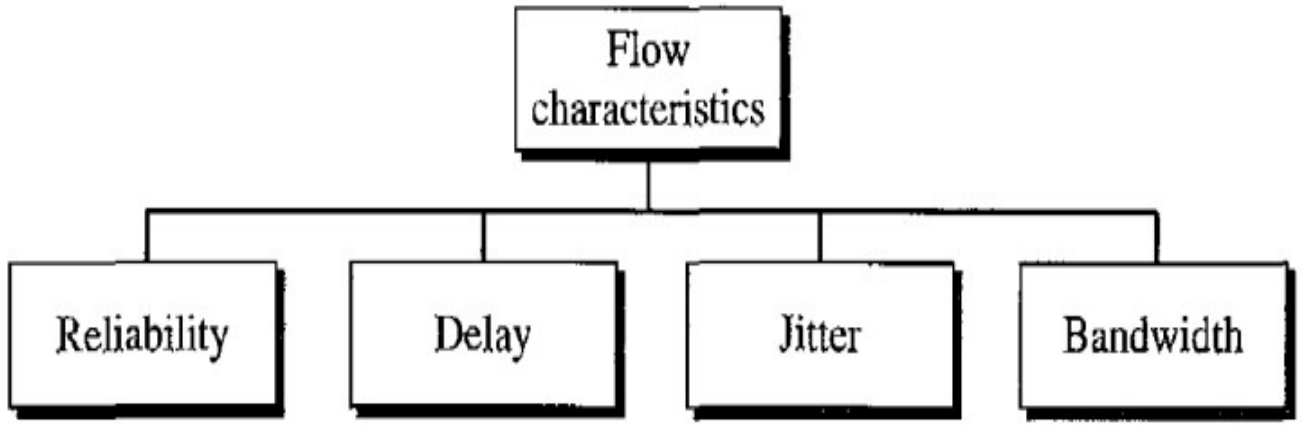
QUALITY OF SERVICE

Flow Characteristics

Figure 24.15 *Flow characteristics*



- Reliability

Reliability is a characteristic that a flow needs. Lack of reliability means losing a packet or acknowledgment, which entails retransmission.

- Delay

Source-to-destination delay is another flow characteristic. Again applications can tolerate delay in different degrees.

- Bandwidth

Different applications need different bandwidths. In video conferencing we need to send millions of bits per second to refresh a color screen while the total number of bits in an e-mail may not reach even a million

Jitter

- Jitter is the variation in delay for packets belonging to the same flow. For example, if four packets depart at times 0, 1, 2, 3 and arrive at 20, 21, 22, 23, all have the same delay, 20 units of time.
- On the other hand, if the above four packets arrive at 21, 23, 21, and 28, they will have different delays: 21,22, 19, and 24.
- For applications such as audio and video, the first case is completely acceptable; the second case is not.
- Jitter is defined as the variation in the packet delay.
- High jitter means the difference between delays is large; low jitter means the variation is small.

TECHNIQUES TO IMPROVE QoS

Techniques that can be used to improve the quality of service

- Scheduling- FIFO, Priority, Weighted Fair
- Traffic shaping- leaky bucket, token bucket
- Admission control,
- Resource reservation

Scheduling

- Packets from different flows arrive at a switch or router for processing. A good scheduling technique treats the different flows in a fair and appropriate manner.

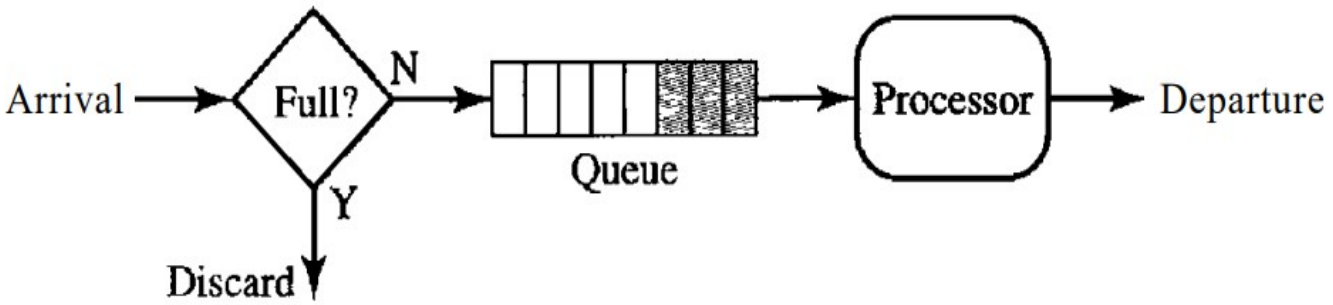
Techniques are

- FIFO Queuing,
- Priority Queuing,
- Weighted Fair Queuing.

FIFO Queuing

- In first-in, first-out (FIFO) queuing, packets wait in a buffer (queue) until the node (router or switch) is ready to process them.
- If the average arrival rate is higher than the average processing rate, the queue will fill up and new packets will be discarded
- Eg

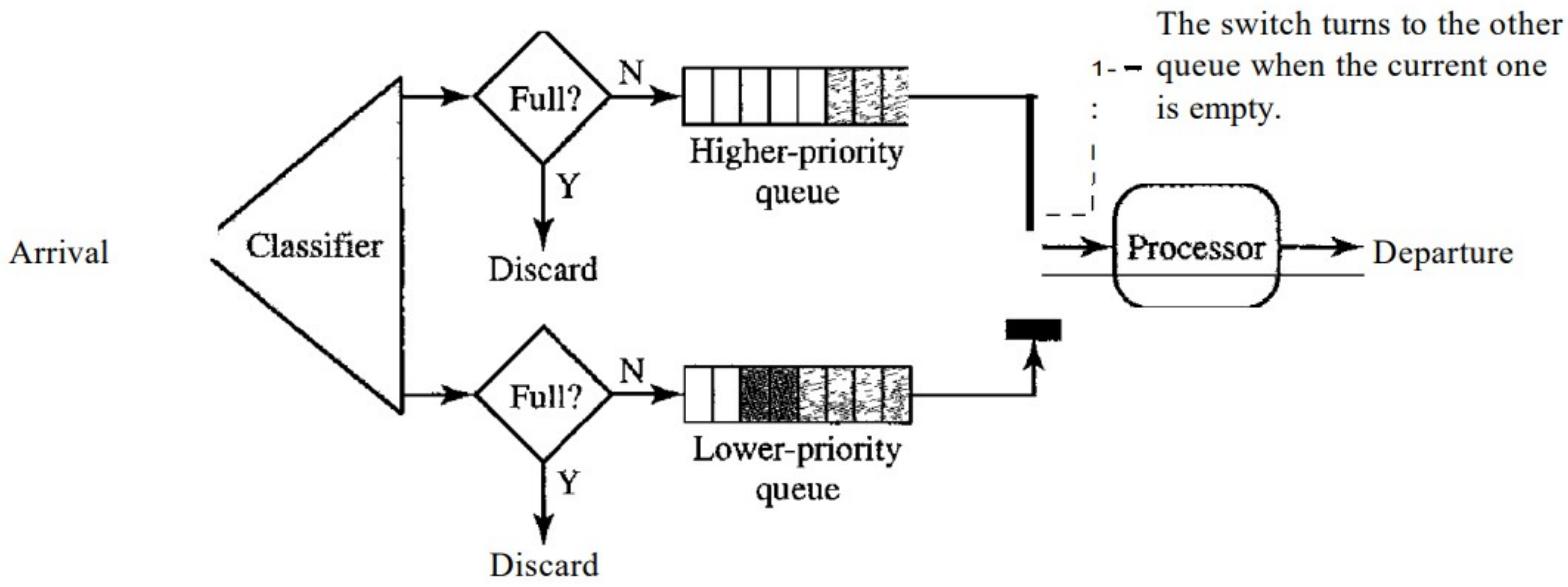
Figure 24.16 *FIFO queue*



Priority Queuing

- In priority queuing, packets are first assigned to a priority class. Each priority class has its own queue.
- The packets in the highest-priority queue are processed first. Packets in the lowest-priority queue are processed last.
- Note that the system does not stop serving a queue until it is empt

Figure 24.17 *Priority queuing*

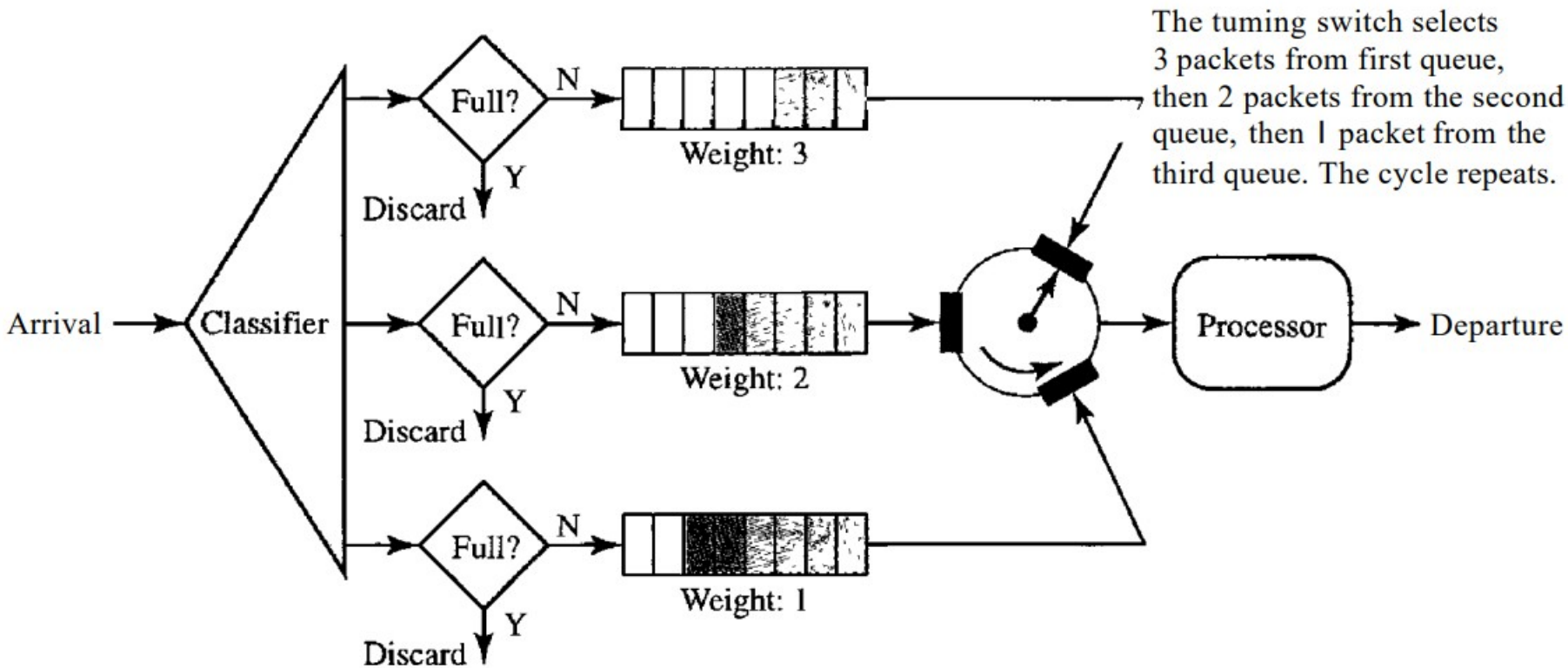


- A priority queue can provide better QoS than the FIFO queue because higher priority traffic, such as multimedia, can reach the destination with less delay.
- However, there is a potential drawback.
- If there is a continuous flow in a high-priority queue, the packets in the lower-priority queues will never have a chance to be processed. This is a condition called starvation.

Weighted Fair Queuing

- The queues, however, are weighted based on the priority of the queues; higher priority means a higher weight.
- The system processes packets in each queue in a round-robin fashion with the number of packets selected from each queue based on the corresponding weight.
- For example, if the weights are 3, 2, and 1, three packets are processed from the first queue, two from the second queue, and one from the third queue.

Figure 24.18 *Weighted fair queuing*



Traffic Shaping

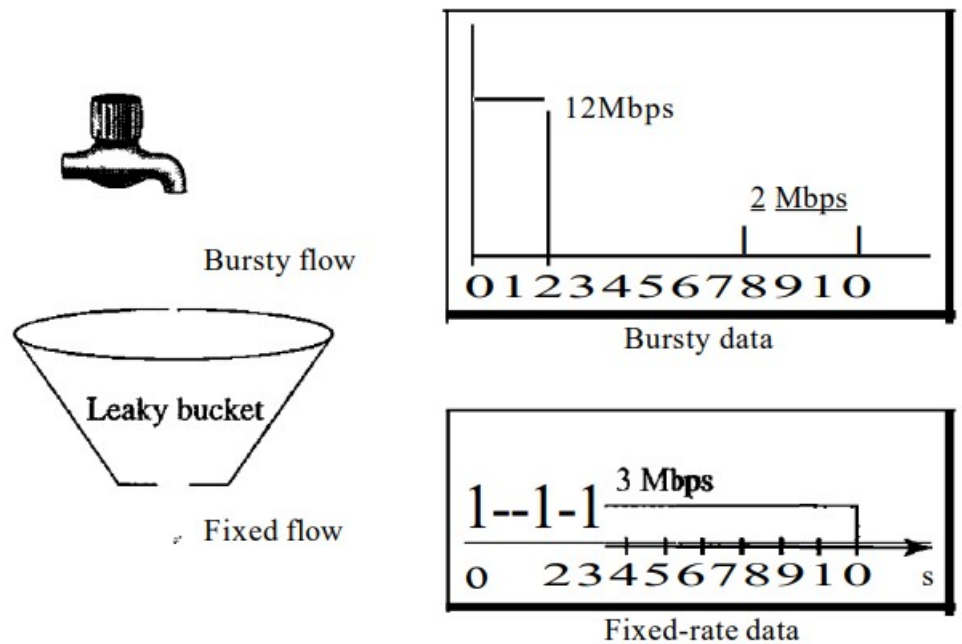
- Traffic shaping is a mechanism to control the amount and the rate of the traffic sent to the network.
- Two techniques can shape traffic:

leaky bucket and token bucket.

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 to the bucket unless the bucket is empty.
- The input rate can vary, but the output rate remains constant. 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

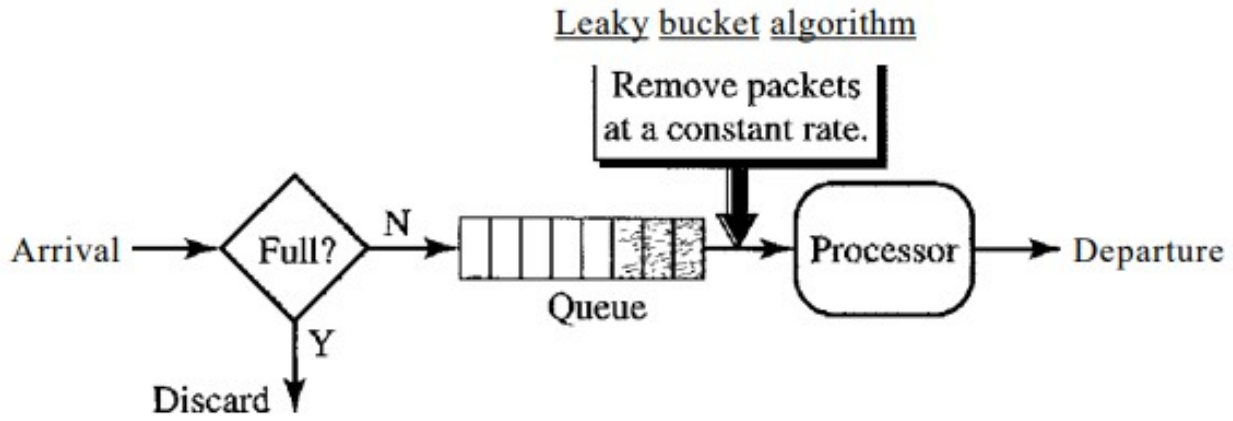
Figure 24.19 *Leaky bucket*



- The host sends a burst of data at a rate of 12 Mbps for 2 s, for a total of 24 Mbits of data.
- The host is silent for 5 s and then sends data at a rate of 2 Mbps for 3 s, for a total of 6 Mbits of data.
- In all, the host has sent 30 Mbits of data in lOs.
- The leaky bucket smooths the traffic by sending out data at a rate of 3 Mbps during the same 10 s.
- 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.

- 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 n is smaller than the packet size.
- 3. Reset the counter and go to step 1.

Figure 24.20 *Leaky bucket implementation*

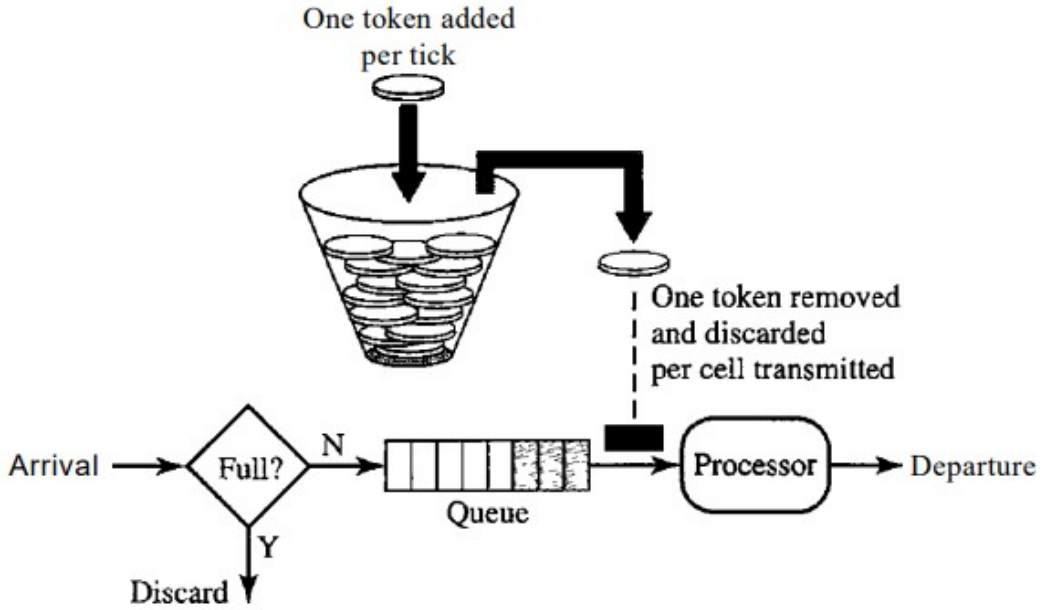


Token Bucket

- If a host is not sending for a while, its bucket becomes empty.
- The token bucket algorithm allows idle hosts to accumulate credit for the future in the form of tokens.
- For each tick of the clock, the system sends n tokens to the bucket.
- The system removes one token for every cell (or byte) of data sent. For example, if n is 100 and the host is idle for 100 ticks, the bucket collects 10,000 tokens.
- Now the host can consume all these tokens in one tick with 10,000 cells, or the host takes 1000 ticks with 10 cells per tick.

- The token bucket can easily be implemented with a counter.
- The token is initialized to zero. Each time a token is added, the counter is incremented by 1. Each time a unit of data is sent, the counter is decremented by 1.
- When the counter is zero, the host cannot send data

Figure 24.21 *Token bucket*



- Resource Reservation

A flow of data needs resources such as a buffer, bandwidth, CPU time, and so on. The quality of service is improved if these resources are reserved beforehand.

- Admission Control

Admission control refers to the mechanism used by a router, or a switch, to accept or reject a flow based on predefined parameters called flow specifications. Before a router accepts a flow for processing, it checks the flow specifications to see if its capacity (in terms of bandwidth, buffer size, CPU speed, etc.) and its previous commitments to other flows can handle the new flow