

Naam	Initialen	Studentnummer	Studierichting

**Tentamen Telematica Systemen en Toepassingen (19261000/1)  
9 november 2011, 8.45 – 12.15**

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 gedrukt materiaal is niet toegestaan.
- Het gebruik van een rekenmachine is toegestaan. Gebruik van PDA, laptop computer, mobiele telefoon, enz., is niet toegestaan. Schakel je mobiele telefoon uit en berg hem op.
- De vragen worden gesteld in het Nederlands. Je mag in het Nederlands of Engels antwoorden, afhankelijk van waar je de voorkeur aan geeft.
- Geef je antwoorden op deze bladen. Wanneer je meer ruimte nodig hebt mag je extra papier 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 mogelijke extra vellen papier die je gebruikt.

Alleen voor de docent:

Vraag	1	2	3	4	5	6	7	Totaal
Punten								
Maximum	7	6	6	14	15	10	15	73

**Lijst van afkortingen:**

ACK	ACKnowledgement
AIMD	Additive Increase Multiplicative Decrease
CRC	Cyclic Redundancy Check
DNS	Domain Name System
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IMAP	Internet Mail Access Protocol
IP	Internet Protocol
MSS	Maximum Segment Size
P2P	Peer-to-Peer
PHP	PHP: Hypertext Preprocessor
POP3	Post Office Protocol – version 3
RTT	Round Trip Time
SMTP	Simple Mail Transfer Protocol
TCP	Transmission Control Protocol
VC	Virtual Circuit
XHTML	eXtensible HyperText Markup Language
XML	eXtensible Markup Language

1) World Wide Web [7]

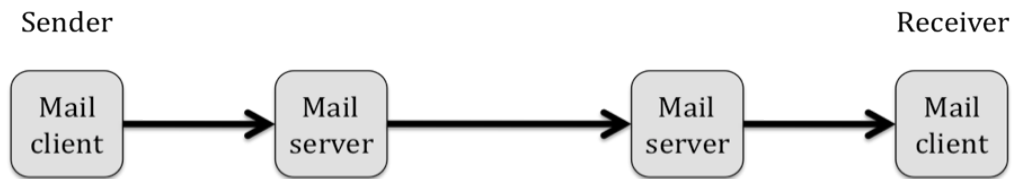
- a) De term Web2.0 is geïntroduceerd in 2004 door O'Reilly. Welke van onderstaande technologieën zijn karakteristiek voor Web2.0? [1]
- Javascript:
  - Asynchronous Javascript:
  - Flash:
  - PHP:
  - XML:
  - XHTML:
- b) Gegeven de onderstaande (vereenvoudigde) HTML pagina. Neem aan dat deze pagina wordt verstuurd over een verbinding met een *throughput* van 1 Gbps, en dat de *Round Trip Time* (RTT) 100 ms is. Neem tevens aan dat alle objecten waarnaar deze pagina refereert 1 KByte groot zijn. Bereken de totale *download* tijd voor het geval (*non-persistent*) HTTP 1.0 wordt gebruikt. Afronden op 10 ms is toegestaan. [3]

```
<!DOCTYPE html PUBLIC ...
<HTML>
<HEAD>
  <TITLE>Homepage Telematica Systemen en Toepassingen </TITLE>
  <META NAME="keywords" CONTENT="homepage">
  <LINK REL="stylesheet" HREF="basic.css">
</HEAD>
<BODY>
  </a>
  <P>Dit is de homepage van het vak Telematica Systemen
    en Toepassingen.</P>
  <P>De URL van de Universiteit Twente is
    <A HREF="http://www.utwente.nl/">UT</A></P>
</BODY>
</HTML>
```

- c) Dezelfde vraag, maar nu voor het geval dat HTTP 1.1, in de variant zonder *pipelining*, wordt gebruikt? [3]

## 2) Elektronische post [6]

- a) Teken de volgende email protocollen in de onderstaande figuur: IMAP, SMTP en POP3. [2]



- b) Zoals uit bovenstaande figuur blijkt, sturen email *clients* in het algemeen elektronische post eerst naar de eigen, lokale mail server. De lokale mail server stuurt vervolgens deze elektronische post door naar de mail server bij de bestemming. Waarom stuurt de zendende email *client* niet rechtstreeks naar de ontvangende email *client*? [2]

- c) Welke voordelen heeft IMAP in vergelijking met POP3? [2]

**3) DNS en Peer-to-Peer [6]**

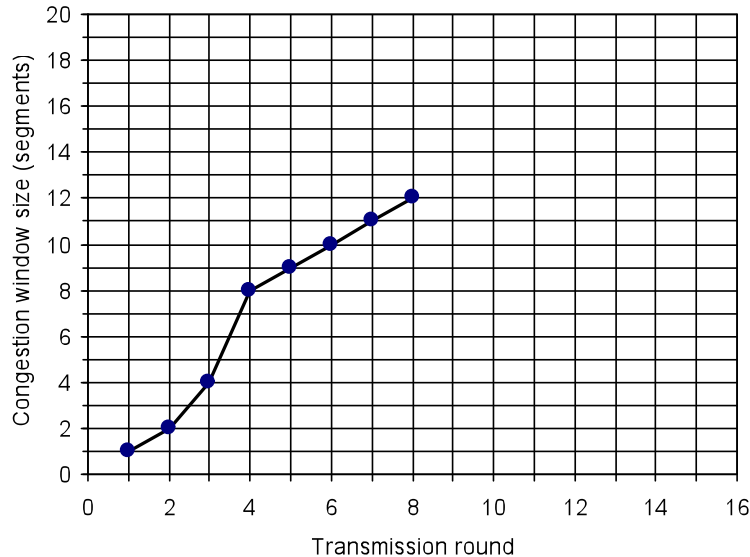
- a) DNS servers halen hun informatie uit databases. Welke databases zijn het grootst (bevatten de meeste gegevens): de databases in een gemiddelde root server of de database in een gemiddelde top-level domein server. Leg uit. [2]
  
- b) Leg het verschil uit tussen recursieve en iteratieve *queries* in DNS. [2]
  
  
  
  
  
  
  
  
  
  
- c) Stel je laptop, waarop je P2P software hebt geïnstalleerd, komt op de een of andere manier in het bezit van een opsporingsinstantie. Is het technisch mogelijk dat deze instantie nagaat met welke P2P *peers* jij het laatst hebt gecommuniceerd (dus voordat je laptop wordt onderzocht)? Leg uit. [2]



- e) Wanneer het dataveld  $D=11001$  is, en bovengenoemde generator wordt gebruikt. Wat zal dan de CRC code zijn? [3]
- f) Noem de twee belangrijkste redenen dat pakketten verloren kunnen gaan in het Internet. [2]
- g) Stel dat in de hierboven geschetste configuratie inderdaad in een bepaald pakket een bitfout optreedt tussen R en B. Hoe komt A te weten dat dit pakket niet correct bij B is aangekomen? Schets 2 verschillende mogelijke scenario's waar A concludeert dat B het pakket misschien niet correct ontvangen heeft. [2]

### 5) TCP Congestion Control [15]

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*).



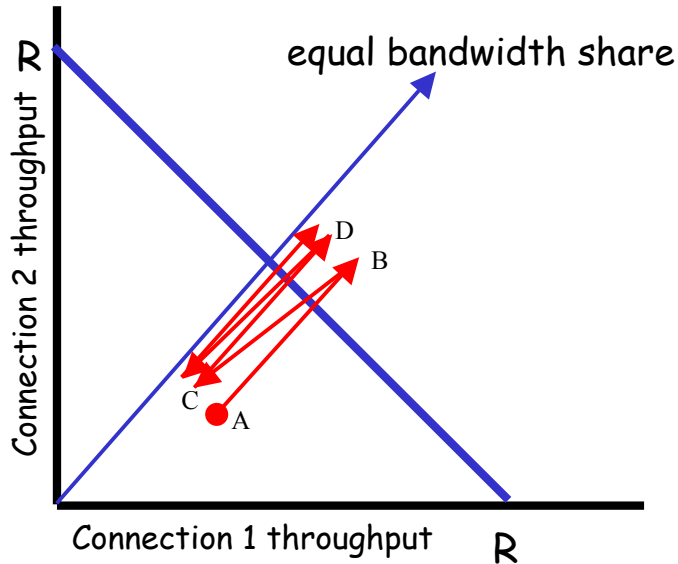
- a) Gedurende welk tijdsinterval werkt *congestion avoidance*? Gedurende welk tijdsinterval *slow start*? [2]
- b) Wat is de initiële waarde van de variabele `Threshold` (tijdens de 1<sup>ste</sup> transmissieronde)? [1]

In de onderstaande vragen moet je aangeven hoe het verloop is van de grootte van het *congestion window* vanaf transmissieronde 8. Hierbij is gegeven dat op twee momenten pakketverlies wordt gedetecteerd, en wel als volgt:

- 1) Direct na transmissieronde 10 wordt pakketverlies gedetecteerd door de ontvangst van 3 *duplicate* ACKs.
  - 2) Direct na transmissieronde 12 wordt pakketverlies gedetecteerd door het optreden van een *time-out*.
- c) Wat is de grootte van het *congestion window* tijdens transmissieronde 12? [2]
- d) Wat is de waarde van de variabele `Threshold` tijdens de 13<sup>e</sup> transmissieronde? [2]
- e) Wat is de grootte van het *congestion window* tijdens transmissieronde 16? [2]



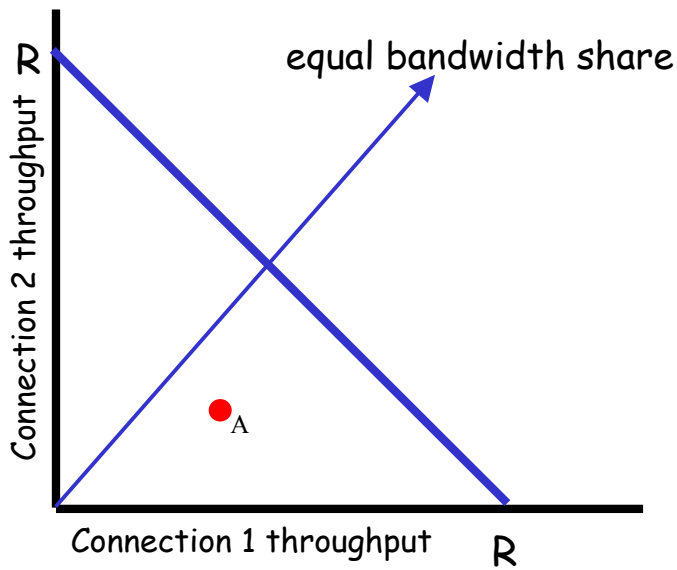
In de rest van deze opgave gaan we de *fairness* van TCP bekijken. We nemen aan dat twee TCP connecties (*connection 1 & connection 2*) een bottleneck link met transmissie *rate*  $R$  delen. De twee connecties hebben dezelfde MSS en RTT, dus wanneer ze dezelfde *congestion window* grootte hebben, hebben ze ook dezelfde *throughput*. Onderstaand diagram illustreert hoe het *Additive Increase Multiplicative Decrease* (AIMD) mechanisme van TCP uiteindelijk een eerlijke verdeling van de bandbreedte oplevert.



- f) Welke fase van de TCP *congestion control* representeert de pijl van A naar B in bovenstaande figuur? [1]
- g) Welk aspect van TCP *congestion control* representeert de pijl van B naar C? [1]

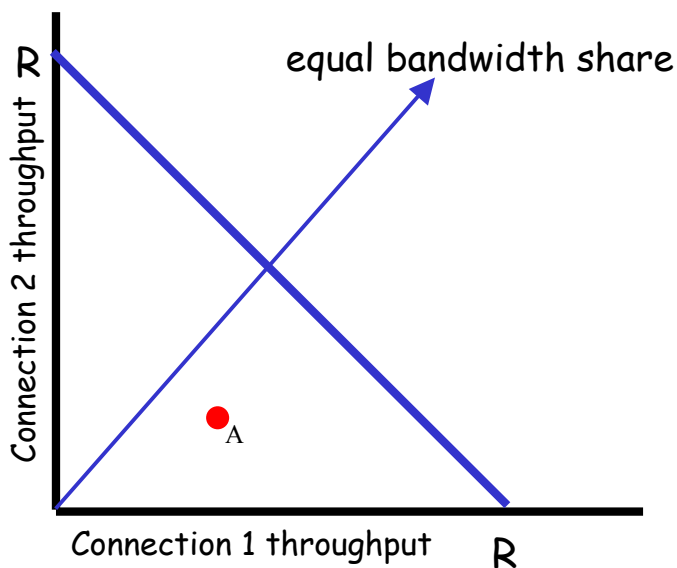
Stel je voor dat gedurende de Additive Increase connection 2 zijn window steeds met 2 maal de MSS verhoogt, terwijl connection 1 zijn congestion window met 1 maal de MSS verhoogt.

- h) Laat in onderstaand diagram zien hoe de *throughputs* (en *congestion windows*) zich ontwikkelen, uitgaande van punt A. Hoe zal de uiteindelijke bandbreedte verdeling zijn tussen connection 1 en connection 2? [2]



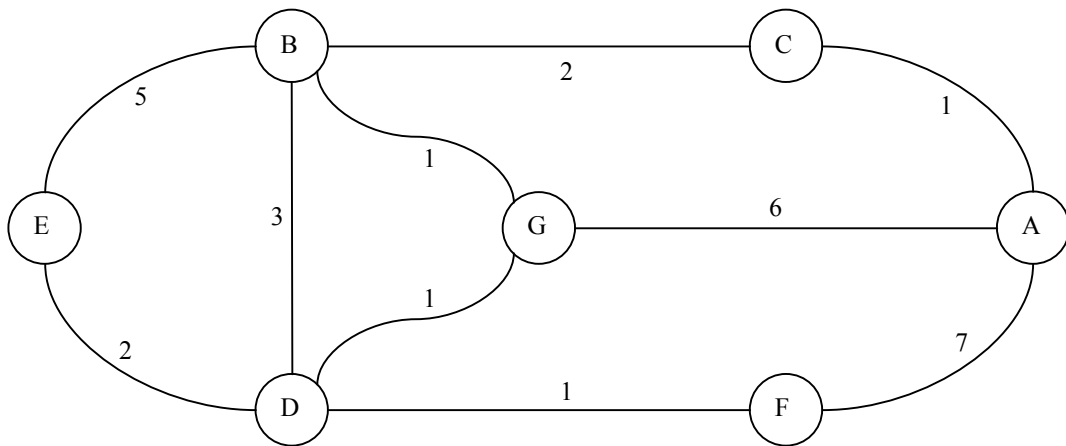
Stel dat TCP niet AIMD maar MIAD (*Multiplicative Increase Additive Decrease*) zou gebruiken.

- i) Laat in onderstaand diagram zien hoe de *throughputs* (en *congestion windows*) zich ontwikkelen, uitgaande van punt A. [2]



**6) Routing [10]**

- a) Wat is het verschil tussen routing en *forwarding*? [1]
- b) Is in een *virtual-circuit* netwerk ook routing nodig? Indien nee, waarom niet? Indien ja, waarvoor? [2]
- c) Waarom bevat de *forwarding* tabel in een *virtual-circuit* netwerk naast *Incoming VC* en *Outgoing Interface* ook kolommen voor *Incoming Interface* en *Outgoing VC*? [2]
- d) Beschouw onderstaand IP-netwerk: de knooppunten zijn de routers en de lijnen de communicatie links. Gebruik het algoritme van Dijkstra om onderstaande tabel in te vullen om zodoende het kortste pad van C tot alle andere nodes te berekenen. Hierbij is  $N$  de verzameling van al afgehandelde nodes,  $D(x)$  de afstand tot node x, en  $p(x)$  de voorganger van x op het kortste pad van C naar x. [5]



step	$N$	$D(A), p(A)$	$D(B), p(B)$	$D(D), p(D)$	$D(E), p(E)$	$D(F), p(F)$	$D(G), p(G)$
0	{C}						
1							
2							
3							
4							
5							
6							

## 7) Multiple Access [15]

In deze vraag beschouwen we verschillende *multiple access* protocollen: *channel partitioning*, Aloha en *polling*.

- a) Op welke twee manieren kan het kanaal worden onderverdeeld, in het geval van *channel partitioning*? [2]
- b) Beschrijf de basiswerking van slotted Aloha. [2]
- c) Waarom is bij gewoon (*pure*) Aloha de kans dat een frame gedurende de transmissie een collision ondervindt precies twee maal zo groot als bij slotted Aloha? [2]

Beschouw een *broadcast* kanaal met  $N = 10$  nodes (voor *polling* is er ook nog een extra *polling node*) en een *data rate* van  $R = 10^8$  bits/s. De grootte van een slot (in het geval van *channel partitioning* en slotted Aloha) is  $Q = 10^5$  bits. In het geval van *polling* is het maximum aantal bits wat verstuurd mag worden als een node *gepoll* wordt ook  $Q$  bits. Voor het *polling* protocol is de *polling delay* (de hoeveelheid tijd die verstrijkt tussen het afronden van een transmissie en de start of transmissie van de volgende node),  $d_{poll} = 1$  ms.

Stel dat slechts **één** node data te versturen heeft, terwijl de andere  $N-1$  nodes geen data te versturen hebben.

- d) Wat is de maximale *throughput* van het *broadcast* kanaal wanneer *channel partitioning* wordt gebruikt? Leg je antwoord uit. [2]

- e) Wat is de maximale *throughput* van het *broadcast* kanaal wanneer slotted Aloha wordt gebruikt? Gevraagd wordt hoeveel bits de actieve node gemiddeld per seconde zal versturen wanneer deze continu data te versturen heeft. Leg je antwoord uit. [2]
- f) Wat is de maximale *throughput* van het *broadcast* kanaal wanneer *polling* wordt gebruikt? Leg je antwoord uit. [2]

Neem nu aan dat **meerdere** nodes data te versturen hebben.

- g) Leg voor ieder van de 3 genoemde *multiple access* protocollen uit hoe welke invloed dit heeft op de totale *throughput* van het *broadcast* kanaal (zoals gevonden in respectievelijk d, e en f). Blijft deze gelijk, stijgt deze, of daalt deze? Leg je antwoord uit. (Opm.: je hoeft geen getal of formule te geven). [3]