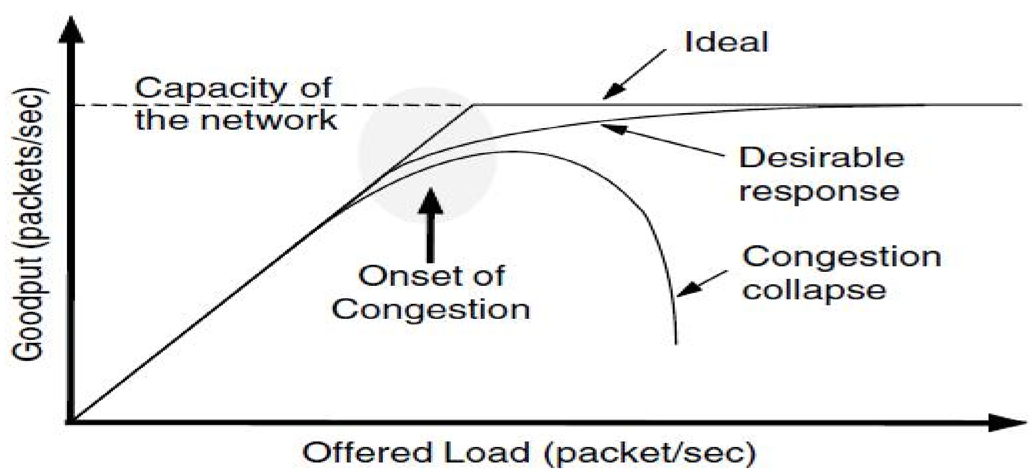
If a three-level hierarchy is chosen, with 8 clusters each containing 9 regions of 10 routers, each router needs 10 entries for local routers, 8 entries for routing to other regions within its own cluster, and 7 entries for distant clusters, for a total of 25 entries
Kamoun and Kleinrock (1979) discovered that the optimal number of levels for an *N router network is ln N, requiring a total of e ln N entries per router*

CONGESTION CONTROL ALGORITHMS

Too many packets present in (a part of) the network causes packet delay and loss that degrades performance. This situation is called **congestion.**
The network and transport layers share the responsibility for handling congestion. Since congestion occurs within the network, it is the network layer that directly experiences it and must ultimately determine what to do with the excess packets.
However, the most effective way to control congestion is to reduce the load that the transport layer is placing on the network. This requires the network and transport layers to work together. In this chapter we will look at the network aspects of congestion.



When too much traffic is offered, congestion sets in and performance degrades sharply

Above Figure depicts the onset of congestion. When the number of packets hosts send into the network is well within its carrying capacity, the number delivered is proportional to the number sent. If twice as many are sent, twice as many are delivered. However, as the offered load approaches the carrying capacity, bursts of traffic occasionally fill up the buffers inside routers and some packets are lost. These lost packets consume some of the capacity, so the number of delivered packets falls below the ideal curve. The network is now congested. Unless the network is well designed, it may experience a **congestion collapse**

<u>difference between congestion control and flow control</u>.

Congestion control has to do with making sure the network is able to carry the offered traffic. It is a global issue, involving the behavior of all the hosts and routers.

Flow control, in contrast, relates to the traffic between a particular sender and a particular receiver. Its job is to make sure that a fast sender cannot continually transmit data faster than the receiver is able to absorb it.

To see the difference between these two concepts, consider a network made up of 100-Gbps fiber optic links on which a supercomputer is trying to force feed a large file to a personal computer that is capable of handling only 1 Gbps. Although there is no congestion (the network itself is not in trouble), flow control is needed to force the supercomputer to stop frequently to give the personal computer a chance to breathe.

At the other extreme, consider a network with 1-Mbps lines and 1000 large computers, half of which are trying to transfer files at 100 kbps to the other half. Here, the problem is not that of fast senders overpowering slow receivers, but that the total offered traffic exceeds what the network can handle.

The reason congestion control and flow control are often confused is that the best way to handle both problems is to get the host to slow down. Thus, a host can get a ‘‘slow down’’ message either because the receiver cannot handle the load or because the network cannot handle it.

Several techniques can be employed. These include:

1. Warning bit
2. Choke packets
3. Load shedding
4. Random early discard
5. Traffic shaping

The first 3 deal with congestion detection and recovery. The last 2 deal with congestion avoidance

Warning Bit

1. A special bit in the packet header is set by the router to warn the source when congestion is detected.
2. The bit is copied and piggy-backed on the ACK and sent to the sender.
3. The sender monitors the number of ACK packets it receives with the warning bit set and adjusts its transmission rate accordingly.

Choke Packets

1. A more direct way of telling the source to slow down.
2. A choke packet is a control packet generated at a congested node and transmitted to restrict traffic flow.
3. The source, on receiving the choke packet must reduce its transmission rate by a certain percentage.
4. An example of a choke packet is the ICMP Source Quench Packet.

   <u>Hop-by-Hop Choke Packets</u>

1. Over long distances or at high speeds choke packets are not very effective.
2. A more efficient method is to send to choke packets hop-by-hop.
3. This requires each hop to reduce its transmission even before the choke packet arrive at the source

Load Shedding

1. When buffers become full, routers simply discard packets.
2. Which packet is chosen to be the victim depends on the application and on the error strategy used in the data link layer.
3. For a file transfer, for, e.g. cannot discard older packets since this will cause a gap in the received data.
4. For real-time voice or video it is probably better to throw away old data and keep new packets.
5. Get the application to mark packets with discard priority.

Random Early Discard (RED)

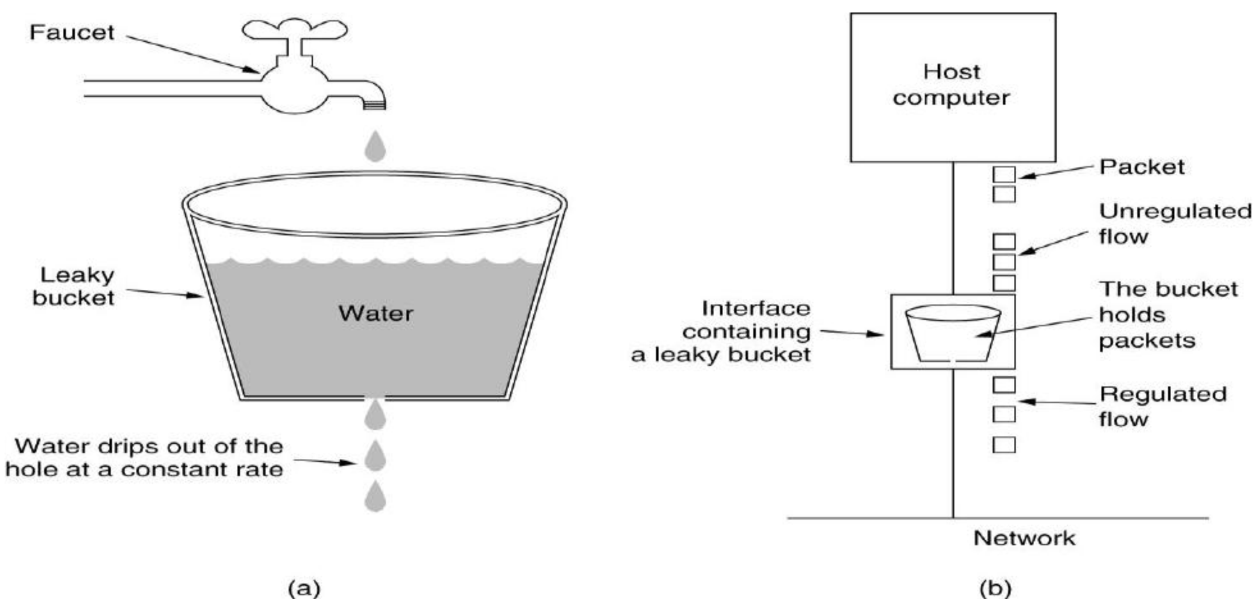
1. This is a proactive approach in which the router discards one or more packets *before* the buffer becomes completely full.
2. Each time a packet arrives, the RED algorithm computes the average queue length, ***avg***.
3. If *avg* is lower than some lower threshold, congestion is assumed to be minimal or non-existent and the packet is queued.
4. If *avg* is greater than some upper threshold, congestion is assumed to be serious and the packet is discarded.
5. If *avg* is between the two thresholds, this might indicate the onset of congestion. The probability of congestion is then calculated.

Traffic Shaping

1. Another method of congestion control is to "shape" the traffic before it enters the network.
2. Traffic shaping controls the *rate* at which packets are sent (not just how many). Used in ATM and Integrated Services networks.
3. At connection set-up time, the sender and carrier negotiate a traffic pattern (shape).

Two traffic shaping algorithms are:
Leaky Bucket
Token Bucket

The **Leaky Bucket Algorithm** used to control rate in a network. It is implemented as a single-server queue with constant service time. If the bucket (buffer) overflows then packets are discarded.

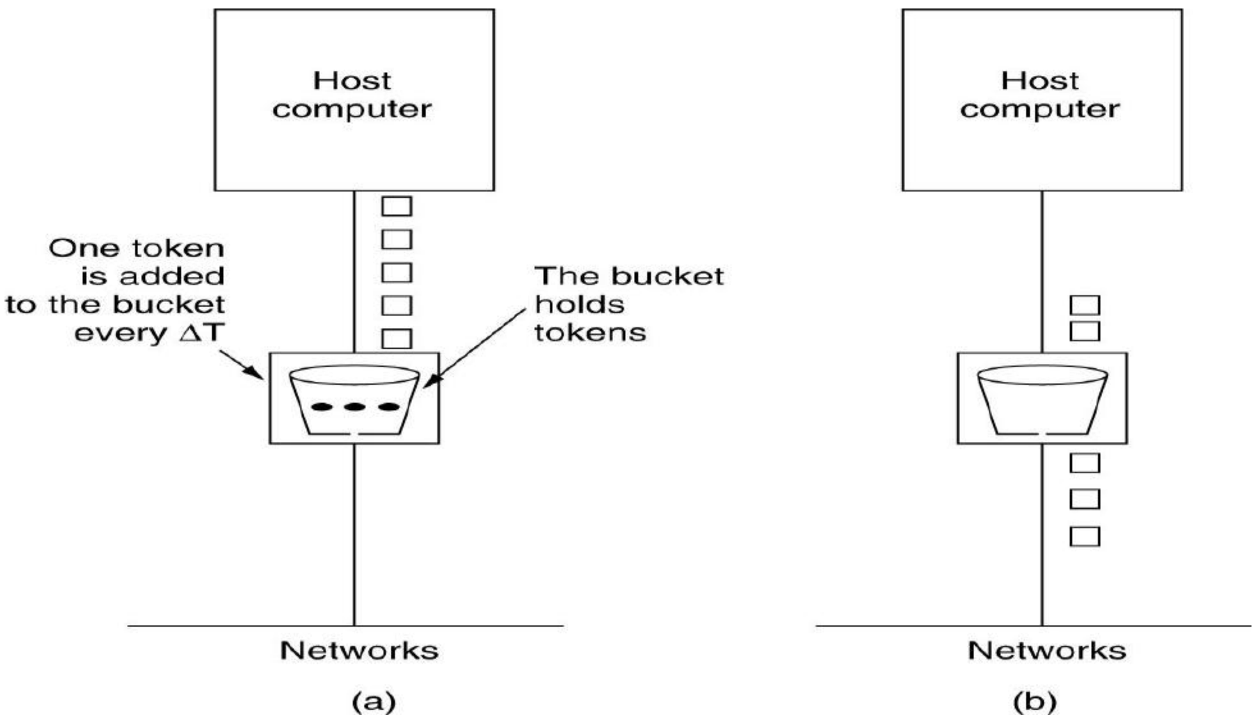


(a) A leaky bucket with water. (b) a leaky bucket with packets.

1. The leaky bucket enforces a constant output rate (average rate) regardless of the burstiness of the input. Does nothing when input is idle.
2. The host injects one packet per clock tick onto the network. This results in a uniform flow of packets, smoothing out bursts and reducing congestion.
3. When packets are the same size (as in ATM cells), the one packet per tick is okay. For variable length packets though, it is better to allow a fixed number of bytes per tick. E.g. 1024 bytes per tick will allow one 1024-byte packet or two 512-byte packets or four 256-byte packets on 1 tick

Token Bucket Algorithm

1. In contrast to the LB, the Token Bucket Algorithm, allows the output rate to vary, depending on the size of the burst.
2. In the TB algorithm, the bucket holds tokens. To transmit a packet, the host must capture and destroy one token.
3. Tokens are generated by a clock at the rate of one token every Δt sec.
4. Idle hosts can capture and save up tokens (up to the max. size of the bucket) in order to send larger bursts later.



(a) Before. (b) After.

Leaky Bucket vs. Token Bucket

1. LB discards packets; TB does not. TB discards tokens.
2. With TB, a packet can only be transmitted if there are enough tokens to cover its length in bytes.
3. LB sends packets at an average rate. TB allows for large bursts to be sent faster by speeding up the output.
4. TB allows saving up tokens (permissions) to send large bursts. LB does not allow saving.