Take Home Midterm (30 Marks) **Oct 8th, 2016**

Dept. of Electrical Engineering, IIT Madras
EE4371 - Data Structures and Algorithms

- ▷ **Please write clear answers**
- ▷ **Code should be well commented and self-explanatory.**

A mix of video on demand and HTTP packets arrive at a router that has to shedule them to a home client through a low bandwidth link (512 kbps or 64 kbytes per second). The HTTP packets are somewhat sensitive to delay (not more than 15 seconds delay permitted) while the video on demand is very delay sensitive (max of 1 second delay is acceptable) and arrive at a rate of 32 kbytes per second. The HTTP packets are 80 bytes in size, while the video packets are 400 bytes long. You are to keep dropped video packet rates to below 10% every second (your algorithm should take this into account).

Time (sec)	**Event**
0	512**kBytes HTTP**
0^+	32**kBytes Video**
1	32**kBytes Video**
2	32**kBytes Video**
3	32**kBytes Video**
4	32**kBytes Video**
5	32**kBytes Video**
6	32**kBytes Video**
7	32**kBytes Video**
8	32**kBytes Video**
9	32**kBytes Video**
10	32**kBytes Video**
11	32**kBytes Video**
12	32**kBytes Video**
13	32**kBytes Video**
14	32**kBytes Video**
15	32**kBytes Video**

For the problem on hand, a 512 kbyte burst of HTTP packets has just arrived and has to be scheduled. The queue size is very large and it does not overflow. However, packets which have had too much delay are to be dropped immediately (show code for this). Note that the HTTP packets arrived in a lump - from a very fast connection. The video packets arrive every 12.5 msec, i.e., 80 packets per second.

Note: Packet delay means the time taken between packet arrival and its departure from the queue.

Note: In a nutsheel, the problem is

- ▷ **If you send out in arrival order, HTTP packets will get sent first and you will have to drop 8 seconds worth of video packets.**

▷ **If you send out 32kBytes of HTTP and video every second, after 15 seconds, there are still HTTP packets left, which are now over 15 seconds in the queue and must be dropped.**

▷ **You can drop 10% of video packets, but only when necessary. How do you decide this (for Q3)?**

1. **Write code to implement a FIFO queue into which all the packets are inserted as they arrive. How do you keep track of packet delay? Print out the current queue length and the % of dropped packets of each type (HTTP or video) will there be over the next 16 seconds? What is the problem with this solution? .... [10]**

   Hint: No decisions are required here. A simple circular queue is sufficient. The HTTP packets are choking your bandwidth and the video packets don't get sent on time. Work out the details and code the problem.

2. **Suppose at entry, video packets are always inserted at "front" of the queue, while the HTTP packets are inserted normally. The order of packets in the queue should be preserved for video packets (i.e., first one received should be the first one dequeued. Which type of queue is best? What is the value function that ensures the desired packet order? Code this design using a single queue. Print out the current queue length and the % of dropped packets of each type (HTTP or video) for the next 16 seconds and discuss if the algorithm worked. .... [10]**

   Hint: The main challenge is to come up with a value to assign to each packet so that a binary tree or max/min heap can keep the queue sorted with respect to that value, and *automatically* put the video packets in front of the HTTP packets. Your solution should not depend on the specific problem. So how will you decide what values to use for the two classes of packets?

3. Suppose at entry, video packets are put into a separate queue. Define extra Queue functions *Current_queue_length* and *Last_packet_age* for HTTP queue. .... [10]

   Current_queue_length: returns the number of bytes in the HTTP queue

   Last_packet_age: returns the age of the most recently queued HTTP packet

   When serving, the router takes a packet from one of the queues (based on your algorithm). How will the router decide whether to take a packet from the video queue or the HTTP queue? Give algorithm and code this case. Which type of queues will you use? How does your algorthm decide whether to keep or drop video packets? What % of dropped packets of each type will there be over the next 16 seconds? Enter your answer in a table of the following form:

Time	HTTP Q	Video Q	HTTP sent	Video sent	%HTTP Drops	%Video Drops
0	512	0	?	?	?	?

Hint: The queues are now just circular buffers and all the intelligence is in your dequeueing algorithm. Make use of the freedom to drop some video packets to ensure all the HTTP go through. But your algorithm must work in general. What algorithm will it be?