *timers* om te gaan. Geef een voor- en een nadeel van je voorstel. [2]

l) In de collegeslides hebben we laten zien dat, zelfs indien de onderliggende service behoud van volgorde garandeert, het *Selective Repeat* protocol niet gegarandeerd goed werkt wanneer het aantal beschikbare sequence nummers kleiner is dan 2*N*. Geldt dit ook voor *Go-Back-N*? Zo ja, waarom? Indien nee, beargumenteer hoeveel beschikbare sequence nummers er dan minimaal moeten zijn? [2]

4) Adressering [16]

Beschouw de volgende configuratie. *Host* A is via een Network Address Translator (NAT) verbonden met het internet. Een *mailserver*, van welke de gebruiker van *host* A zijn emails wil ophalen m.b.v het POP3 protocol is ook verbonden met het internet. Het volgende kan gezegd worden over de IP adressen van de verschillende nodes:

host A:	10.0.0.5
NAT intern (kant van host A):	10.0.0.1
NAT extern (kant van het internet):	130.89.227.131
mailserver:	64.147.163.246



De pakketten die over de links van het hier getekende netwerk verstuurd worden zullen *header* informatie bevatten om ze uiteindelijk bij het juiste proces op de juiste host af te kunnen afleveren. De belangrijkste *header* informatie is *Source IP Address* en *Destination IP Address, Source MAC Address* en *Destination MAC Address* en *Source Port Number* en *Destination Port Number*.

a) Geef voor ieder van deze paren aan bij welke laag van de Internet protocol *stack* ze horen. [2]

Source IP Address en *Destination IP Address*:

Source Port Number en *Destination Port Number*:

Source MAC Address en *Destination MAC Address*:

b) Wat is het voordeel van het gebruik van een NAT boven het gebruik van een *router* om *hosts* aan te sluiten op het internet? [2]

c) Noem een nadeel dat het gebruik van een NAT heeft, bijvoorbeeld voor *peer-to-peer* toepassingen. [2]

Om in te loggen op de *mailserver* verstuurt *host* A op zeker moment een pakket naar de NAT met het POP3 commando “user bob” (pakket (1)). We gaan er van uit dat daarvoor al een TCP verbinding geopend is. De NAT zal dit pakket met enigszins gewijzigde *header* informatie doorsturen in de richting van het internet (pakket (2)). Uiteindelijk zal het pakket de *mailserver* bereiken, en zal de *mailserver* een pakket terugsturen (bijvoorbeeld met het POP3 antwoord “+OK”). Op zeker moment zal dit pakket bij de NAT aankomen (pakket (3)). Tenslotte zal de NAT het pakket met wat wijzigingen aan de *header* informatie doorsturen naar *host* A (pakket (4)).

In onderstaande tabel is voor de genoemde pakketten (1) – (4) een deel van de *header* informatie weergegeven. De bedoeling van de rest van deze opgave is de lege velden in de tabel in te vullen. Aan het eind van de opgave moet de tabel dus geheel gevuld zijn! De kleine letters met haakje (bv. c)) geven aan bij welke deelvraag het betreffende veld ingevuld moet worden.

	packet (1) (host A → NAT)	packet (2) (NAT → internet)	packet (3) (internet → NAT)	packet (4) (NAT → host A)
source MAC address	22-22-22-22-22-22	33-33-33-33-33-33	d)	d)
destination MAC address	44-44-44-44-44-44	55-55-55-55-55-55	d)	d)
source IP address	e)	e)	e)	e)
destination IP address	e)	e)	e)	e)
source port number	1023	h)	h)	f)
destination port number	110	g)	1034	f)

d) Bedenk aan de hand van de MAC adressen zoals die in pakket (1) en (2) te vinden zijn wat de MAC adressen zijn van de adapters van de relevante netwerk nodes. Gebruik deze informatie om in bovenstaande tabel de MAC adressen van pakket (3) en (4) in te vullen. (4 MAC adressen invullen) [2]

e) Gebruik het lijstje met IP adressen aan het begin van de opgave om in bovenstaande tabel de IP adressen in te vullen. Vraag je daarbij af in hoeverre een NAT IP adressen verandert. (8 IP adressen invullen) [3]

f) Het gebruik van een NAT is niet zichtbaar voor *host* A. Bedenk wat dit betekent voor de poortnummers in pakket (4)? Vul de poortnummers in de bovenstaande tabel in. (2 poortnummers invullen) [2]

g) Beredeneer of, en zo ja, op welke manier, de NAT het *destination* poortnummer verandert voor pakket (2) (ten opzichte van pakket (1)). Vul het gevonden poortnummer in in de tabel. (1 poortnummer invullen) [1]

h) Ook voor de *mailserver* is het gebruik van de NAT niet zichtbaar. Wat is dus het *source* poortnummer geweest van pakket (2) en van pakket (3)? (2 poortnummers invullen) [2]

5) Routers [16]

a) Noem 4 verschillende componenten van een router. In welke component(en) kan queueing optreden? In welke component worden routeringsprotocollen uitgevoerd? [4]

1:

2:

3:

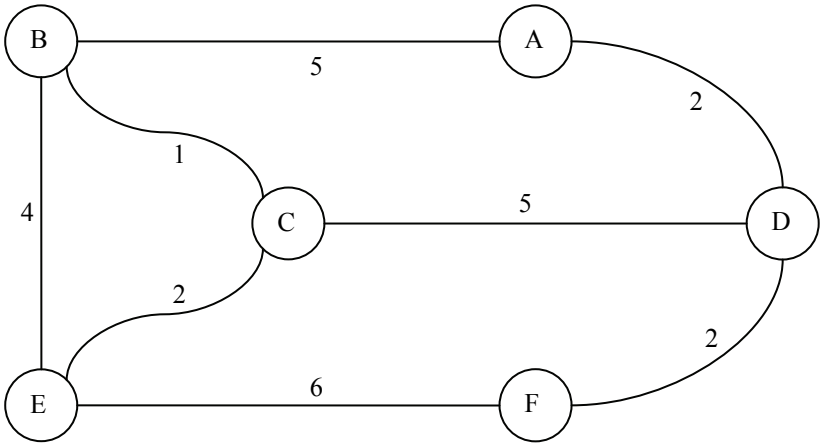
4:

queueing:

routering:

b) Geef de 2 belangrijkste redenen waarom hierarchische routering gebruikt wordt (bijvoorbeeld in het Internet). [2]

Beschouw nu het volgende netwerk met de daarin aangegeven link-kosten:



c) Gebruik Dijkstra's algoritme voor *link-state* routering om de kortste paden van node A naar alle andere nodes te berekenen. Gebruik het algoritme om de volgende tabel in te vullen, waarin N' de verzameling van al afgehandelde nodes is, D(x) de afstand tot node x, en p(x) de voorganger van x op het kortste pad van A naar x. [5]

step	N'	D(B), p(B)	D(C), p(C)	D(D), p(D)	D(E), p(E)	D(F), p(F)
0	{A}	5,A	∞	2,A	∞	∞
1						
2						
3						
4						
5						

Een alternatief algoritme voor het bepalen van kortste (of goedkoopste) paden is het *distance-vector*-algoritme. In de rest van deze opgave richten we ons op dit algoritme, en in het bijzonder op node A, en het gezichtspunt dat node A heeft op het hierboven gegeven netwerk. We gaan uit van het gebruik van het standaard *distance-vector*-algoritme, **zonder** *poisoned reverse*. Op tijdstip t_0, na convergentie van het *distance-vector*-algoritme, heeft node A de volgende afstandtabel gecreëerd. Hierbij geeft de regel van A, A's **eigen** *distance vector*. De regels van B en D geven de *distance vectors*, **zoals A ze heeft ontvangen** van B, respectievelijk D.

A		Cost to					
		A	**B**	**C**	**D**	**E**	**F**
from	**A (computed)**	0	5	6	2	8	4
	B (received)	*5*	*0*	*1*	*6*	*3*	*8*
	D (received)	*2*	*6*	*5*	*0*	*7*	*2*

d) Enige tijd later, op tijdstip t_1 ($t_0<t_1$), ontdekt node A dat de kosten van de link van node A naar node B gewijzigd zijn van 5 naar 7. Als gevolg hiervan verandert de afstandtabel van node A. Geef alle waarden van de afstandtabel in de onderstaande tabel aan. [3]

A		Cost to					
		A	**B**	**C**	**D**	**E**	**F**
from	**A (computed)**						
	B (received)						
	D (received)						

e) Weer enige tijd later, op tijdstip t_2 ($t_0<t_1<t_2$), ontdekt node D dat de kosten van zijn link naar node C gewijzigd zijn van 5 naar 3. Node D berekent een nieuwe *distance vector* en stuurt die naar zijn buren. Deze *distance vector* wordt ontvangen door node A op tijdstip t_3 ($t_0<t_1<t_2<t_3$), en luidt
$D_D = [D_D(A), D_D(B), D_D(C), D_D(D), D_D(E), D_D(F)] = [2, 4, 3, 0, 5, 2]$.
Als gevolg hiervan verandert de afstandtabel van node A. Geef alle waarden van de afstandtabel in de onderstaande tabel aan. [2]

A		Cost to					
		A	**B**	**C**	**D**	**E**	**F**
from	**A (computed)**						
	B (received)						
	D (received)						

6) Aloha, slotted Aloha en Ethernet [7]

Voor Ethernet is afgeleid dat de efficiency kan worden benaderd via de volgende formule: $Eff = 1 / (1 + 5\ T_{prop}/T_{trans})$, waarbij T_{trans} de transmissie tijd van een pakket is, en T_{prop} de maximale propagatietijd.

a) Waarom is voor Ethernet een korte kabellengte efficient? [1]

b) Waarom gebruikt een netwerkbeheerder in een typische kantooromgeving liever switches dan routers? En waarom liever switches dan hubs? [2]

c) Gegeven de vraagstelling in b), waarom worden er in het internet uberhaupt nog routers gebruikt, en niet alleen switches? [1]

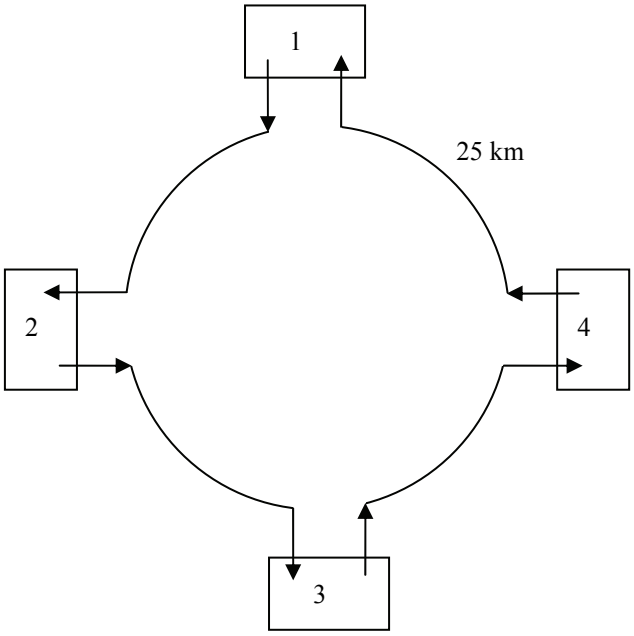
In het boek is aangegeven dat de efficiency voor slotted Aloha bij N actieve nodes gelijk is aan $E_N(p)= Np(1-p)^{N-1}$. Daarbij is ook aangegeven dat de maximale efficientie wordt bereikt wanneer $p=1/N$ wordt gekozen.

d) Laat voor het geval $N=3$ zien dat $E_3(p)$ inderdaad gemaximaliseerd wordt voor $p = 1/3$. **Tip**: beschouw $E_3(p)$ als een functie van p op het domein [0,1] en gebruik de afgeleide daarvan. [2]

e) Wat is in dat optimum de waarde van de bereikte efficientie? [1]

7) Locale netwerken: token ring [8]

We beschouwen een token ring die functioneert volgens het *token-passing taking-turns* protocol (zoals beschreven in Section 5.3). In zo'n token-ring systeem mag het station dat het *token* "in bezit heeft" een pakket versturen. Nadat een pakket is verstuurd en weer terugontvangen is (zie hieronder) wordt het *token* doorgegeven aan het volgende station.



We beschouwen de volgende configuratie (zie ook de afbeelding). We hebben te maken met $N = 4$ stations, die allen voortdurend een wens hebben pakketten te verzenden, ter (vaste) grootte van $P = 16000$ bits (inclusief alle header en trailer informatie). Het *token* zelf beslaat 160 bits (all-in). De propagatiesnelheid bedraagt $C = 2 \cdot 10^5$ km/s, en de ring is $L = 100$ km lang; tussen naburige stations is de afstand telkens 25 km. Als een pakket of *token* "langs een station komt" of door een station daadwerkelijk wordt ontvangen, treedt daardoor een extra (*processing*) vertraging van $D = 80$ bittijden op. De transmissiesnelheid R van het medium ("de kabel") bedraagt 8 megabit per seconde (dwz. $R = 8 \cdot 10^6$ bit/s).

a) Hoeveel meter is het *token* lang? Toon je berekening. [1]

b) Wanneer een station het token ontvangt van zijn buurstation, mag het station precies 1 pakket (met vaste grootte) zenden. Hoe lang duurt het zenden. Toon je berekening! [1]

c) Het verzonden pakket propageert geheel rond de ring. Alle stations bekijken het pakket (wat die 80 bittijden vertraging per station verklaart). Alleen het station waarvoor het pakket bestemd is (dit kan dat station vaststellen door gedurende die 80 bittijden naar onder andere het adresveld in de *header* te kijken) kopieert het pakket. Het pakket circuleert gewoon verder over de ring, totdat het zendende station het pakket terugontvangt (dit kost ook 80 bittijden). Daarna geeft dat station het *token* door aan zijn buurstation. Hoe groot is de tijd die verstrijkt tussen het verzenden van het laatste bit door het zendende station, en het geheel terugontvangen van dat pakket door het zendende station? Toon je berekening! [3]

d) Hoe lang duurt het om het *token* te versturen naar het buurstation en het door het buurstation geheel te ontvangen. Toon je berekening! [1]

e) Wanneer je je antwoorden van deelopgaves (b) t/m (d) combineert, hoe groot is dan de efficientie van dit *token-passing* protocol? De efficientie wordt daarbij gedefinieerd als de ratio van de tijd die effectief besteed wordt aan pakket-transmissie en de totaal verstreken tijd. Toon je berekening! [2]

---einde tentamen---