Second Technique

IN QOS

Traffic shaping

By Theertha Praveen N

- Network traffic shaping is a process of managing data traffic on a computer network.
- It is used to manage the amount of data that can be sent or received over a network.
- Traffic shaping helps in reducing network congestion and improving the performance of the network by reducing latency, jitter, and packet loss.
- By controlling the rate at which data flows through a network, traffic shaping helps in ensuring that all users get equal access to resources and bandwidth.

Two techniques can shape or police the traffic

- Leaky Bucket
- Token Bucket

Leaky Bucket

Mathematical Logic of Leaky Bucket

- If a bucket has a small hole at the bottom, the water leaks from the bucket at a constant rate as long as there is water in the bucket.
- The rate at which the water leaks does not depend on the rate at which the water is input unless the bucket is empty
- If the bucket is full, the water overflows. The input rate can vary, but the output rate remains constant.

Leaky Bucket

Similarly, in networking, a technique called leaky bucket can smooth out bursty traffic. Bursty chunks are stored in the bucket and sent out at an average rate.

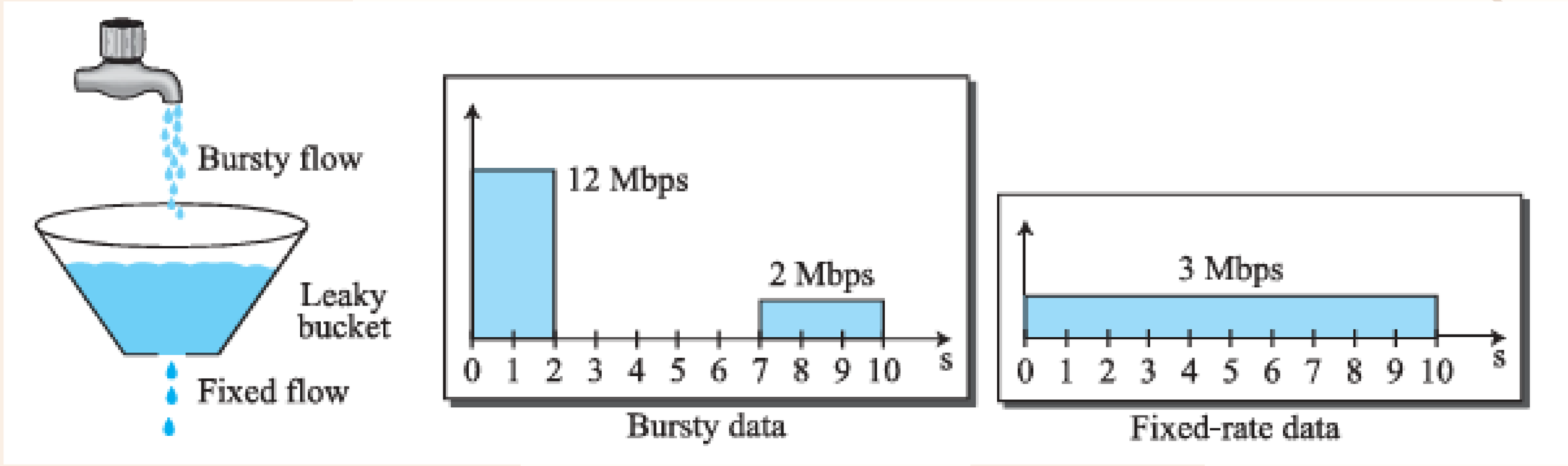
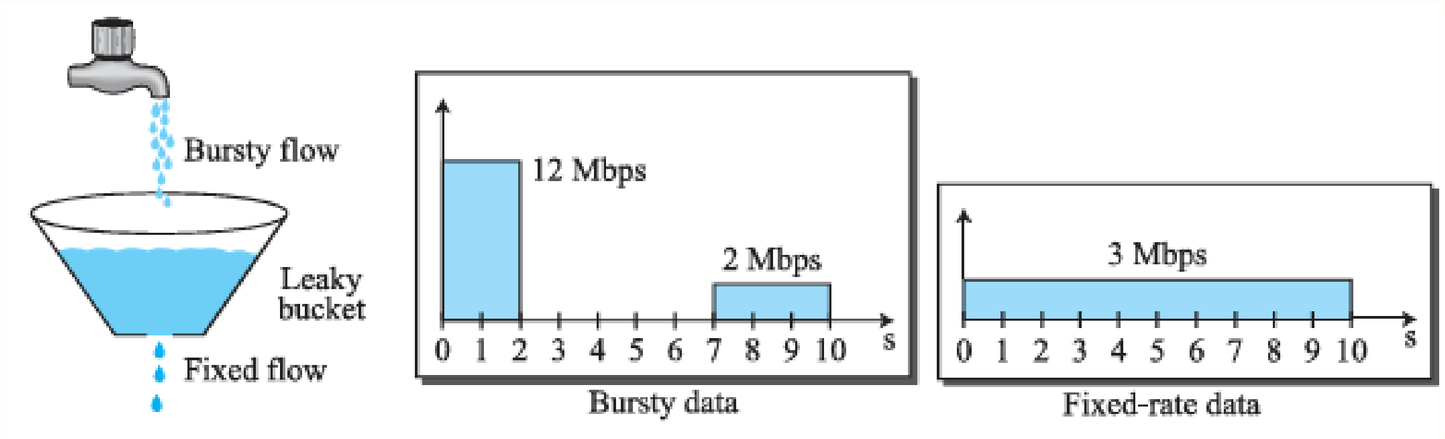
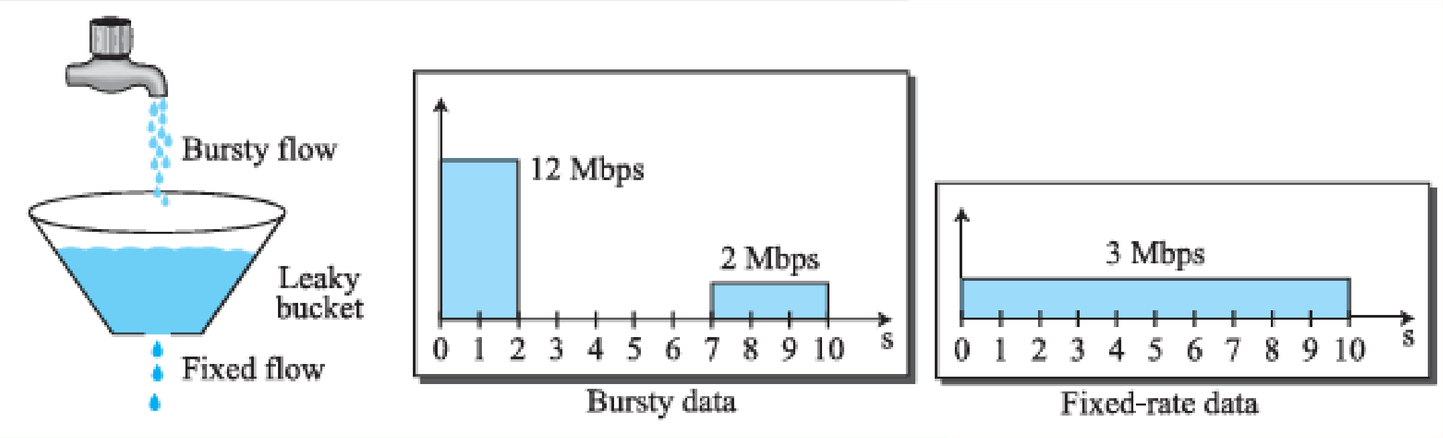


fig : Leaky bucket



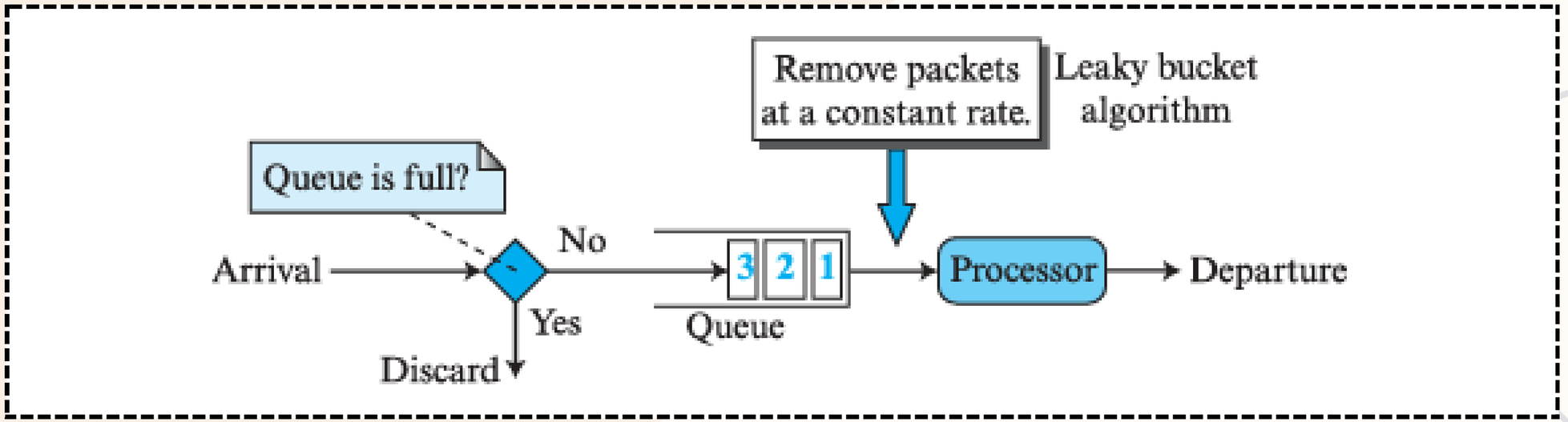
- In Figure the host sends a burst of data at a rate of 12 Mbps for 2 seconds, for a total of 24 Mb of data
- The host is silent for 5 seconds and then sends data at a rate of 2 Mbps for 3 seconds, for a total of 6 Mb of data.
- In the figure, we assume that the network has committed a bandwidth of 3 Mbps for a host. The use of the leaky bucket shapes the input traffic to make it conform to this commitment.



- In all, the host has sent 30 Mb of data in 10 seconds. The leaky bucket smooths the traffic by sending out data at a rate of 3 Mbps during the same 10 seconds

- Without the leaky bucket, the beginning burst may have hurt the network by consuming more bandwidth than is set aside for this host. We can also see that the leaky bucket may prevent congestion.

Leaky Bucket Implementation



A simple leaky bucket implementation is shown in Figure. A FIFO queue holds the packets. If the traffic consists of fixed-size packets (e.g., cells in ATM networks), the process removes a fixed number of packets from the queue at each tick of the clock. If the traffic consists of variable-length packets, the fixed output rate must be based on the number of bytes or bits.

The following is an algorithm for variable-length packets:

1. Initialize a counter to n at the tick of the clock.

2. If n is greater than the size of the packet, send the packet and decrement the counter by the packet size. Repeat this step until the counter value is smaller than the packet size.

3. Reset the counter to n and go to step 1.

Note :A leaky bucket algorithm shapes bursty traffic into fixed-rate traffic by averaging the data rate. It may drop the packets if the bucket is full.

Token Bucket

Token Bucket with Reference with Leaky Bucket

- The leaky bucket is very restrictive. It does not credit an idle host.

  For example

  - if a host is not sending for a while, its bucket becomes empty.
  - Now if the host has bursty data, the leaky bucket allows only an average rate.
  - The time when the host was idle is not taken into account.
  - On the other hand, the token bucket algorithm allows idle hosts to adumulate credit for the future in the form of tokens.

Token Bucket

It is designed to control the amount of data that a system can send or receive in some sort of period, ensuring that the traffic conforms to a specified rate.

- Allows for burst traffic by saving tokens
- Uses tokens to control data transmission
- Data waits until enough tokens are available
- More flexible with bursts
- Suitable for bursty traffic and rate limiting

Token Bucket

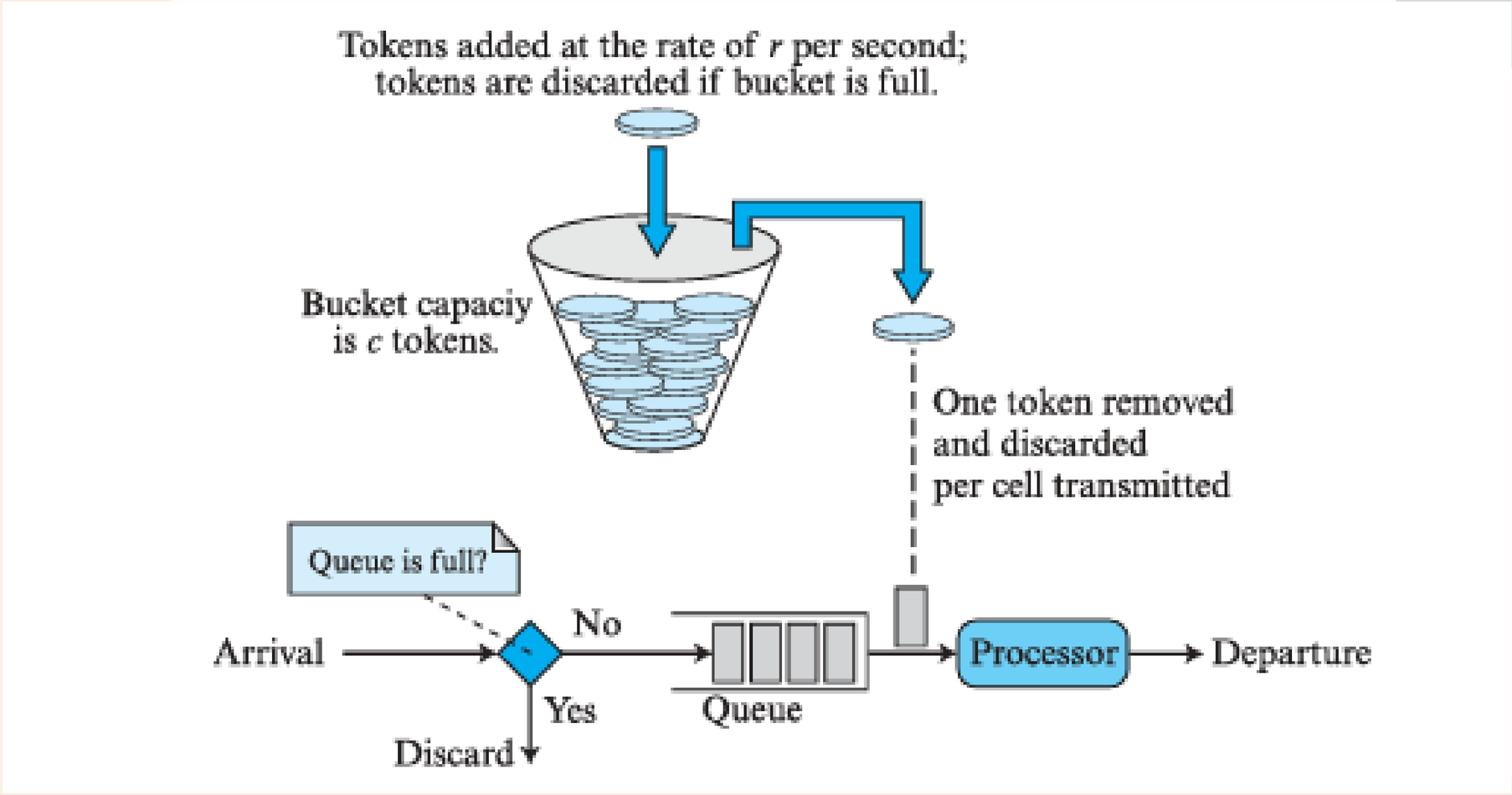


fig :Token Bucket

Assume the capacity of the bucket is c tokens and tokens enter the bucket at the rate of r token per second. The system removes one token for every cell of data sent. The max-imum number of cells that can enter the network during any time interval of length : is shown below.

$$\textbf{Maximum number of packets} = rt + c$$

The maximum average rate for the token bucket is shown below.

$$\textbf{Maximum average rate} = (rt + c)/t \textbf{ packets per second}$$

Combining Token Bucket and Leaky Bucket

The two techniques can be combined to credit an idle host and at the same time regulate the traffic. The leaky bucket is applied after the token bucket the rate of the leaky bucket needs to be higher than the rate of tokens dropped in the bucket.

Thank you!