CONGESTION CONTROL

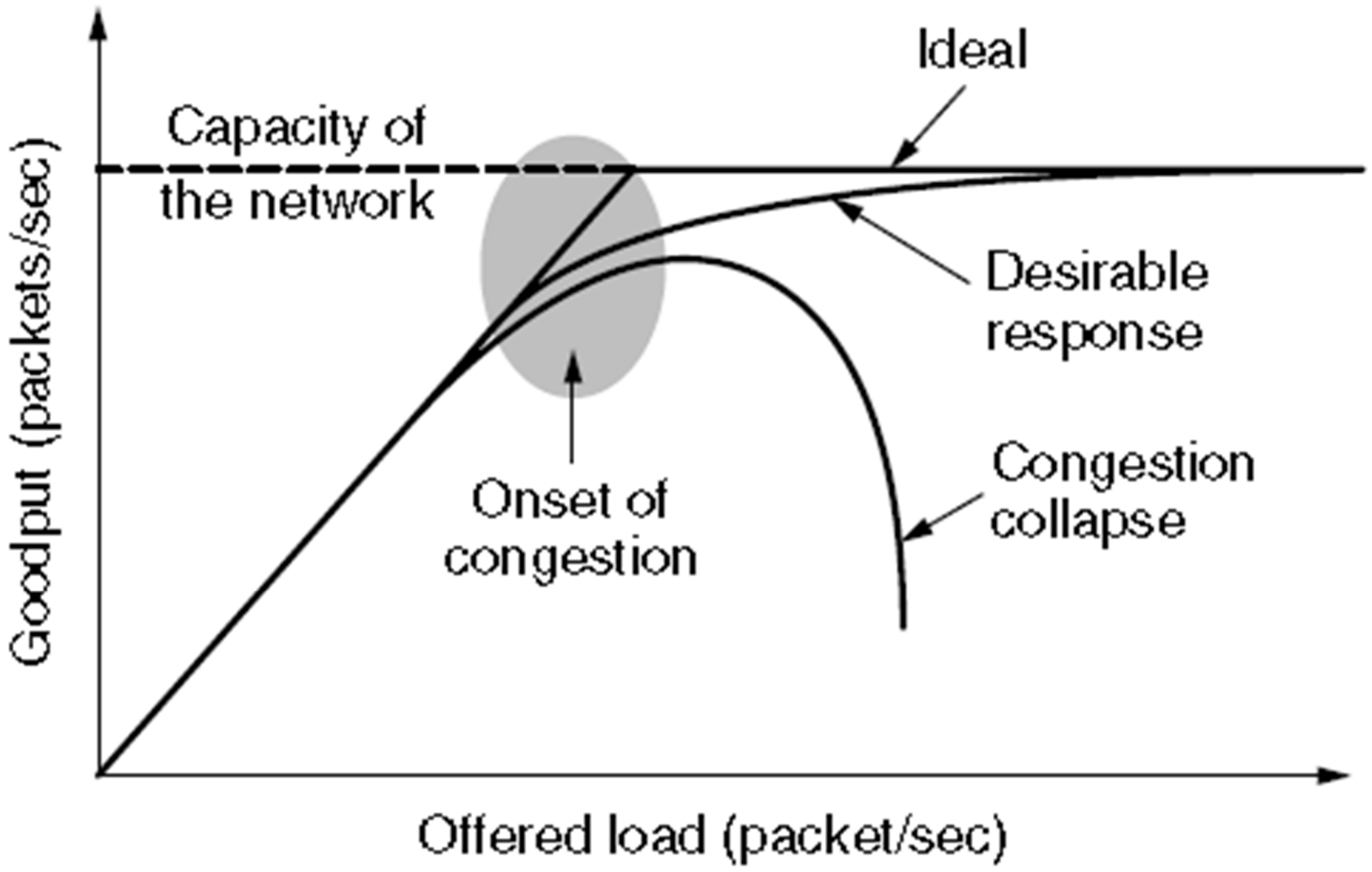
Too many packets present in (a part of) the network causes packet delay and loss that degrades performance. This situation is called **congestion**. The network and transport layers share the responsibility for handling congestion.

CONGESTION CONTROL

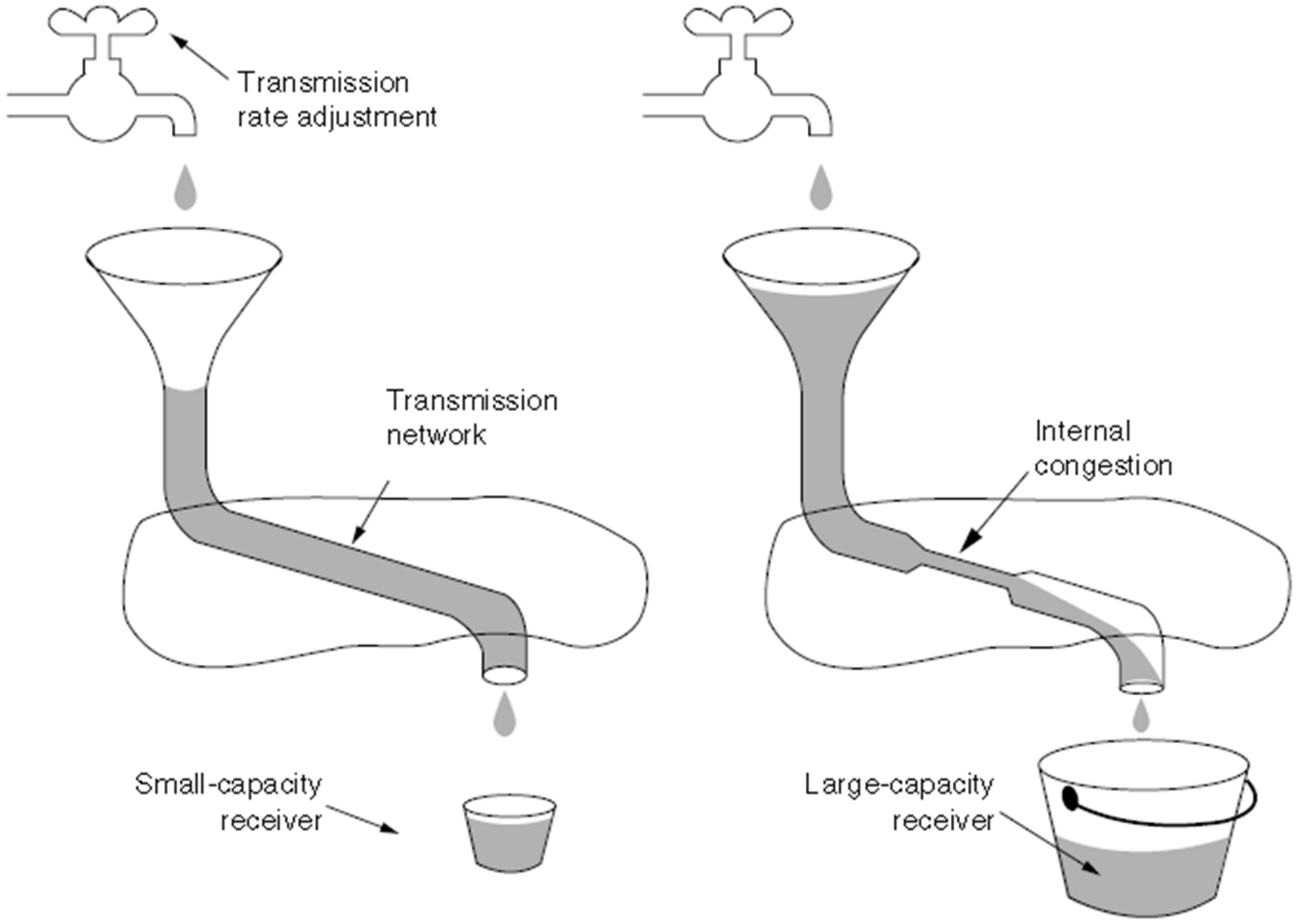
When the number of packets hosts send into the network is well within its carrying capacity, the number delivered is proportional to the number sent. However, as the offered load approaches the carrying capacity, bursts of traffic occasionally fill up the buffers inside routers and some packets are lost.

Unless the network is well designed, it may experience a **congestion collapse**, in which performance plummets as the offered load increases beyond the capacity.

CONGESTION CONTROL

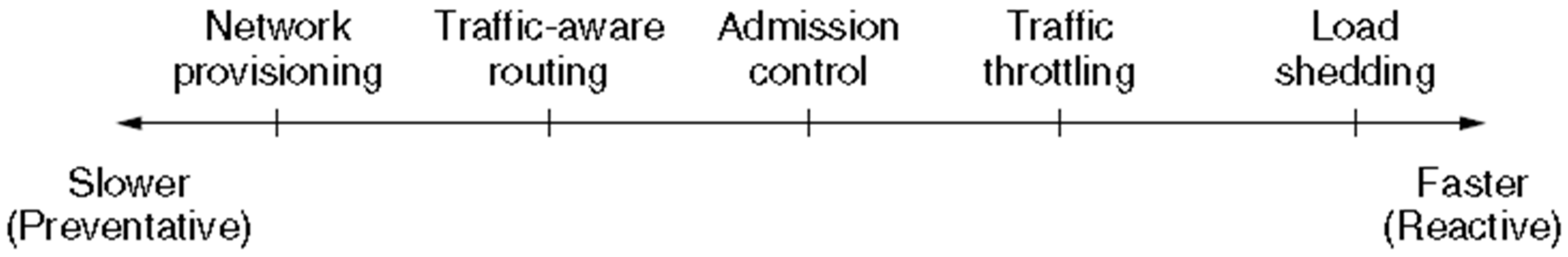


Difference between congestion control and flow control



Approaches to Congestion Control

We would like to design networks that avoid congestion where possible and do not suffer from congestion collapse if they do become congested. Unfortunately, congestion cannot wholly be avoided.



Approaches to Congestion Control

Network Provisioning: The most basic way to avoid congestion is to build a network that is well matched to the traffic that it carries. If there is a low-bandwidth link on the path along which most traffic is directed, congestion is likely. More often, links and routers that are regularly heavily utilized are upgraded at the earliest opportunity. This happens on a time scale of months, driven by long-term traffic trends.

Traffic-aware routing: To make the most of the existing network capacity, routes can be tailored to traffic patterns that change during the day as network users wake and sleep in different time zones.

Approaches to Congestion Control

Admission control: sometimes it is not possible to increase capacity. The only way then to beat back the congestion is to decrease the load. In a virtual-circuit network, new connections can be refused if they would cause the network to become congested.

Traffic throttling: when congestion is imminent the network can deliver feedback to the sources whose traffic flows are responsible for the problem. The network can request these sources to throttle their traffic, or it can slow down the traffic itself.

Load shedding : Finally, when all else fails, the network is forced to discard packets that it cannot deliver.

Admission control

The idea is simple: do not set up a new virtual circuit unless the network can carry the added traffic without becoming congested. Traffic is often described in terms of its rate and shape. The problem of how to describe it in a simple yet meaningful way is difficult because traffic is typically bursty. A commonly used descriptor that captures this effect is the **leaky bucket** or **token bucket**.

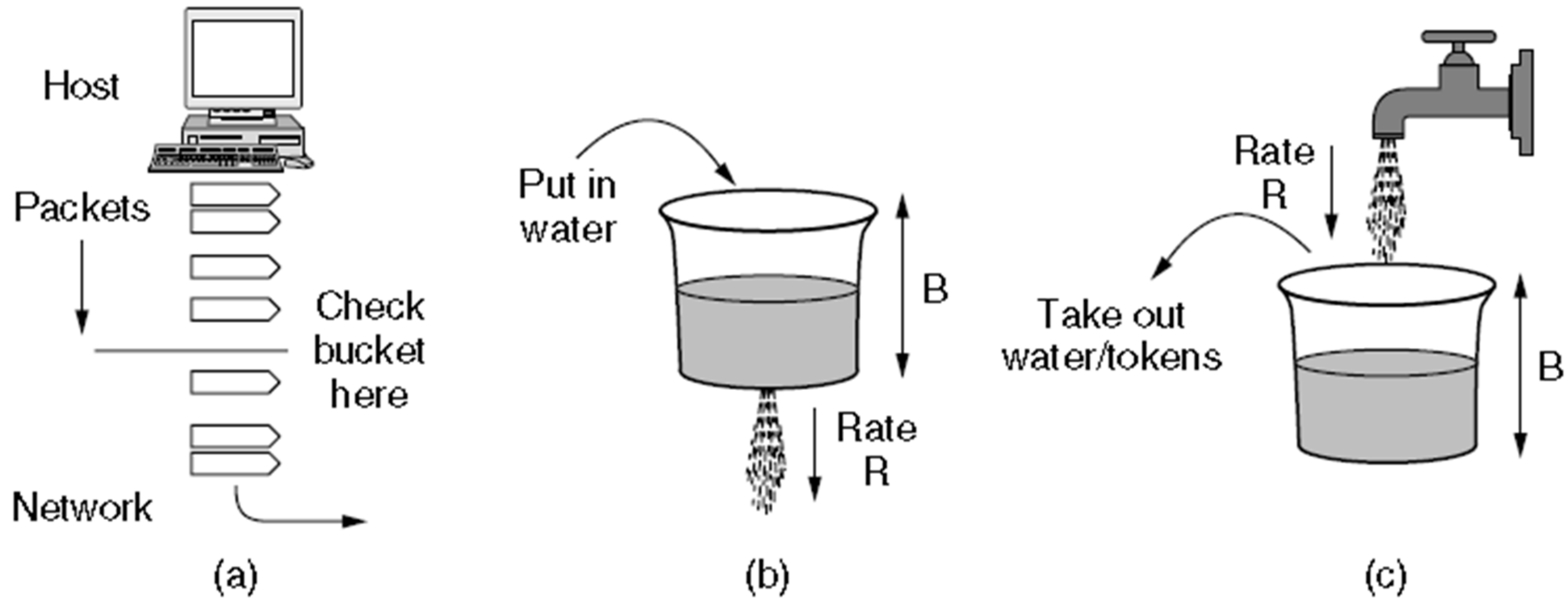


Figure 5-28. (a) Shaping packets. (b) A leaky bucket. (c) A token bucket.

Traffic Throttling

Each approach must solve two problems. First, routers must determine when congestion is approaching, ideally before it has arrived. To do so, each router can continuously monitor the resources it is using. Three possibilities are the utilization of the output links, the buffering of queued packets inside the router, and the number of packets that are lost due to insufficient buffering. Of these possibilities, the second one is the most useful.

The second problem is that routers must deliver timely feedback to the senders that are causing the congestion. Congestion is experienced in the network, but relieving congestion requires action on behalf of the senders that are using the network. To deliver feedback, the router must identify the appropriate senders. It must then warn them carefully, without sending many more packets into the already congested network. Different schemes use different feedback mechanisms, as we will now describe.

Traffic Throttling

The queuing delay inside routers directly captures any congestion experienced by packets. It should be low most of time, but will jump when there is a burst of traffic that generates a backlog. To maintain a good estimate of the queueing delay, *d*, a sample of the instantaneous queue length, *s*, can be made periodically and *d* updated according to:

$$d_{\text{new}} = \alpha d_{\text{old}} + (1 - \alpha)s$$

This is called an **EWMA** (**Exponentially Weighted Moving Average**). It smoothes out fluctuations and is equivalent to a low-pass filter. Whenever *d* moves above the threshold, the router notes the onset of congestion.

Traffic Throttling

The most direct way to notify a sender of congestion is to tell it directly. In this approach, the router selects a congested packet and sends a **choke packet** back to the source host, giving it the destination found in the packet. An example of a choke packet used in the early Internet is the SOURCE QUENCH message.

Instead of generating additional packets to warn of congestion, a router can tag any packet it forwards (by setting a bit in the packet's header) to signal that it is experiencing congestion. When the network delivers the packet, the destination can note that there is congestion and inform the sender when it sends a reply packet. The sender can then throttle its transmissions as before. This design is called **ECN (Explicit Congestion Notification)** and is used in the Internet.

Traffic Throttling

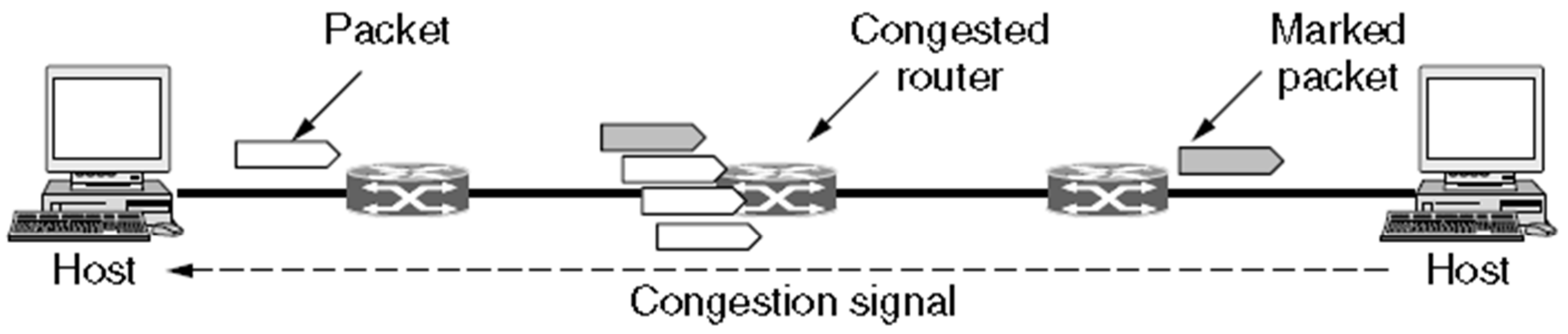


Figure 5-25. Explicit congestion notification

Traffic Throttling

Hop-by-Hop Backpressure

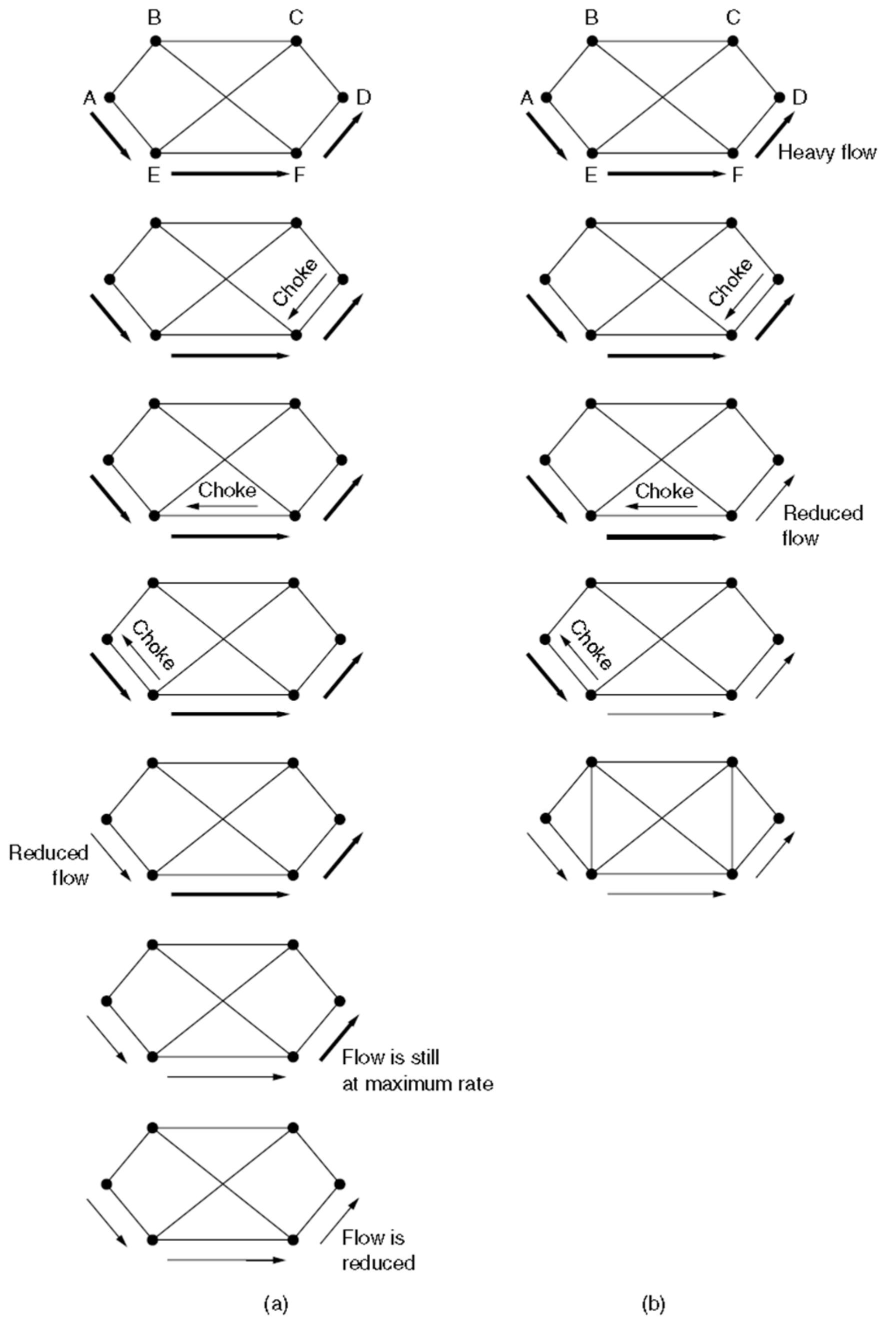


Figure 5-26. (a) A choke packet that affects only the source. (b) A choke packet that affects each hop it passes through.

Load shedding

When none of the above methods make the congestion disappear, routers can bring out the heavy artillery: load shedding. **Load shedding** is a fancy way of saying that when routers are being inundated by packets that they cannot handle, they just throw them away.

Load shedding

The key question for a router drowning in packets is **which packets to drop.** The preferred choice may depend on the type of applications that use the network.

- For a file transfer, an old packet is worth more than a new one.
- In contrast, for real-time media, a new packet is worth more than an old one. This is because packets become useless if they are delayed and miss the time at which they must be played out to the user.

The former policy (old is better than new) is often called **wine** and the latter (new is better than old) is often called **milk** because most people would rather drink new milk and old wine than the alternative.

Load shedding
Random Early Detection

By having routers drop packets early, before the situation has become hopeless, there is time for the source to take action before it is too late. A popular algorithm for doing this is called **RED** (**Random Early Detection**)