

Naam	Initialen	Studentnummer	Geboortedatum	Studierichting

Examen Telematica Systemen en Toepassingen (261000)

22 juni 2004

Opmerkingen:

- Alleen 1 dubbelzijdig blad met aantekeningen / samenvatting (ongeacht letter grootte / dichtheid) en een woordenboek zijn toegestaan als referentiemateriaal. Het gebruik van het boek van Kurose en Ross of enig ander materiaal is niet toegestaan.
- Gebruik van rekenmachine, PDA, laptop computer, mobiele telefoon, enz., is niet toegestaan. Schakel je mobiele telefoon uit.
- Geef je antwoorden op dit formulier. Wanneer je meer ruimte nodig hebt mag je extra bladen gebruiken.
- Aanduidingen zoals "[10]" bij vragen betekenen dat je 10 punten voor die vraag kunt verdienen.
- Vul je naam, studentnummer enz. bovenaan deze bladzijde in, en op eventuele verdere bladen die je gebruikt.

Alleen voor de docent:

Vraag	1	2	3	4	5	Totaal
Punten						
Maximum	21	21	21	20	17	100

Lijst van afkortingen:

CSMA/CD	Carrier Sense Multiple Access with Collision Detection
DNS	Domain Name System
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IP	Internet Protocol
MSS	Maximum Segment Size
NS	Name Server
RTT	Round Trip Time
TCP	Transmission Control Protocol

1) Algemene vragen [21]

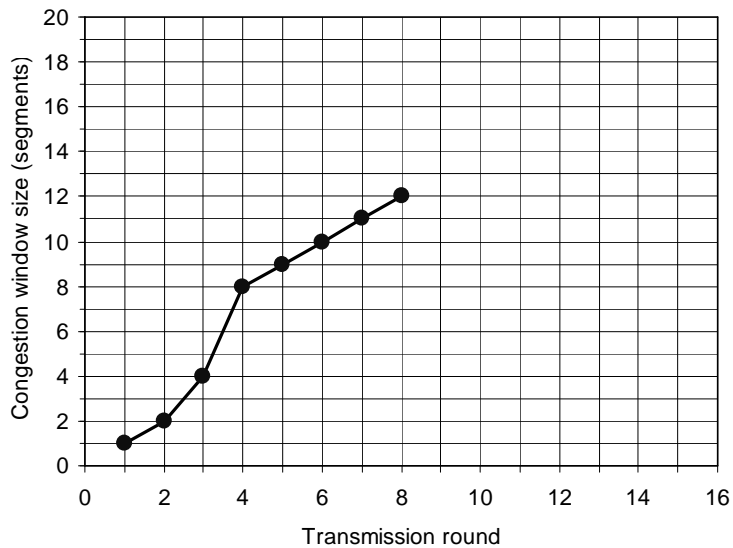
- a) Welke lagen van de *Internet protocol stack* worden in een router voor ieder pakket uitgevoerd? [2]
- b) In een netwerk dat gebruik maakt van *packet switching* wordt een packet van node naar node gestuurd totdat de bestemming bereikt is. In ieder van die nodes (en op de uitgaande link tot aan de volgende node) treedt een bepaalde vertraging (*delay*) op. Uit welke 4 typen *delay* bestaat deze zogenaamde *node delay*? Geef voor ieder van de 4 typen aan hoe dit type *delay* zo klein mogelijk gemaakt kan worden. [4]
- 1)
 - 2)
 - 3)
 - 4)
- c) Stel *host A* verstuurt een aantal IP datagrammen via het Internet naar *host B*. Geef 3 mogelijke redenen waarom niet alle datagrammen dezelfde *delay* ondervinden op hun weg van A naar B. [3]
- 1)
 - 2)
 - 3)
- d) Kan een *reliable data transfer protocol* met alleen negatieve *acknowledgements* (NAKs) en geen positieve *acknowledgements* (ACKs) een betrouwbare dataoverdracht bieden? Licht je antwoord toe. [3]
- e) Wat is het verschil tussen *flow control* en *congestion control*? [3]

f) De lengte van het *timeout interval* wordt in het TCP protocol aangepast aan actuele metingen van de RTT. Stel dat TCP op een bepaald moment een RTT meet die veel kleiner is dan voorgaande gemeten RTTs. Geef aan waarom de lengte van het *timeout interval* als gevolg van die kleine gemeten RTT groter kan worden. [3]

g) Hoeveel IP adressen bevat het adresblok 130.89.13.0/28 ? [3]

2) **TCP Congestion Control [21]**

Beschouw het volgende verloop van de grootte van het TCP *congestion window* als functie van de tijd. De gebruikte versie van TCP is TCP Reno (TCP met *fast retransmit* en *fast recovery*).



a) Gedurende welk tijdsinterval werkt *congestion avoidance*? Gedurende welk tijdsinterval *slow start*? [3]

b) Wat is de initiële waarde van de variabele *Threshold* (tijdens de 1^{ste} transmissieronde)? [2]

In de onderstaande vragen moet je aangeven hoe het verloop is van de grootte van het *congestion window* vanaf transmissieronde 8. Hierbij is het volgende gegeven:

- 1) Direct na transmissieronde 8 wordt pakketverlies gedetecteerd door de ontvangst van 3 *duplicate ACKs*.
 - 2) Direct na transmissieronde 11 wordt pakketverlies gedetecteerd door het optreden van een *time-out*.
- c) Wat is de grootte van het *congestion window* tijdens transmissieronde 11? [3]
- d) Wat is de waarde van de variabele *Threshold* tijdens de 12^e transmissieronde? [2]
- e) Wat is de grootte van het *congestion window* tijdens transmissieronde 16? [2]

In de volgende vragen nemen we de vereenvoudigende aanname dat een transmissieronde precies 1 RTT duurt. Bovendien gaan we uit van een vaste segmentgrootte van 1500 bytes.

- f) Stel de RTT bedraagt 100 ms. Wat is de TCP *throughput* (gemiddeld over een transmissieronde) 200 ms na de eerste transmissieronde? [3]
- g) Stel de RTT bedraagt 500 ms. Hoeveel tijd verstrijkt vanaf de eerste transmissieronde tot het moment waarop de TCP *throughput* 240 kbit/s bedraagt. [3]

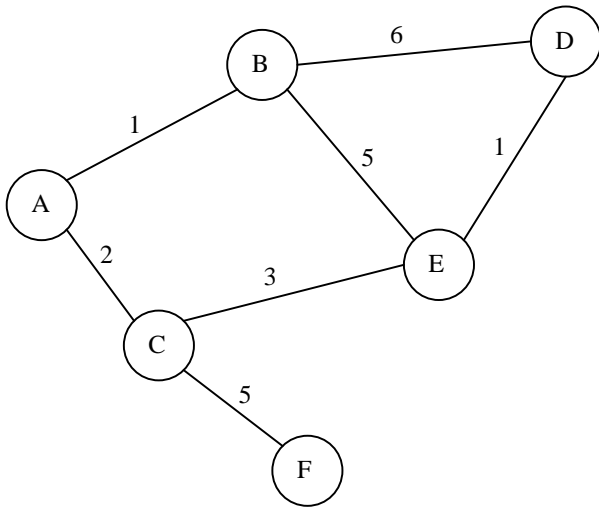
Uit meting aan een gegeven systeem blijkt dat geen pakketverlies optreedt en dat de TCP *throughput* van een bepaalde verbinding nooit boven de 5Mbit/s komt.

- h) Wat is een mogelijke oorzaak dat de TCP *throughput* nooit boven de 5 Mbit/s komt? [3]

3) Routing [21]

- a) Geef 2 belangrijke functies van de netwerk laag. [2]
- b) Geef de 2 belangrijkste redenen waarom hierarchische routing gebruikt wordt (bijvoorbeeld in het Internet). [2]

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



- c) Gebruik Dijkstra's kortste-pad-algoritme 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. [6]

step	N	D(B), p(B)	D(C), p(C)	D(D), p(D)	D(E), p(E)	D(F), p(F)
0	{A}	1,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 wat node A heeft op het hierboven gegeven netwerk. In opgave 3d t/m 3g gaan we 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:

$D^A()$		Cost to destination via	
		B	C
destination	B	1	5
	C	4	2
	D	7	6
	E	6	5
	F	9	7

- d) Enige tijd later, op tijdstip t_1 ($t_0 < t_1$), ontvangt node A van node C een bericht dat het pad van node C naar node F nu kosten 4 heeft. Dit noteren we als $D^C(F) = 4$. Als gevolg hiervan verandert de afstandtabel van node A. Geef alle waarden van deze afstandtabel in de onderstaande tabel aan, en beschrijf ook de berichten die node A naar zijn burens stuurt (als die er zijn). Dit kun je doen in de vorm $D^X(Y) = z$, wat betekent: de kosten van het pad van node X naar node Y bedragen z. [2]

$D^A()$		Cost to destination via	
		B	C
destination	B		
	C		
	D		
	E		
	F		

$$D^A(_) = _$$

$$D^A(_) = _$$

$$D^A(_) = _$$

(opm.: invullen op de streepjes; leeg laten indien geen of minder berichten worden verstuurd)

- e) Weer enige tijd later, op tijdstip t_2 ($t_0 < t_1 < t_2$), ontvangt node A van node B een bericht dat het pad van node B naar node F nu kosten 8 heeft: $D^B(F) = 7$. Als gevolg hiervan verandert de afstandtabel van node A. Geef alle waarden van de afstandtabel in de onderstaande tabel aan, en beschrijf ook de berichten die node A naar zijn burens stuurt (als die er zijn). [2]

$D^A()$		Cost to destination via	
		B	C
destination	B		
	C		
	D		
	E		
	F		

$$D^A(_) = _$$

$$D^A(_) = _$$

$$D^A(_) = _$$

(opm.: invullen op de streepjes; leeg laten indien geen of minder berichten worden verstuurd)

- f) Opnieuw enige tijd later, op tijdstip t_3 ($t_0 < t_1 < t_2 < t_3$), ontvangt node A van node C een bericht dat het pad van node C naar node D nu kosten 7 heeft: $D^C(D) = 7$. Als gevolg hiervan verandert de afstandtabel van node A. Geef alle waarden van de afstandtabel in de onderstaande tabel aan, en beschrijf ook de berichten die node A naar zijn burens stuurt (als die er zijn). [2]

$D^A()$		Cost to destination via	
		B	C
destination	B		
	C		
	D		
	E		
	F		

$$D^A(_) = _$$

$$D^A(_) = _$$

$$D^A(_) = _$$

(opm.: invullen op de streepjes; leeg laten indien geen of minder berichten worden verstuurd)

- g) Tenslotte, op tijdstip t_4 ($t_0 < t_1 < t_2 < t_3 < t_4$), ontdekt node A dat de kosten van de link van node A naar node B gewijzigd zijn van 1 naar 10. Als gevolg hiervan verandert de afstandtabel van node A. Geef alle waarden van de afstandtabel in de onderstaande tabel aan, en beschrijf ook de berichten die node A naar zijn burens stuurt (als die er zijn). [2]

$D^A()$		Cost to destination via	
		B	C
destination	B		
	C		
	D		
	E		
	F		

$$D^A(_) = _$$

$$D^A(_) = _$$

$$D^A(_) = _$$

(opm.: invullen op de streepjes; leeg laten indien geen of minder berichten worden verstuurd)

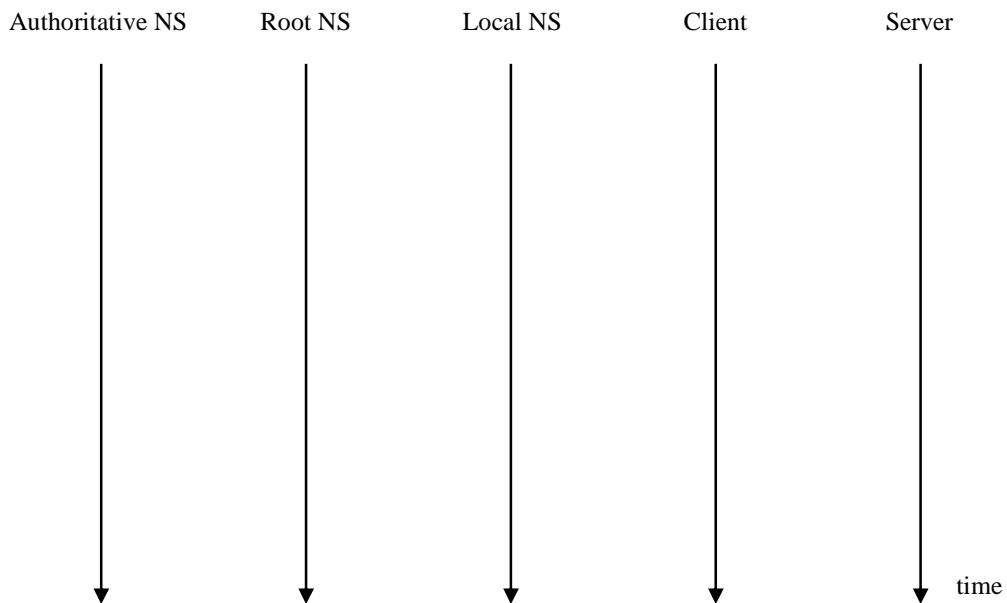
- h) Neem nu aan dat het gebruikte *distance-vector*-algoritme **wel** de *poisoned-reverse*-techniek gebruikt. Geef aan hoe de oorspronkelijke tabel (dus op tijdstip t_0 , na convergentie van het algoritme; boven opgave 3d) er uit zal zien in dit geval. Doe dat door de volgende tabel in te vullen. [3]

$D^A()$		Cost to destination via	
		B	C
destination	B		
	C		
	D		
	E		
	F		

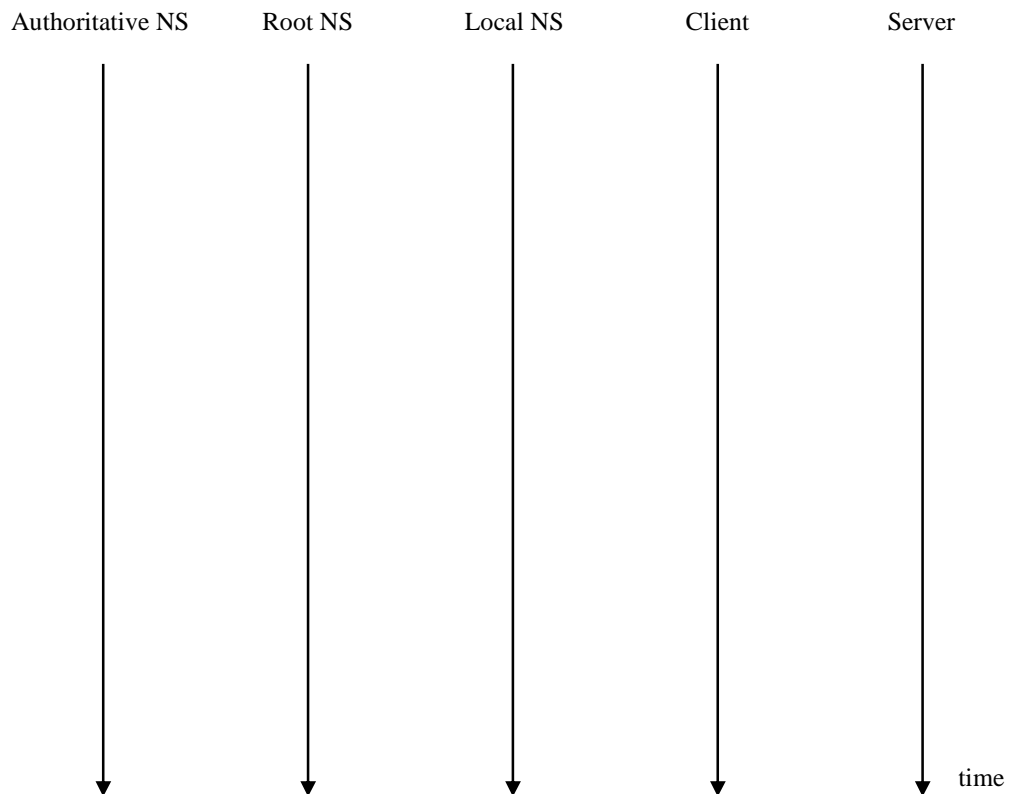
4) HTTP / DNS [20]

Een vriend uit de VS belt je op en meldt je dat hij z'n nieuwe web-server draaiend heeft onder de naam `www.formyfriend.com`. Naast een boel HTML tekst bevat de site ook een mooie JPEG afbeelding vertelt je vriend je nog per telefoon. Als eerste vanuit het `cs.utwente.nl` domein wil je nu zijn website bezoeken en je typt daartoe `http://www.formydearfriends.com/` in je browser in.

- a) Geef in onderstaand sequence diagram aan welke HTTP- en DNS-berichten tussen welke nodes worden uitgewisseld alvorens je de nieuwe website inclusief de afbeelding op je scherm ziet. De DNS servers werken daarbij iteratief en HTTP werkt met persistente connecties en pipelining. [4]



- b) Hoe groot is de *downloadtijd DLT* van de website? Introduceer daarbij de benodigde RTT variabelen (tussen met elkaar communicerende nodes). Ga uit van een statisch TCP *congestion window (W)* ter grootte van 3 segmenten ($W=3$). De grootte van de HTML tekst (M) is twee keer de Maximum Segment Size (S) en die van de afbeelding (N) drie keer S ($M=2S, N=3S$). De transmissie snelheid van de verbinding tussen de client en de server is R segmenten (van grootte S) per seconde. Ga uit van persistent HTTP met pipelining, en neem aan dat alle berichten die niet een deel van de webpagina transporteren dusdanig klein zijn dat daarvoor alleen RTTs van belang zijn (we maken dus dezelfde aannames als in het boek). Geef niet alleen een expressie voor *DLT*, maar leidt deze, stapsgewijs met behulp van een figuur af, waarin je nu ook de TCP berichten zichtbaar maakt. [5]



- c) Welke twee gevallen moet je in je berekening onderscheiden in het geval de windowgrootte 2 segmenten is ($W=2$)? [4]
- d) Hoe verandert DLT wanneer er sprake is van non-persistent HTTP? Licht je antwoord toe. [3]
- e) Leg uit hoe *cookies* kunnen worden gebruikt om een '*shopping cart*' in een *e-commerce site* te implementeren. Besteed hierbij aandacht aan zowel de *client*- als de *serverzijde*. [4]

5) Ethernet - CSMA/CD [17]

Gegeven een shared Ethernet transmissiekabel ter lengte L , waaraan een groot aantal nodes verbonden zijn die actief zijn met verzenden en ontvangen van frames. Tussen de twee nodes A en B, verbonden via dit Ethernet, is de afstand ook L . Stel dat gebruikt gemaakt wordt van 100BaseT Ethernet technologie, dat $L = 100\text{m}$ is, en dat de propagatievertraging $5 \mu\text{sec/km}$ is. Neem tenslotte aan dat de maximale lengte van een frame 1500 bytes is (deze maximale lengte bevat ook de lengte van het *preamble* veld). Deze aannames gelden voor de gehele opgave.

- a) Bereken de efficiëntie (efficiency) E . Wat is dan de maximale throughput? [3]
Gebruik hierbij dat

$$E = \frac{1}{1 + \frac{5t_{prop}}{t_{trans}}}$$

- b) In CSMA/CD luistert een node of andere nodes wat versturen, alvorens zelf wat te versturen. Waarom kunnen dan alsnog botsingen (*collisions*) optreden? [3]
- c) Welke minimale framegrootte is in de gegeven configuratie noodzakelijk? [3]
- d) Op tijdstip $t=0$ beginnen A en B met het verzenden van een frame. Op welke tijdstip(pen) detecteren A en B een collision? Aan welk bit (nummer) begint A dan net te zenden? En B? [4]
- e) Stel dat de parameter K , die gebruikt wordt in CSMA/CD exponential backoff, voor node A, oftewel $K_A = 20$ is, en voor node B, $K_B = 0$ is. Op welke tijdstip(pen) beginnen A en B met de hertransmissie? Gebruik hierbij de volgende notatie: het tijdstip waarop A en B klaar zijn met het verzenden van het CSMA/CD jam signaal is gelijk aan t_{jam} . De tijd die het kost het medium te *sensen* (*carrier sense*) is gelijk aan t_{sense} . [4]