Gerard Mundo Bosch *scriber*

October 29 *date*

CS7260 – WFQ & GPS (continued)

WFQ Review

Select among all packets currently in queue, the one with the smallest GPS finish time[1] to serve next.

In both real and virtual time, the starting point[2] restarts after a period of no activity: it starts a busy period.

New arrivals will not change the virtual finish time of the current packets.

If you can figure out how the GPS clock evolves over time, the rest is relatively easy. That GPS finish time is the tricky part.

Let's define what an event is:

- The arrival of a new packet (either of a new flow or a previously inactive flow): at that moment you know its real arrival time, but we don't know its virtual time.
- A packet that finishes service (and makes or not a previously active flow inactive): at that moment you know the virtual finish time but you have to derive the real finish time.

Tracking GPS Clock

The virtual finish time can be defined as:

$$\boldsymbol{v}\left(F_K^{(i)}\right) = max\left(v(F_{k-1}^{(i)}), v(A_k^{(i)})\right) + \frac{L_k^{(i)}}{w^{(i)}}$$

Where v(S) is the virtual start time, L is length, W is weight and A is the arrival of a new packet.

The virtual start time can be defined as:

$$\boldsymbol{v}\left(S_K^{(i)}\right) = max\left(v(F_{k-1}^{(i)}), v(A_k^{(i)})\right)$$

[1] GPS finish time: finish time under hypothetical GPS schedule

[2] Starting point, also called "time zero"

Therefore, the virtual finish time can be redefined in terms of the virtual start time as:

$$\boldsymbol{v}\left(F_k^{(i)}\right) = \boldsymbol{v}\left(S_k^{(i)}\right) + \frac{L_k^{(i)}}{w^{(i)}}$$

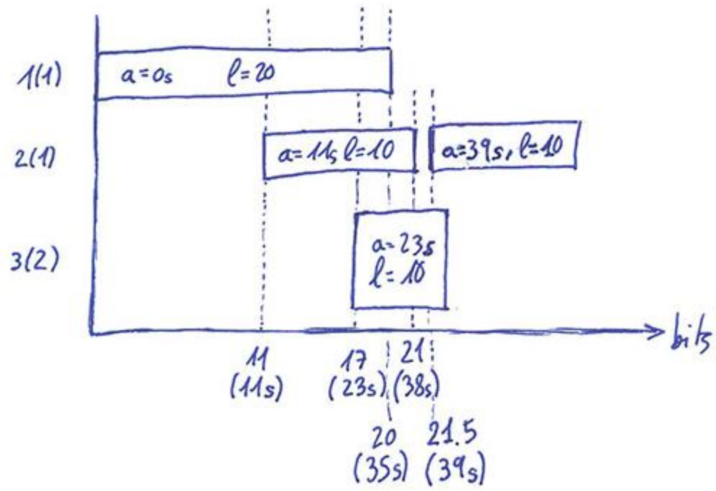

Figure 1: example of WFQ

Assuming the service rate is 1 bit per second, let's explain what is there in Figure 1:

Note: when referring to real time, time will be specified in seconds.

1. No arrival happens until time 11sec, and 11 bits of 1st packet have been already served.
2. Two packets will be competing for the service, which will alternate for both until the 3rd packet arrives at time 23sec. Each packet will have received 6 bits, so the virtual time is 11+6=17.
3. Now 3 packets will compete. Next event will be the finish service of the 1st packet. We know that this moment is time 20, because this is the length of the packet. The real time is: 20-17 = 3, 3*(1+1+2) = 12s, 12s+23s = 35s
4. Packet 2 ends at time 21. Real time is 35s + 1s*1 + 1s*2 = 38s
5. Packet 4 arrives at time 39s. As weight is 2, in 1 bit you can perform half bit service, so virtual time is 21.5.

The actual mapping between t and v(t) is, for any $\tau \le t_j - t_{j-1}$:

$$v(0) \coloneqq 0$$

$$v\left(t_{j-1} + \tau\right) \coloneqq V\left(t_{j-1}\right) + \frac{\tau}{\sum_{i \in B_j(t)} w^{(i)}}$$

Where $B_j(t)$ the set of active flows during $\left[t_{j-1}, t_j\right]$.

So, it is a piecewise linear function (in between two neighbor events, there's no other event). Think it like:

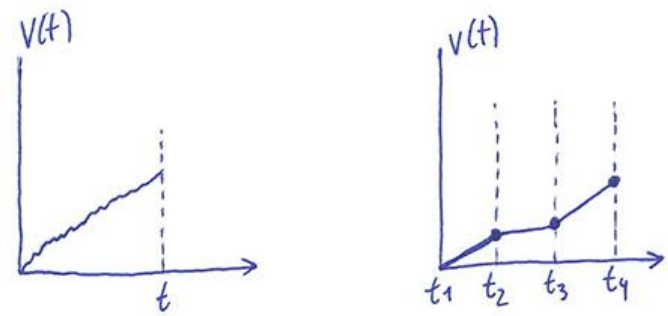

Figure 2: v(t) can be seen as a piecewise linear function

For example, taking the time of arrival for the packet 1 as t_0, and the arrival of packet 1 as t_1,

$$v(t_0 + \tau) \coloneqq V(t_0) + \frac{\tau}{w^i}$$

Because $t_0 = 0$, $t_1 = 11$ and $\tau = 11 - t_0$,

$$v(0 + 11) \coloneqq V(0) + \frac{11}{1} = 11$$

At time 11, a decision is made that the mapping function during the next period will be this:

$$v(t_1 + \tau) \coloneqq V(t_1) + \frac{\tau}{2}$$

As $t_2 = 23$,

$$v(23) \coloneqq V(11) + \frac{23 - 11}{2} = 11 + \frac{12}{2} = 17$$

Finally, the departure of packet 1 will happen at time 29. So the system sets an alarm clock for 15 seconds later and goes to sleep. However, there's an abrupt awakening because a new packet arrives. Then, $\tau = 23 - 11 = 12$.

$$v(t_2 + \tau) \coloneqq V(t_2) + \frac{\tau}{4} = 17 + \frac{12}{4} = 20$$

This time the clock is scheduled at time 35s and there's no abrupt awakening this time.

What does the V function look like for the given example?

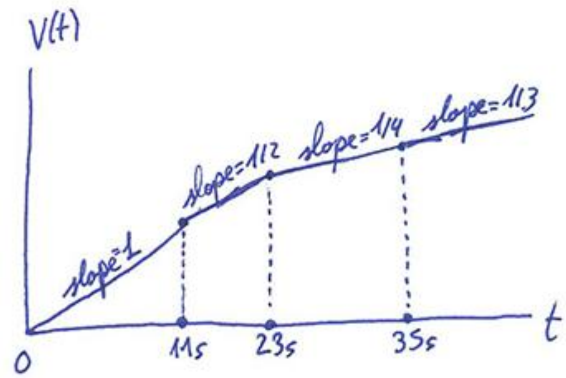

Figure 3: v(t) for the given example shown in Figure 1