Congestion Control Algorithm

When too many packets are present in (a part of) the subnet, performance degrades. This situation is called congestion.

Factors Causing Congestion

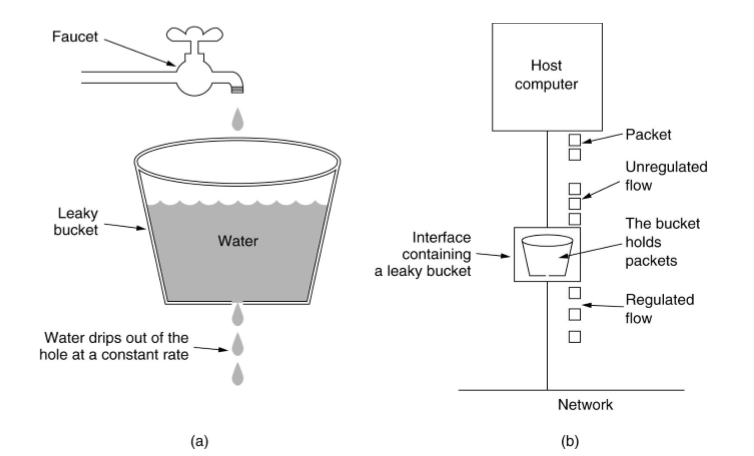
* Packet arrival rate exceeds the outgoing link capacity.
* Insufficient memory to store arriving packets
* Bursty traffic
* Slow processor

General Principles of Congestion Control

The two approaches of congestion Control are :

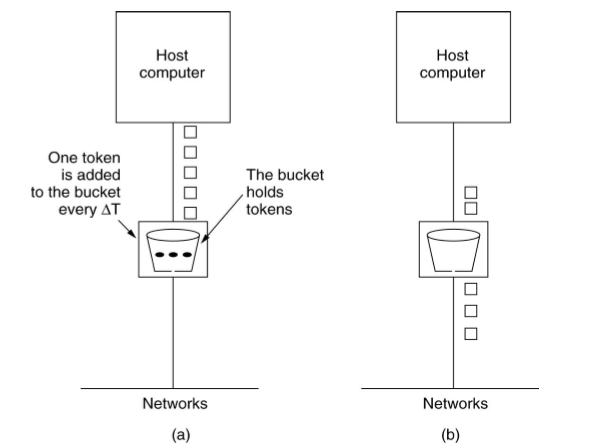
* Open loop: try to prevent congestion occurring by good design
* Closed-loop: monitor the system to detect congestion, pass this information to where action can be taken, and adjust system operation to correct the problem (detect,feedback and correct).

Leaky Bucket

Figure : Leaky Bucket Algorithm

* Imagine a bucket with a small hole in the bottom.
* No matter the rate at which water enters the bucket, the outflow is at a constant rate, when there is any water in the bucket and zero when the bucket is empty.
* Also, once the bucket is full, any additional water entering it spills over the sides and is lost.
* The same idea can be applied to packets, as shown in Fig. (b).
* Conceptually, each host is connected to the network by an interface containing a leaky bucket, that is, a finite internal queue.
* If a packet arrives at the queue when it is full, the packet is discarded. In other words, if one or more processes within the host try to send a packet when the maximum number is already queued, the new packet is unceremoniously discarded.

The Token Bucket Algorithm

Token Bucket Algorithm

* For many applications, it is better to allow the output to speed up somewhat when large bursts arrive, so a more flexible algorithm is needed, preferably one that never loses data.
* One such algorithm is the token bucket algorithm.
* Tokens arrive at the constant rate in the token bucket.
* If the bucket is full, tokens are discarded.
* A packet from the buffer can be taken out only if a token in the token bucket can be drawn.
* The token bucket algorithm provides a different kind of traffic shaping than that of the leaky bucket algorithm. The leaky bucket algorithm does not allow idle hosts to save up permission to send large bursts later.
* The token bucket algorithm does allow saving, up to the maximum size of the bucket, n. This property means that bursts of up to n packets can be sent at once, allowing some burstiness in the output stream and giving the faster response to sudden bursts of input.

Reference