Home assignment sets 3 and 4 – Solutions

Set 3: Congestion control

Problem 1

Consider an AIMD (additive increase multiplicative decrease) scheme, similar to the one addressed in the lecture, with additive increase parameter b and multiplicative increase parameter a. Consider constant rtt and p, and assume that packet loss happens always at the same window size, as we assumed in the model presented on the lecture. Express the throughput as a function of parameters $a = (0, 1)$, and $b = (0, 1)$, end evaluate the effect of these parameters on the throughput.

What are the advantages and drawbacks to select large (or small) values for a and b?

Let us perform exactly the same steps of modeling as in class, but leaving parameters a and b as parameters.

$$
\begin{aligned}
Th &= \frac{(N-1)L}{T_0} \\
N &= \frac{1}{p} \\
T_0 &= \frac{(1-a)}{b} W_m rtt \\
N &= (\frac{a(1-a)}{b} + \frac{(1-a)^2)}{2b}) W_m^2 = \frac{1-a^2}{2b} W_m^2 \\
Th &= \frac{(N-1)L}{T_0} =\approx \sqrt{\frac{b(1+a)}{2(1-a)}} \frac{1}{\sqrt{p}} \frac{L}{rtt}.
\end{aligned}
$$

Consequently, the throughput increases with a and with $\sqrt{b}$. For the original parameters $a = 0.5$ and $b = 1$ we get the original constant $\sqrt{3/2}$.

The throughput increases with a constant multiplier, but the characteristics remains the same in terms of dependence on loss and round trip time. If the multiplicative decrease parameter a is too high, then the source does not decrease its rate significantly, which in real systems may mean that congestion is not avoided by this back off. It is important to note, this is not a problem in the model considered above, when congestion happens at the same window size all the time. If it is too low, more resources are lost. Similarly, larger b would increase the throughput, but in a real system may bring the network quickly in sever congested state.

Problem 2

Consider the TCP paper you have read at home. It contains three, increasingly complex models of TCP throughput. Assume you need to use a TCP model for some analysis. How can you decide which one of the models to select?

The three models presented, in the order of increasing complexity, consider triple-duplicate ACKs (TD) and then as well timeouts (TO) for loss indication, and finally the limit on the receiver window size.

We look at Figure 7 in the paper, and also read Section III with the analysis of the measurement results.

Considering the full model, from the results it seems that the receiver window size limits the rate only when this window size is very small. Therefore, in most of the practical cases, the first or the second model may be accurate enough. (The effect of window size is reflected by the large gap between the TD and the full model curves.)

Considering the need for the TO model, we see that at small p values the analytic results are close to each other, in this case the basic, TD model could work well, simplifying the analysis significantly.

Finally, we can note, that the TD only model always gives a best case, so could be used to derive upper bounds on the throughput.

Problem 3

*a) Consider a traffic flow that is rate controlled with a Leaky Bucket (LB) controller. The controller is designed to limit the average transmission rate and the maximum burst size. Assume, that packets are of one unit, and the rate controller controls the number of packets to be transmitted. Consider LB parameters $b = 3$, $r = 0.25$. Give the traffic envelope of the LB. Assume, that at time zero, the token buffer is full (stores **3** tokens), and the source generates 6 packets to transmit. When are these packets sent to the transmit buffer of the rate controller? When are they transmitted if the link capacity is one packet per time unit?*

The traffic envelope for $b = 3$, $r = 0.25$ is:

$$b(t) = b + rt = 3 + 0.25t.$$

That is, 3 out of the 6 are transmitted to the transmit buffer at time 0. The rest of the packets are transmitted when a new token arrives. Now, one could discuss when these new tokens arrive. The traffic envelope actually assumes a "fluid mode" for the tokens, which means the first token arrives at time 4, etc. This means that the rest of the packets are transmitted at $t = 4, 8, 12$. From the transmit buffer one packet can be transmitted in one time slot, that is, packets are finally transmitted from the node at times $t = 0, 1, 2, 4, 8, 12$.

b) When are the same 6 packets transmitted, if the LB controls the peak and the average rate, the peak rate is 0.5 *and the average rate is* 0.25*. Draw the traffic envelope of the LB and give the point of times when the packets are transmitted.*

The traffic envelope for $p = 0.5$, $r = 0.25$, $b = 3$ is:

$$b(t) = \min(pt, b + rt) = \min(0.5t, 3 + 0.25t).$$

This gives a break point at $0.5t = 3 + 0.25t$, that is $t = 12$. Below that the peak rate controller limits the rate, above that the average rate controller. This gives packet transmission times to the transmit buffer, as well as transmission times from the transmit buffer as $t = 2, 4, 6, 8, 10, 12$.

Set 4: Scheduling

Problem 1

Consider two flows sharing a link. The packet sizes are one unit, and the link can transmit one packet in one time unit. Give the scheduling under GPS and PGPS for the following cases.

a) Both of the flows generate bursty traffic, both of them generating 4-4 packets at times 0,8,16, etc. The weights of the flows are the same.

b) The same packet generation, but now the weights of the flows are 2:1. How are the packets served now? What is the effect of the weights?

c) Again, the two flows generate 4-4 packets, but now at times 0,6,12, etc. The weights are 2:1 again. What is the effect of weights now?

See handwritten page.

Problem 2

This problem aims at providing an example for the M/M/1-PS queue.

Consider a wireless access point (or base station) that shares downstream bandwidth equally among requests.The transmission rate is C=10Mbit/s, and no loss happens due to the bandwidth sharing. Large file downloads are initiated randomly by a large population, the file sizes are considered to be exponential. The average file size is L=1MByte. Assume, file downloads are initiated with a rate of λ=0.5 per second. Answer the following questions:

- How much time does it take in average to download a file, if noone else is downloading?

In this case the entire bandwidth is used for a single download.

$$T(1) = L/C = 4/5 = 0.8 sec.$$

- Give the Markov Chain of the system.

Let define the state of the system by the number of concurrent downloads. The Markov chain of the system is given by the state transitions. For state $i \leq 0$ $\lambda_i = \lambda$. Let μ give the service rate when only one file is downloaded, $\mu = C/L$. At state i an individual file is downloaded with rate μ/i, but the first finished download triggers state change, and therefore, for $i > 0$ $\mu_i = i(\mu/i) = \mu$. That is, the MC is like the MC of an M/M/1 queue. However, do not forget, that the meaning of the states are different for the two queues!

- What is the probability that the network is empty?

Since this is a single server, lossless system,

$$P(empty) = 1 - P(busy) = 1 - \lambda(1/\mu) = 1 - \rho = 1 - 0.4 = 0.6$$

- What is the mean number of concurrent downloads and time to download a file?

Since the state defines the number of concurrent downloads, by definition

$$N = \sum_{i=0}^{\infty} i p_i = \rho/(1-\rho) = 4/6.$$

We can calculate the mean service time using Little's formula (note it is hard to calculate this directly, since the state of the system changes during a download time. Using Little's formula we get

$$T = N/\lambda = 4/3 sec.$$

As for sanity check, it should be higher than $T(1)$, and it is.

- Express the probability that the instantaneous rate a download receives is less than 1Mbit/s.

The instantaneous rate is less than 1Mbit/s if there are more than 10 downloads active.

$$P(n > 10) = 1 - \sum_{i=0}^{10} p_i = 1 - \sum_{i=0}^{10} \rho^i (1-\rho) = \rho^{11}.$$

Problem 3

a) Compare WFQ and WF2Q considering the complexity of the scheduling decisions and the achieved performance (select performance parameters). In what scenarios is it beneficial to use WF2Q?

WF2Q compares not only the finishing times in GPS, but also the starting times, and do not schedule a packet to transmit if its transmission has not started in GPS.

The delay, delay jitter and buffer requirement bounds are the same for the two scheduling solutions. However, these are worst case bounds. WF2Q typically generates less bursty flows, specifically if there are many connections, which should mean lower end-to-end delay jitter for typical cases, as well as less loss, if buffers are not large. A disadvantage of WF2Q is its slightly increased complexity.

b) Compare Jitter-Earliest-Due-Date and Stop-and-Go, considering the complexity of the scheduling decisions. Explain the achieved delay and delay jitter limits.

Jitter-EDD maintains a transmission time deadline. To be more flexible that a rigid non-work-conserving scheme, it allows transmission before this deadline, but then the packet needs to wait at the "entrance" at the next node until the deadline, and can be scheduled only after that.

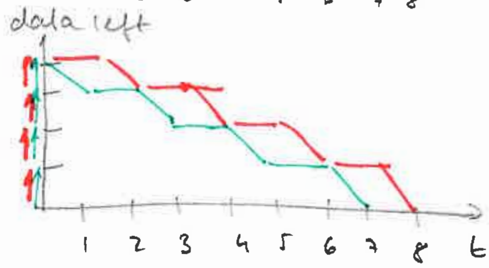
Consequently (assuming that the network is not overloaded....), the maximum end-to-end delay is nD, where n is the number of nodes on the path and D is the deadline. The maximum jitter is D, if the packet does not have to wait at the last node. Otherwise zero.

Stop-and-Go maintains a time frame structure of length T (not synchronized at the nodes!), and packets received in one frame are transmitted in the next frame only.

The maximum end-to-end delay depends on T, and has some additive part, that shows the possible additional shift a packet position can have within a frame, which is a maximum of T. All in all, the maximum delay is $2nT$. The delay jitter is limited by the frame size, that is, T.

Scheduling, Problem 2.

a)



GPS: packets of both flows are served with $r = 0.5$.

Finishing times: $F(1,1) = F(2,1) = 2$

$F(1,2) = F(2,2) = 4$

⋮

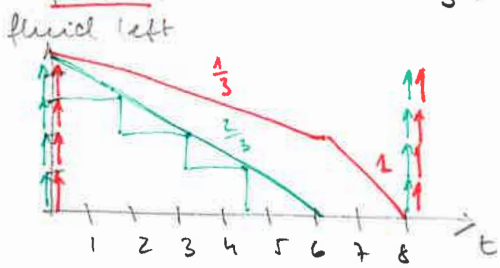
PGPS: as finishing times are equal, select packet to transmit randomly. One possible solution is this:

Finishing times:

Flow 1: 1, 3, 5, 7

Flow 2: 2, 4, 6, 8

b) Weights 2:1 $\Rightarrow$ In GPS flow 1 is served with rate $\frac{2}{3}$, flow 2 with rate $\frac{1}{3}$.



Finishing times:

Flow 1: $1\frac{1}{2}$, 3, $4\frac{1}{2}$, 6

Flow 2: 3, 6, 7, 8

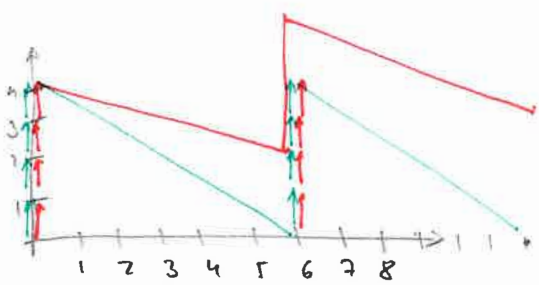
PGPS (one possible)

Flow 1: 1, 3, 4, 5

Flow 2: 2, 6, 7, 8

Flow 1 packets are served with some priority.

c)



GPS: Flow 1 receives rate $\frac{2}{3}$, flow 2 rate $\frac{1}{3}$. The packet generation rate is $\frac{4}{6} = \frac{2}{3}$ for both flows. That is, flow 2 will be unstable!

Finishing times: Flow 1: $1\frac{1}{2}$, 3, $4\frac{1}{2}$, 6, $7\frac{1}{2}$, 9.

Flow 2: 3, 6, 9, 12 ...

PGPS Finishing times (one possible)

Flow 1: 1, 2, 4, 5, 7, 8 ...

Flow 2: 3, 6, 9 ...