Token Bucket Algorithm

Let

\[ P_i = \text{size of the bucket} \]

\[ r_i = \text{Rate at which the bucket is filled} \]

* The filling rate corresponds to how much the network allows you to transmit in a particular flow.
* The idea is if there are enough tokens in a bucket, packets are transmitted. Tokens are subtracted as packets are transmitted.
* If there are not enough tokens, the packet is held until a certain number 'L' of tokens are available.
* If the bucket is full, it means you cannot accumulate tokens beyond \( P(i) \).

This has implications -

* \( \sum r_i \leq \text{Total rate of the link, } r \). In other words, the sum of the rate of individual flows must be lesser than the total transmission rate of the link. This is required for deterministic guarantee.

**Theorem on End to End delay guarantee of WFQ** -

* Let a flow pass through a set of links -

\[ \text{Flow } i \rightarrow \text{Router 1} \rightarrow \text{Router 2} \rightarrow \ldots \rightarrow \text{Router } h \]

**Part 1** -

* For flow \( 'i' = (P(i),r(i)) \) through a single router in which WFQ is applied, what is the delay bound of the packets through that router for that flow?

The delay is \( P_i / r_i + \Theta_i \)

* \( P_i / r_i \) is the time taken by a burst of size \( P(i) \) to be cleared. \( P(i) \) is the maximum size of the burst.

Example -

* If the size of a bucket is 5 bits of tokens, then if each packet uses 2 bits of token, at time \( t=0 \), 2 packets will use a total of 4 bits of token. Then 1 bit of token is left. The third packet will not be transmitted until \( t=1 \).
\[ \Theta(i) = \frac{L_{\text{max}}}{r_i} + \frac{L_{\text{max}(i)}}{r_i} \]

L_{\text{max}(i)} = \text{Maximum size of packet inside Flow 'i'.}
L_{\text{max}} = \text{Maximum size of packet on that link.}

L_{\text{max}(i)}/r_i = \text{Time taken to transmit maximum sized packet in flow 'i'.}

**Part 2 - When the same flow flows through two links**

* The delay over the first link would be \( P_i/r_i + \Theta_{i,1} \).
* The delay over the first link would be \( P_i/r_i + \Theta_{i,2} \).
* Total delay over 2 links is \( P_i/r_i + \Theta_{i,1} + P_i/r_i + \Theta_{i,2} \).
* \( \Theta_{i,1} \) and \( \Theta_{i,2} \) could be different because there could be multiple flows on a single link. \( L_{\text{max}} \) could be different on different links. \( L_{\text{max}(i)} \) could be the same but \( r_i \) could be different across links.

* Since \( P_i/r_i \) appears twice in the end to end delay equation, the customer of such a network would be paying twice for the same service.

* If the customer does not want to pay twice, the end to end delay equation ought to be \( P_i/r_i + \Theta_{i,1} + \Theta_{i,2} \).

* Therefore, by extending the same formula to 'h' links the delay is

\[ P_i/r_i + \sum(\Theta_{i,j}) \text{ where } j \text{ ranges from 1 to 'h'}. \]

**Note:** The leaky bucket rate should always be lesser than the rate of flow 'i'.

* The above proof can be proved using **latency rate framework.**

**Latency Rate** -

* If 'T' is the time our traffic arrives and 't' is a time greater than 'T', then \( w(T,t) \) is the amount of work that can be done between T and t.
* $w(T,t) \geq \max\{0, r_i(t - \gamma - \delta_i)\}$

* $r_i, \delta_i$ is called the **latency rate server**.

Note: 'T' must be the start of a busy period of flow 'i'.