U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

MIEEC | 4º ANO

EEC0033 | REDES DE COMPUTADORES | 2015/16 - 1º SEMESTRE

Memory aids allowed. Duration: 90 minutes.

Second part

Name: ....

1. You want to develop an Automatic Repeat Request (ARQ) protocol for connections in a high performance system. You need to connect 1) two cores within a multicore chip using a 3 µs/km, 2 cm link and 2) two multicore chips on the same system using a 4 µs/km, 20 cm link. Packets are 400 bits long and the link capacity is 1 Tbps.

   a. **(1.5 / 20)** What is the efficiency and maximum throughput of a Stop and Wait protocol on these links? Assume no errors.

Stop and Wait	L1 (within chip, 2cm)	L2 (between chips, 20cm)
Tp	60 ps	0.8 ns
Tf	0.4 ns	0.4 ns
a	0.15	2
Efficiency (%)	76.9%	20.0%
Maximum throughput	769 Gbps	200 Gbps

Tp_1 = 0.02 [m] * 3µs / km = 2E-2 * 3E-6 [m.s] / 1E3[km] = 6E-11 s = 60 ps

Tp_2 = 0.2 [m] * 4µs / km = 2E-1 * 4E-6 [m.s] / 1E3[km] = 8E-10 s = 0.8ns

T_f = Tf_1 = Tf_2 =400 [bit/packet] / 1E12 [bit/s] = 4E-10s = 0.4 ns

a_1 = Tp_1 / Tf = 0.15

a_2 = Tp_2 / Tf = 2

S_1 = 1/(1+2*a_1) = 1/1.3 =76.9%

S_2 = 1/(1+2*a_2) = 1/5 =20.0%

   b. **(1.5 / 20)** Given 3 bits for frame numbering, what is the efficiency of a Go-Back-N and of a Selective Reject protocol for the 20 cm connection between multiprocessor chips? First assume no errors, then consider a bit error rate of 0.03%.

Between chips (20cm)	Go-Back-N	SRJ
Efficiency (%), no errors	100%	80%
FER	11.3%	11.3%
Efficiency (%), with errors	61%	71%

GBN:
W = 2^3-1 = 7

W_max = 1+2*a_2=5

SRJ:
W = 2^(3-1) = 4

Without errors

GBN:
W > W_max = > S=100%

SRJ:
W < W_max = > S=W/(1+2*a_2)=4/5

With errors

FER = 1 - (1 - 3E-4)^400=11.3%

GBN:
S = (1 - 0.113)/(1+2*a_2*0.113) =61%

SRJ:
S = 4 (1 - 0.113) / (1+2*a_2) = 71%

c. **(1 / 20)** How many bits would we need for frame numbering to achieve the maximum efficiency of the SRJ protocol for the 20 cm connection between multiprocessor chips assuming the same bit error rate of 0.03%?

Between chips (20cm)	**SRJ**
Number of bits for frame numbering under maximum efficiency	4
Maximum efficiency (%)	89%

Max. efficiency SRJ => $W= 2^(n-1) >= 1+2*a_2 =5$

n=3 => W=4 < 5; not ok

n=4=> W=8 > 5; ok

S_max_SRJ = 1 - FER = 89%

2. Consider that the system from question 1 is expanded as follows. Each chip has 4 cores: 3 of the 4 cores are connected via 2 cm links directly to the 4th core, which will be queuing and multiplexing the packets out of the chip. The multiplexing core connects via the 20 cm link to a de-multiplexing core on the other chip. You want to use queuing theory and in particular the M/M/1 queue to analyze the performance of this system.

   a. **(1.5 / 20)** Assume that three cores generate on average 250 Gbit/s of traffic each. How much traffic on average can the fourth core generate such that the **average delay in the system** does not exceed 4 ns?

Average traffic from the 4 nodes (Gbit/s)	900 Gbps
Average traffic from the 4th node (Gbit/s)	150 Gbps

The service rate for the 20 cm output link to the other chip is:
μ = 1E12 [bit/s] / 400 [bit/packet] = 2.5E9 [packet/s]

Average delay in the system T = N/λ =λ/(μ-λ).1/λ =1/(μ-λ) and must not exceed 4ns. Thus:
1/(μ-λ) <= 4ns

This means that average traffic λ entering the 20 cm output link must not exceed:
λ <= μ - 1/(4ns) =2.5E9 - 0.25E9=2.25E9 [packet/s]

Traffic from the 4 nodes enters the output link; so the average traffic from the 4 nodes must not exceed:
2.25E9 [packet/s] * 400 [bit/packet] = 900 Gbps

Traffic from the first 3 nodes is fixed to 250 Gbps each so the remaining for the fourth node is:
900 - 3*250 = 150 Gbps

Name: ....

b. **(1 / 20)** Now consider that we have another, similar chip but where the output traffic that the cores on this chip generate is the following: 600, 300, 50, 50 (Gbit/s). The capacity of the 20 cm link is not enough to carry this traffic out of the chip. Why?

Although the capacity of the link is 1 Tbps which equals the 600+300+50+50 Gbps traffic input, this traffic input is the average of an exponential distribution. Applied to the M/M/1 queue, this traffic will yield an infinite average delay on the link, which is not practical at all. With $\mu=\lambda$ then $T = 1/(\mu-\lambda) = \infty$.

c. **(1.5 / 20)** If we add a 20 cm link to carry the traffic out of the chip to double chip interconnection capacity, how would you assign the traffic from each of the cores to each of the output links such that the **average packet delay in the system** is smallest?

	T (ms)		
Options (Gbit/s)	C_{out}^1	C_{out}^2	Average
C_{out}^1: 600 ; 50 C_{out}^2 : 300 ; 50	1.14 ns	0.62 ns	0.99 ns
C_{out}^1: 300 ; 50 ; 50 C_{out}^2 : 600	0.67 ns	1.00 ns	0.87 ns

$T = 1/(\mu-\lambda)$ = L / (C - R), with L=400 [bit/packet], C=1 Tbps is the link capacity, and R is the average traffic in bit/s.

T_11 = 400/(1000-650)E-9 = 1.14 ns

T_12 = 400/(1000-350)E-9 = 0.62 ns

T_1 = (T_11 * 650 + T1_2*350)/1000=0.99 ns

T_21 = 400/(1000-400)E-9 = 0.67 ns

T_22 = 400/(1000-600)E-9 = 1.00 ns

T_2 = (T_21 * 400 + T2_2*600)/1000=0.87 ns

3. Each of the cores has been assigned an IP address. Consider a system with multiple chips where each chip has its own IP network. Network addresses are assigned consecutively according to chip id: the first address of chip C1's network is the address immediately after the last address of chip C0's network. IP addresses for cores are assigned using the same logic: core 0 uses the first IP address in the chip's network, core 1 the following, and so forth. The first address used in this system should be 10.0.0.0.

   a. **(1 / 20)** Fill in the table for the network to assigned to chip C4. Use the smallest possible number of addresses per network.

Network and mask addresses (255.255.255.255 format)	Network: 10.0.0.32 Network mask: 255.255.255.248
Broadcast address	10.0.0.39
Number of available addresses for network interfaces	2^3 - 2=6 available addresses for NICs

Note: Each chip network needs 4 NICS (one for each core) + Network +Broadcast = 6 addresses => 3 bits => /(32-3) = /29

b. **(1 / 20)** The last two addresses of each chip's network are reserved for gateways. The address before the last is the gateway for accessing the network of the next chip. The last address is the default gateway. Chips C2 and C7 are disabled and entries referring to these networks have been purged from all routing tables. What is the ARP response that core 2 in chip C4 obtains when it tries to do the following pings?

`ping 10.0.0.9`	core 0 in chip C1 => MAC of default GW (10.0.0.38)
`ping 10.0.0.33`	core 0 in chip C4 => MAC of core 0, chip 4 (10.0.0.33)
`ping 10.0.0.35`	core 2 in chip C4 => ARP not issued, this is the localhost
`ping 10.0.0.43`	core 2 in chip C5 => MAC of next chip GW (10.0.0.37)
`ping 10.0.0.58`	core 1 in chip C7 => MAC of default GW (10.0.0.38), ping unsuccessful
`ping 11.3.1.4`	some address on the private network => MAC of default GW (10.0.0.38)

c. **(1 / 20)** A system with 192 of these chips is being deployed in a datacenter. A /21 address space is available to create sub-networks to make chip management easier. The following options are being considered. Check if it is possible to organize this address space with the following number of chips per sub-network, and if it is possible, define the corresponding sub-network addresses.

Option 1: 100, 50, 30, 12
Option 2: 90, 40, 40, 22

	Possible?	Sub-networks (address and /x; start on 10.0.0.0)
Option 1	YES allocation within /21 range	100 (800 IPs) => /22, 10.0.0.0/22, 10.0.0.0=>10.0.3.255 50 (400 IPs) => /23, 10.0.4.0/23, 10.0.4.0=>10.0.5.255 30 (240 IPs) => /24, 10.0.6.0/24, 10.0.6.0=>10.0.6.255 12 (96 IPs) => /25, 10.0.7.0/25 => 10.0.7.0=> 10.0.7.127
Option 2	NO allocation out of /21 range	90 (720 IPs) => /22, 10.0.0.0/22, 10.0.0.0 => 10.0.3.255 40 (320 IPs) => /23, 10.0.4.0/23, 10.0.4.0=>10.0.5.255 40 (320 IPs) => /23, 10.0.6.0/23, 10.0.6.0=>10.0.7.255 22 (176 IPs) => /24, 10.0.8.0/24, 10.0.8.0=>10.0.8.255 (out of range)

Note: the range for 10.0.0.0/21 is 10.0.0.0 => 10.0.7.255

Note: chip networks are /29 so each chip uses 8 addresses from its subnet range