

Naam	Initialen	Studentnummer	Geboortedatum	Studierichting

Tentamen Telematica Systemen en Toepassingen (261000)

3 november 2004

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 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.

Let op: Opgave 1 t/m 6 zijn voor alle studenten. Van opgave 7 zijn 2 versies, 7A en 7B. Studenten die TST herkansen hebben de **keuze** het tentamen te doen op basis van de stof zoals die in 2003/2004 gegeven werd (2^e editie van het boek van Kurose en Ross). In dat geval dienen zij opgave 7B te maken in plaats van opgave 7A, en dat hieronder aan te kruisen. Slechts één van beide opgaven wordt nagekeken.

Ik ben herkanser, en ik wil dat opgave 7B wordt nagekeken ipv opgave 7A.

Alle andere studenten worden geacht opgave 7A te maken, en **niet** 7B.

Alleen voor de docent:

Vraag	1	2	3	4	5	6	7	Totaal
Punten								
Maximum	10	12	10	15	12	15	15	89

Lijst van afkortingen:

DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
FDM	Frequency-Division Multiplexing
FEC	Forward Error Correction
GBN	Go-Back- <i>N</i>
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IP	Internet Protocol
LAN	Local Area Network
MAC	Medium Access Control
MSS	Maximum Segment Size
NAT	Network Address Translat(ion/or)
RTT	Round Trip Time
SR	Selective Repeat
TCP	Transmission Control Protocol
TDM	Time-Division Multiplexing

1) Algemene vragen [10]

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

- a) De transmissietijd van een pakket hangt af van de propagatiesnelheid van het transmissiemedium.
waar / niet waar
- b) Door statistisch *multiplexing* kunnen meer gebruikers goed ondersteund worden.
waar / niet waar
- c) Een satelietverbinding naar de VS leidt tot betere servicekwaliteit (*quality of service*) dan een traditionele zee kabel.
waar / niet waar
- d) De *queueing delay* in een *router* is een lineaire functie van de verkeersintensiteit in de *router*.
waar / niet waar
- e) In gelaagd gestructureerde protocolsystemen worden sommige operaties regelmatig meervoudig uitgevoerd hetgeen tot efficiëntieverlies leidt.
waar / niet waar
- f) Met Ethernet kun je een maximale *throughput* van 37% van de transmissiesnelheid behalen.
waar / niet waar
- g) Door zijn *soft-state* eigenschappen heeft het DHCP protocol een *plug-and-play* functionaliteit.
waar / niet waar
- h) *Time-division multiplexing* (TDM) met N slots gecombineerd met *frequency-division multiplexing* (FDM) met M frekwenties leidt tot $M \cdot N$ logische kanalen.
waar / niet waar
- i) *Forward error correction* (FEC) leidt vaak tot extra hertransmissies.
waar / niet waar
- j) Op basis van Ethernet kan een efficiënt *token-passing* protocol worden gebouwd.
waar / niet waar

2) HTTP en DNS [12]

Stel dat je met de *Web browser* klikt op een URL om een *Web page* te verkrijgen. Stel dat het IP adres behorend bij die URL niet bewaard (*not cached*) is in jouw locale *host*, zodat een *DNS query* nodig is. Stel dat er één *DNS server* wordt geraadpleegd voordat jouw *host* het IP adres van de *DNS server* ontvangt. Deze *DNS query* heeft een RTT (Round Trip Time) nodig die gelijk is aan RTT_1 (seconden). Stel dat de lengte van elk DNS bericht verwaarloosbaar klein is.

Stel dat de *Web page* die geassocieerd is met die URL een *HTML file* (met lengte L bytes) bevat die verwijst naar vijf objecten (elk met een lengte van L bytes) op dezelfde *server*. L is kleiner dan TCP's MSS, zodat TCP's *window* mechanismen niet beperkend zijn. Laat RTT_0 (seconden) de waarde van de RTT zijn, die nodig is tussen de locale *host* en de *server* die de objecten bewaart. Stel dat de transmissie snelheid R bps is. Hoeveel tijd is dan nodig (vanaf het moment dat je op de URL klikt totdat je alle vijf objecten hebt ontvangen) met:

a) *nonpersistent* HTTP **zonder** parallelle TCP connecties? [3]

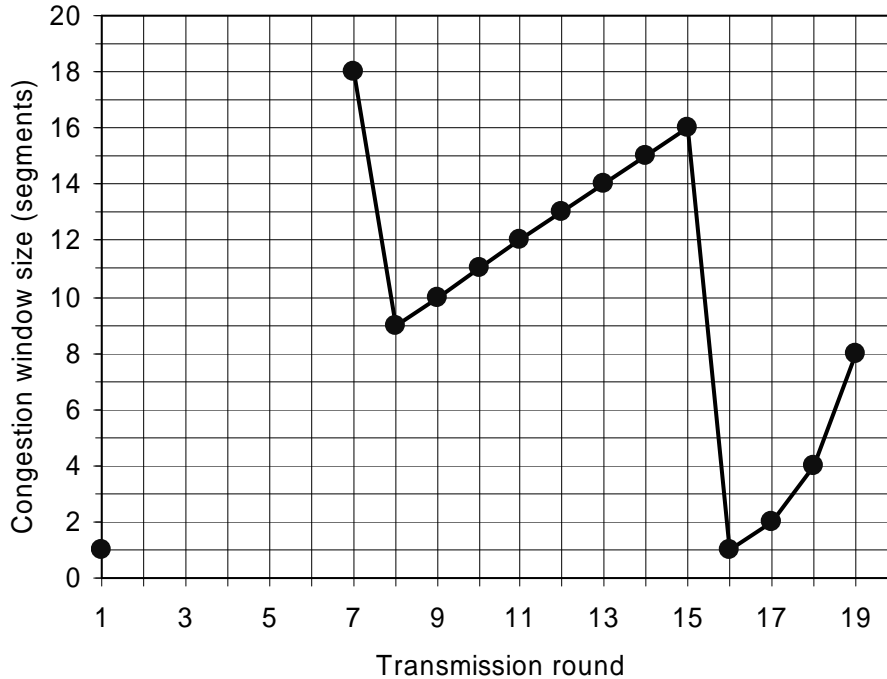
b) *nonpersistent* HTTP **met** parallelle TCP connecties? [3]

c) *persistent* HTTP **zonder** *pipelining*? [3]

d) Wat is dan de effectieve *throughput* bij gebruik van *persistent* HTTP **zonder** *pipelining*? [3]

3) TCP Congestion Control [10]

Beschouw het volgende verloop van de grootte van het TCP *congestion window* als functie van de tijd. De gebruikte versie van TCP is Reno (TCP met *fast retransmit* en *fast recovery*). De grootte van het congestion window van transmissieronde 2 t/m 6 en vanaf transmissieronde 20 is (nog) niet ingevuld in de grafiek.



- a) Gedurende welk(e) tijdsinterval(len) tussen transmissieronde 7 en 19 werkt *congestion avoidance*? Gedurende welk(e) tijdsinterval(len) *slow start*? [2]

congestion avoidance:

slow start:

- b) Wordt na de 7^e transmissieronde pakketverlies gedetecteerd door het ontvangen van 3 *duplicate ACKs*, of door een *time-out*? En na de 15^e transmissieronde? [2]

na de 7^e transmissieronde:

na de 15^e transmissieronde:

- c) Wat zal de grootte van het *congestion window* geweest zijn gedurende de 4^e transmissieronde? En gedurende de 6^e transmissieronde? [2]

gedurende de 4^e transmissieronde:

gedurende de 6^e transmissieronde:

- d) Wat kun je zeggen over de initiële waarde van de variabele *Threshold*? [2]

- e) Na transmissieronde 19 vindt **geen** pakketverlies plaats. Wat is de grootte van het *congestion window* tijdens transmissieronde 20? [2]

4) TCP Throughput [15]

In deze vraag bestuderen we TCP's *additive increase, multiplicative decrease congestion control* mechanisme. Stel, een groot bestand wordt verstuurd met TCP en de grootte van het *congestion window* w varieert tussen C en $C/2$ segmenten. Hierbij is C de kleinste *congestion window* grootte waarbij pakketverlies plaats vindt. Alle segmenten worden verstuurd met dezelfde grootte van *MSS (Maximum Segment Size)* bytes. De *Round Trip Time, RTT* is constant. Om de berekeningen simpel te houden mag je er van uit gaan dat C een even getal is.

a) Druk de snelheid waarmee TCP data verstuurt uit in *RTT, MSS* en windowgrootte w . [2]

We definiëren een *Systeem Cyclus* als de periode die begint met de *multiplicative decrease* van het *congestion window*, dus $(w = C) \rightarrow (w = C/2)$, en eindigt na de transmissieronde waarin w weer gelijk is aan C , en een pakketverlies optreedt. Gedurende een *Systeem Cyclus*, wordt w iedere transmissieronde (*RTT*-periode) met 1 opgehoogd.

b) Hoeveel transmissierondes bevat een *Systeem Cyclus*? [1]

c) Laat zien dat het aantal segmenten dat verstuurd wordt in één *Systeem Cyclus* gelijk is aan $(3/8) \cdot C^2 + (3/4) \cdot C$. [3]

Hint: reken het aantal segmenten uit dat verstuurd wordt in de 1^{ste} transmissieronde, de 2^e transmissieronde, enz., t/m de laatste transmissieronde. Gebruik verder dat $1 + 2 + \dots + n = (n + 1) \cdot n/2$.

d) Wat is de *bit rate* waarmee TCP data verstuurt, gemiddeld over één *Systeem Cyclus*? [2]

De *loss rate* L is gedefinieerd als het aantal verloren segmenten gedeeld door het totaal aantal segmenten dat door de zender verstuurd wordt. Neem aan dat in iedere *Systeem Cyclus* precies 1 segment verloren gaat.

e) Waardoor wordt het verlies van zo'n segment veroorzaakt? Geef aan hoe het mogelijk is dat meer dan 1 segment per *Systeem Cyclus* verloren gaat. (Wij gaan er verder vanuit dat precies 1 segment per *Systeem Cyclus* verloren gaat). [2]

f) Druk de *loss rate* L uit in C en laat zien hoe deze voor grote waarden van C benaderd kan worden door $L = 8/(3 \cdot C^2)$. [2]

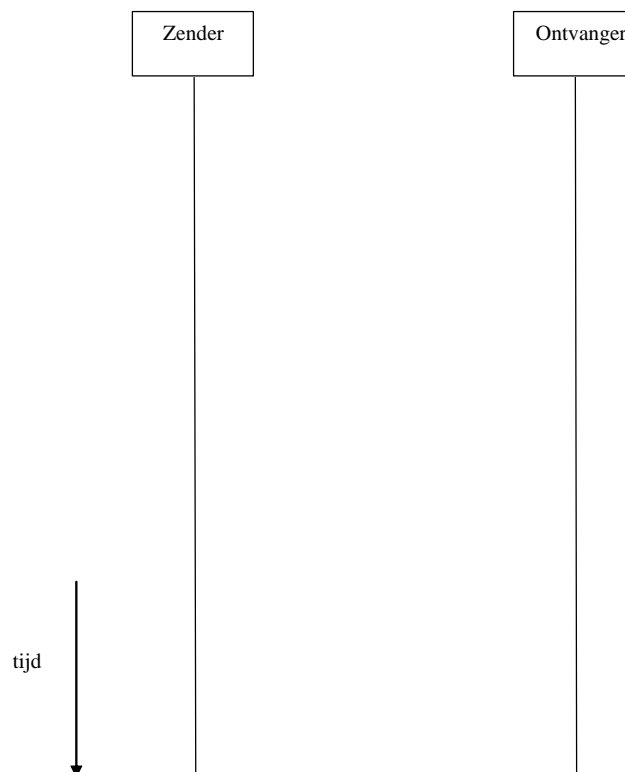
g) Gebruik deze benadering om te laten zien dat de gemiddelde *bit rate* van TCP (zoals berekend in (d)) benaderd kan worden door de uitdrukking $\frac{1.22 \cdot MSS}{RTT \cdot \sqrt{L}}$. [3]

5) Selective Repeat en Go-Back-N [12]

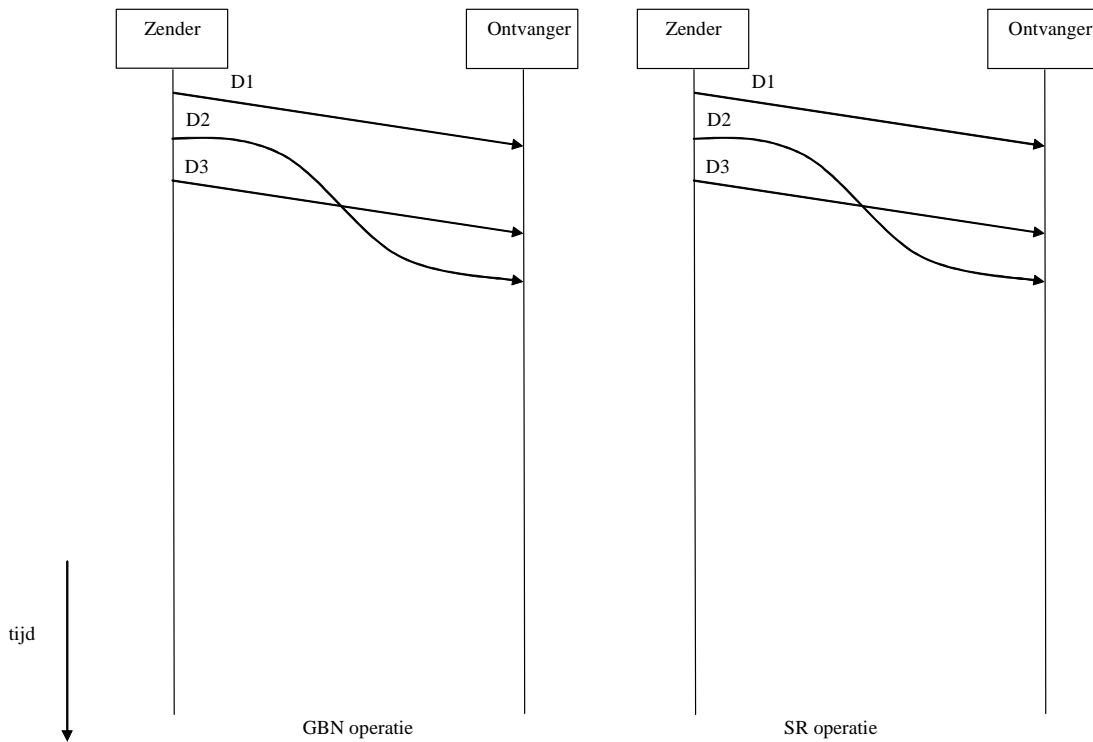
- a) Beschrijf de situatie waarin het gedrag van het *alternating-bit* protocol, *Selective Repeat* (SR) en *Go-Back-N* (GBN) precies gelijk is. [2]
- b) Beschrijf wanneer en hoe SR efficiënter dan GBN kan zijn. [2]

Stel, voor de volgende delen van deze opgave, dat de SR en GBN protocollen gebruikt worden met een zender en een ontvanger *window* met grootte 3.

- c) Laat in het volgende *time-sequence* diagram een situatie zien waarin het gedrag van SR en van GBN identiek is. Het diagram moet het versturen en ontvangen van de data (*D*) en de *acknowledgements* (*A*) laten zien. [2]

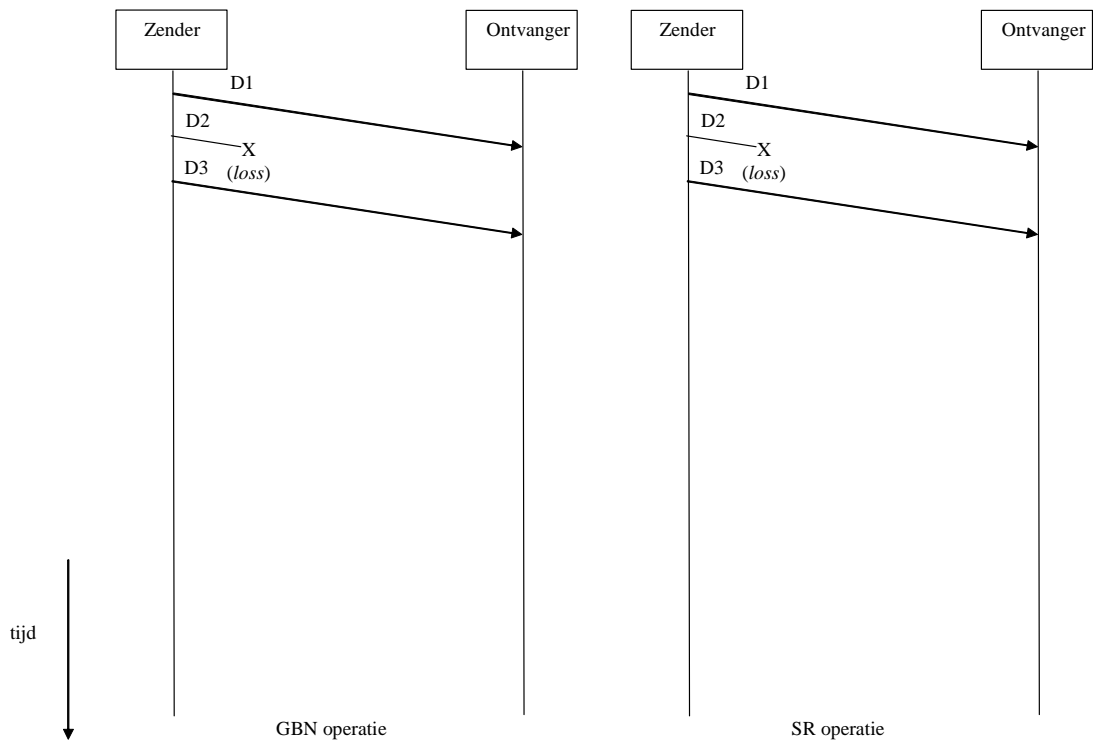


- d) Gebruik het volgende *time-sequence* diagram. Merk op dat het datapakket *D2* arriveert in de ontvanger nadat *D3* is ontvangen. Schets de *acknowledgements* (*A*) met hun *sequence* nummers en, wanneer nodig, de hertransmissies van de datapakketten (*D*) na een *time-out*, in de situatie dat:
- (1) GBN wordt gebruikt [1½]
 - (2) SR wordt gebruikt [1½]



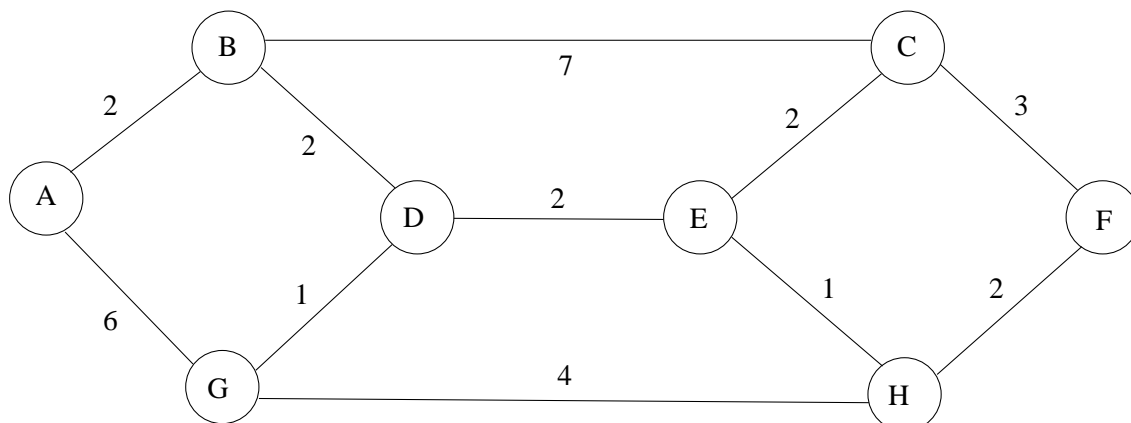
e) Gebruik het volgende *time-sequence* diagram. Merk op dat het datapakket *D2* verloren gaat en niet arriveert in de ontvanger. Schets de *acknowledgements* (*A*) met hun *sequence* nummers en, wanneer nodig, de hertransmissies van de datapakketten (*D*) na een *time-out* in de situatie dat:

- (1) GBN wordt gebruikt [1½]
- (2) SR wordt gebruikt [1½]



6) Routing [15]

We beschouwen het volgende netwerk met 8 routers (A t/m H) en link-kosten zoals aangegeven bij de verbindingen (*links*).



- a) Bereken, met behulp van het algoritme van Dijkstra het kortste pad van A naar alle andere routers; vul daartoe onderstaande tabel leesbaar en volledig in. In deze tabel is N' de verzameling van al afgehandelde nodes, $D(x)$ de afstand van node A tot node x, en $p(x)$ de voorganger van x op het kortste pad van A naar x. [7]

step	N'	$D(B), p(B)$	$D(C), p(C)$	$D(D), p(D)$	$D(E), p(E)$	$D(F), p(F)$	$D(G), p(G)$	$D(H), p(H)$
0	{A}	2,A	∞	∞	∞	∞	6,A	∞
1								
2								
3								
4								
5								
6								
7								

- b) Na enige tijd blijkt dat in bovenstaand netwerk de link (D,E) niet meer te functioneren. Wat wordt in dat geval de kortste route van A naar de nodes C, E, F en H. Geef voor iedere route alle knooppunten aan, in de notatie (als voorbeeld) A-K-L-M-C. [2]

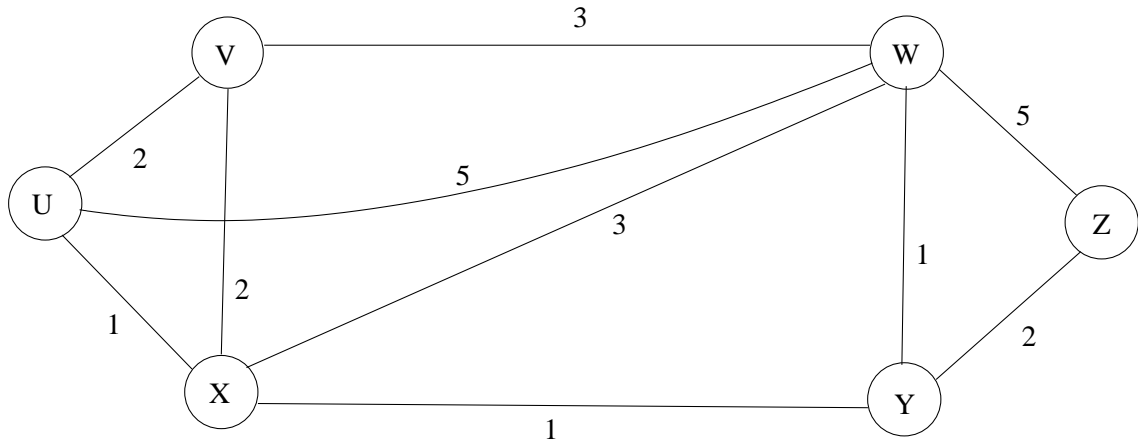
Van A naar C:

Van A naar E:

Van A naar F:

Van A naar H:

Voor de deelvragen (c) en (d) bekijken we het volgende netwerk:



Analyse met behulp van het algoritme van Dijkstra levert de volgende "laatste regel" op voor het kortse pad vanuit node U naar alle andere nodes:

D(V), p(V)	D(W), p(W)	D(X), p(X)	D(Y), p(Y)	D(Z), p(Z)
:	:	:	:	:
2,U	3,Y	1,U	2,X	4,Y

- c) Stel dat node U per tijdseenheid 10 pakketten verstuurt naar alle andere nodes (dus 10 van U naar V, 10 van U naar W, 10 van U naar X, etc.). Hoeveel pakketten moeten dan alle routers per tijdseenheid verwerken? Vul hiertoe de volgende tabel in: [2]

router	V	W	X	Y	Z
pakket/tijd					

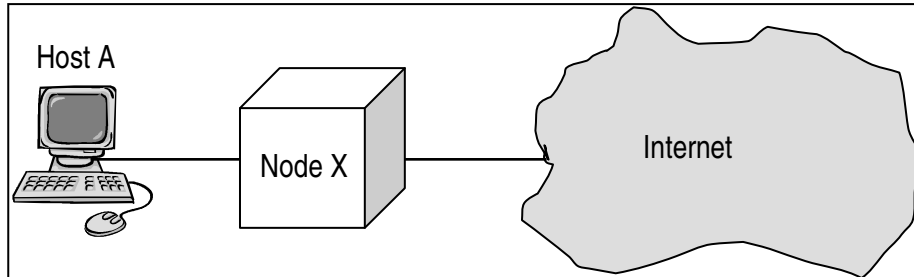
- d) Stel dat node U per tijdseenheid 10 pakketten verstuurd naar alle andere nodes (dus 10 van U naar V, 10 van U naar W, 10 van U naar X, etc.). Hoeveel pakketten per tijdseenheid hebben de volgende links dan te verwerken? [2]

link	UV	UW	UX	VW	VX	WY	WZ	XY	YZ
pakket/tijd									

- e) Wat lijkt dus een nadeel van kortste-pad routing te zijn? [2]

7A) Adressering [15]

Beschouw de volgende configuratie. *Host A* is via een andere node (*Node X*) verbonden met het Internet. Stel *Host A* verstuurt pakketten naar het Internet via *Node X*. Op de link tussen *Host A* en *Node X* zullen die pakketten *header* informatie bevatten om ze uiteindelijk bij het juiste proces op de juiste *host* te kunnen afleveren. De belangrijkste hiervan zijn *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. [3]

Source IP Address en *Destination IP Address*:

Source MAC Address en *Destination MAC Address*:

Source Port Number en *Destination Port Number*:

- b) We bekijken vier verschillende mogelijkheden voor *Node X*. We beschouwen de gevallen dat *Node X* een **hub**, een **switch**, een **router** of een **NAT** is. Gevraagd wordt in onderstaande tabel voor ieder van deze gevallen aan te geven welke *header* informatie wel of niet verandert in *Node X*, wanneer *Node X* een pakket van *Host A* doorstuurt naar het Internet. Omcirkel op iedere plaats waar “wel/niet” staat het woord dat van toepassing is (dus in totaal 24 maal). [6]

Verandert dit veld in de <i>header</i> wanneer <i>Node X</i> een is?	Hub	switch	router	NAT
<i>Source IP Address</i>	wel / niet	wel / niet	wel / niet	wel / niet
<i>Destination IP Address</i>	wel / niet	wel / niet	wel / niet	wel / niet
<i>Source MAC Address</i>	wel / niet	wel / niet	wel / niet	wel / niet
<i>Destination MAC Address</i>	wel / niet	wel / niet	wel / niet	wel / niet
<i>Source Port Number</i>	wel / niet	wel / niet	wel / niet	wel / niet
<i>Destination Port Number</i>	wel / niet	wel / niet	wel / niet	wel / niet

We definiëren de pakket *delay* van Node X als de tijd die verstrijkt tussen het ontvangen van het laatste bit van een pakket in Node X en het versturen van het laatste bit van hetzelfde pakket door Node X. Veronderstel dat het netwerk onbelast is. Er zijn dus geen andere pakketten onderweg op de links tussen *Host A* en het Internet.

c) Geef aan of en waarom er (geen) verschil is in de pakket *delay* van een *switch* en een *router*. (Dus tussen de gevallen dat Node X een *switch* is en een *router*.) [1½]

d) Geef aan of en waarom er (geen) verschil is in de pakket *delay* van een *switch* en een *hub*. [1½]

Veronderstel nu dat het netwerk belast is. Er zijn dus andere pakketten onderweg op de links tussen *Host A* en het Internet.

e) Geef aan of en waarom er (geen) verschil is in de pakket *delay* van een *switch* en een *router*. [1½]

f) Geef aan of en waarom er (geen) verschil is in de pakket *delay* van een *switch* en een *hub*. [1½]

7B) Code Division Multiple Access [15]

Alleen voor herkansers (ipv opgave 7A). Op het voorblad aankruisen wanneer je deze opgave maakt, en wanneer deze nagekeken moet worden ipv 7A.

We beschouwen 4 zender-ontvanger paren (A, B, C en E) die over een draadloos medium communiceren met een datasnelheid van 4 Mbit/s, en daarbij gebruik maken van CDMA. We gebruiken de volgende afkortingen:

ZA: zender-A, OA: ontvanger-A,

ZB: sender-B, OB: ontvanger-B,

ZC: zender-C, OC: ontvanger-C.

ZD: zender-D, OD: ontvanger-D

ZA en OA maken gebruik van chipping-code A: -1 -1 +1 -1 +1 +1 +1 -1;

ZB en OB maken gebruik van chipping-code B: -1 +1 -1 +1 +1 +1 -1 -1;

ZC en OC maken gebruik van chipping-code C: -1 +1 -1 -1 -1 -1 +1 -1;

ZD en OD maken gebruik van chipping-code D: -1 -1 -1 +1 +1 -1 +1 +1;.

- a) Wat is in dit geval de gehanteerde *chipping rate*? [1]

- b) ZA verstuurt een '1' aan OA; welk signaal, dwz, welke bit-sekwentie, wordt daadwerkelijk verstuurd? [2]

- c) Welk signaal wordt verstuurd wanneer ZA, ZB, ZC en ZD **tegelijktijd** resp. een '1', een '0', een "1" en een '1' versturen? Beschrijf je berekening in detail. [3]

- d) Laat zien hoe OC het ontvangen signaal gebruikt om te 'decoderen' dat in dit geval ZC inderdaad een '1' heeft verstuurd. Beschrijf je berekening in detail. [3]

- e) OA ontvangt de sekwentie -2 0 0 0 2 2 0 -2. Wat heeft ZA verstuurd? [3]

- f) OB ontvangt de sekwentie 0 0 -2 +2 0 -2 0 +2. Wat heeft ZB verstuurd? [3]