Part VI: Advanced Topics (Bonus Material on CD-ROM)

This part includes additional material that are related to Part IV and Part V; it consists of two sub-parts.

In the first sub-part, three chapters (Chapter 21, Chapter 22, and Chapter 23) cover functions and components of a router in further detail as a continuation of Part IV. First, different approaches to architect the switch fabric of a router are presented in Chapter 21. Second, packet queueing and scheduling approaches are discussed along with their strengths and limitations in Chapter 22. Third, traffic conditioning, an important function of a router, especially to meet service level agreements, is presented in Chapter 23.

In the second sub-part, we include two chapters (Chapter 24 and Chapter 25). Transport network routing is presented first in its general framework, followed by a formal treatment of the transport network route engineering problem over multiple time periods, in Chapter 24. The final chapter (Chapter 25) covers two different dimensions: optical network routing and multi-layer network routing. In optical network routing, we discuss both SONET and WDM in a transport network framework; more importantly, we also point out the circumstances under which a WDM on-demand network differs from a basic transport network paradigm. Furthermore, we discuss routing in multiple layers from the service network to multiple views of the transport networks; this is done by appropriately considering the unit of information on which routing decision is made and the time granularity of making such a decision. We conclude by presenting overlay network routing and its relation to multilayer routing.

23 Traffic Conditioning

We used to think that if we knew one, we knew two, because one and one are two. We are finding that we must learn a great deal more about "and."

Arthur Eddington

Reading Guideline

The traffic manager at a router is responsible for traffic conditioning. This chapter can be read independently; however, the background material presented in Chapter 14 on router architectures helps in understanding the overall role of the traffic manager.

While the original service model for IP is the best-effort service model, over the past decade the differentiated service ("diff-serv") model has become popular. The difficulty with the best effort service model is that it provides little guarantee of any kind on quality of service. The differentiated services model allows traffic classification to be applied to arriving packets for treatment through a traffic conditioner. Through this process, a form of soft quality of service can be assigned to different service classes. For example, such classification can be for certain customer's traffic that may be based on service level agreements. In other words, to meet service level agreements for a certain customer's traffic, a quality of service mechanism is needed. This is where the diff-serv model comes into the picture, and the need for traffic conditioning. In essence, service providers need mechanisms to discriminate traffic from various subscribers and limit their rate and volume of traffic entering into the network. In this chapter, we focus on such mechanisms, namely, *traffic shaping*, *traffic policing*, and *packet marking* toward ensuring service level agreements.

23.1 Service Level Agreements

We first start with a discussion of service level agreements. A service level agreement (SLA) is a formal contract that exists between a customer and a service provider that specifies the forwarding service the customer will receive from the service provider during the duration of the agreement. In addition, it details the penalties assessed when any of the clauses in the SLA are violated by the service provider. As more and more enterprises depend on outsourced services, they rely on SLAs to guarantee specific levels of functionality and availability.

While there are different types of SLAs governing different aspects of the outsourced business, we focus our attention on the network service SLA, referred to as a network SLA. A network SLA specifies its objectives in terms of network performance between one or more exchange points on its network. A network SLA typically covers the following:

- *Physical network characteristics:* This covers the type of network infrastructure service that the service provider is willing to provide. These are expressed as network availability (system uptime) and network capacity (throughput). While most enterprises desire 100% availability, it might not be necessary in many environments. For instance, in environments for e-commerce, 100% availability is critical. However, for traditional business environments, an average ranging from 99.5% to 99.9% might be acceptable. When specifying throughput, a network's capacity is detailed in the capacity of the backbone connections within the network's core, such as 10 Gbps.
- *Network connection characteristics:* This aspect of the SLA provides details about the bandwidth being provisioned, the acceptable rate of data losses, error rate, end-to-end latency, and jitter. While most of the service providers guarantee 99% packet delivery rates, this might not be sufficient for real-time applications such as voice over IP (VoIP), interactive video, and so on. For the predominant web browsing traffic, losses of up to 5% might be acceptable. Similar to data loss, latency and jitter are critical for VoIP and multimedia traffic; these applications require response times of 100 millisec or less. Many service providers in the United States and Europe often guarantee a round-trip delay of 85 millisec between the routers in their core networks.

We next present an example of a network SLA.

Example 23.1 *A network SLA.*

An enterprise customer uses an Internet service provider (ISP) to receive dedicated Internet access to connect to its remote sites. The customer and the ISP agree to the following network SLA, which defines the requirements of the service by the customer and the commitments by the service provider.

- *Network availability:* The network will be available to the customer free of network outages 99.95% of the time, which is the standard service-level guarantee. In the case of unplanned down time, the customer will be compensated 10% of the monthly bandwidth charge. The service provider may suspend the service occasionally for operational reasons and network upgrades without invalidating the service-level guarantee of 99.95%.
- *Network connection:* The amount of bandwidth available to the customer is 10 Mbps for download and 5 Mbps for upload. The average available bandwidth will not be less than the specified amount for more than 0.1% per month.
- *Latency and jitter:* The average monthly packet loss on the core network will not exceed 0.2%. The average monthly latency on the core network will be 50 millisec or less. The average jitter will be 250 μsec or less. The maximum jitter on the core network will not exceed 10 millisec more than 0.2% per month. ▲

23.2 Differentiated Services

From the above discussion, we can see that SLAs provide a way to specify agreements. Conceptually, however, how do we provide differentiated services as an operational framework so that SLAs can be met? A simple way to understand this framework is shown in Figure 23.1. Arriving packets are first processed through a packet classifier, which is then handled through the traffic conditioning phase. As you can see, the traffic conditioning phase has four main components: traffic meter, packet marker, traffic shaper, and traffic policer. Through these

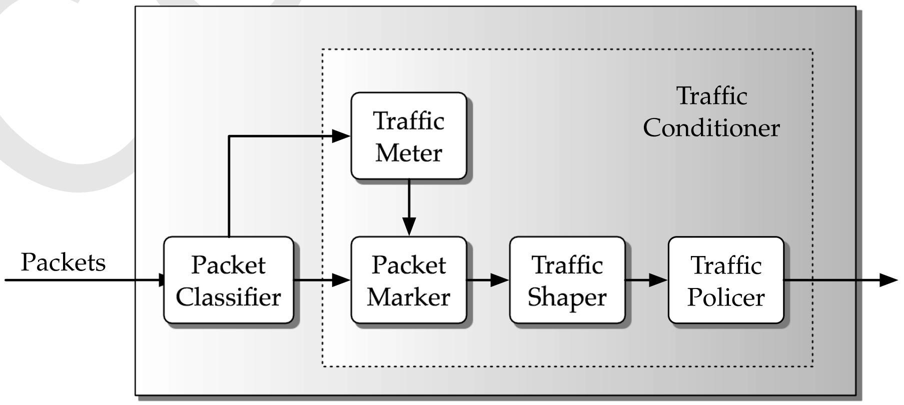

FIGURE 23.1 An operational framework for differentiated services.

components, the traffic conditioning phase attempts to ensure SLAs. In particular, it ensures that an arriving packet does not violate agreements.

How do you communicate that a packet is needed to be treated in a certain manner? The differentiated services architecture allows such packet marking using the type of service (TOS) field in the IPv4 header or the traffic class field in IPv6 header. Specifically, 6 bits are used for differentiated service code point (DSCP), which is then part of the TOS or traffic class field.

23.3 Traffic Conditioning Mechanisms

Now let us illustrate traffic condition mechanisms through a network example. Consider Figure 23.2 where the possible locations of traffic shaping, marking, and policing are shown; here, ISP2 serves as the backbone service provider for the tier 2 ISPs, ISP1 and ISP3. In this case, the ISP1 and ISP3 are subscribers to ISP2. For the sake of discussion, assume that as a part of the SLA between ISP1 and ISP3, ISP1 can upload a committed rate of 1 Gbps worth of traffic (upload bandwidth of 1 Gbps) and download a committed rate of 3 Gbps of traffic. Similarly, assume that the SLA between ISP2 and ISP3 specifies that ISP2 upload bandwidth is limited to 5 Gbps and download bandwidth to 10 Gbps.

When traffic from ISP1 is handed off to ISP2, it is the responsibility of ISP1 not to exceed the committed upload rate of 1 Gbps. The mechanism used to enforce that a stream of packets transmitted from a router conforms to a specific rate is called *traffic shaping*. In other words, when the traffic is handed off to another network, the traffic is shaped so that the SLAs are not violated.

Similarly, when traffic is received by ISP2 from ISP1, it needs to ensure that the traffic sent by ISP1 indeed does not exceed 1 Gbps. When the rate is exceeded, it is considered as a violation of SLA and ISP2 has two options for dealing with such traffic. One option is to drop the excess traffic; the other option is to mark this excess traffic as "out-of-profile" for the given SLA. The out-of-profile traffic can be handled according to a certain policy. For instance, one policy might be to drop such traffic and the other is to admit this traffic into the network as best-effort delivery. In other words, the out-of-profile traffic dropped when congestion is encountered in upstream routers; otherwise, the packet is delivered to its destination.

The mechanism of monitoring the incoming traffic rate and dropping packets when the agreed rate is exceeded is called *traffic policing*. Marking the packets as out-of-profile when they exceed the rate and processing them according to a policy is called *packet marking*. One

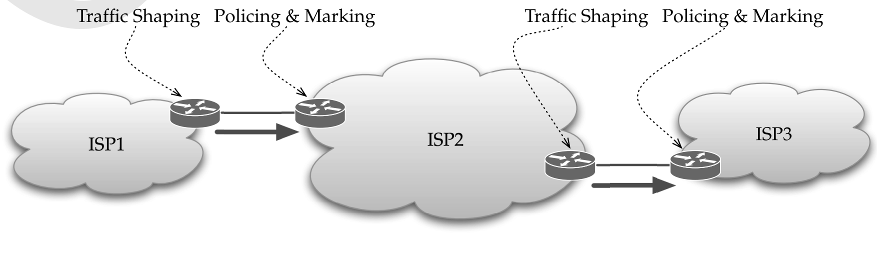

FIGURE 23.2 Traffic shaping, policing, and marking in networks.

can view packet marking as a gentler incarnation of traffic policing which provides the flexibility to handle excess traffic.

Various traffic conditioning mechanisms assume that the packets from one subscriber can be distinguished from another subscriber. For instance, if ISP1 and ISP3 are connected to ISP2 using a single router, there is a need to segregate (classify) the packets. Such a classification is possible in practice using several fields in the IP header along with efficient algorithms for processing the same, as outlined in Chapter 16. In other words, packet classification is possible, other than fully relying on diff-serv code points.

23.4 Traffic Shaping

Traffic shaping regulates the rate and volume of traffic of an individual flow or an aggregate of flows admitted into the network. Note that an aggregate flow could mean all the flows whose traffic is sent out on the same interface. While traffic shaping can be used in many situations, the primary use is to smooth the traffic to a certain rate and ensure that traffic adheres to the SLA. Additionally, traffic shaping can be used to mitigate congestion by smoothing the bursty traffic such that the transmitted rate does not exceed the access speed of the target interface. Now let us see a few concrete examples of use of traffic shaping.

Example 23.2 *The need for traffic shaping.*

Consider the network shown in Figure 23.3. The service provider network directly connects to two content providers, CP1 and CP2 and an enterprise customer E. In addition, two residential customers C1 and C2 are connected indirectly through a point of presence (POP).

- *Control access to bandwidth:* Consider the enterprise customer E interested in controlling the bandwidth of outbound traffic due to a tier-pricing agreement with the service provider. The service provider might have provided a high-bandwidth connection circuit such as DS3 to customer E; however, the pricing might be based on the average utilization of the circuit. Thus, the customer has the flexibility to move to a higher bandwidth quickly in the future, but in the short term, to control expenses, can use shaping at router R9 so that the traffic submitted to ISP does not exceed a subrate. The shaped traffic, from router R9 to R8, is indicated in the Figure 23.3 using bold arrows.
- *Limiting per-user traffic:* Assume that the residential customers C1 and C2 connected to the ISP have signed up for services that provides different download speeds. For the sake of discussion, let us say that customer C1 has signed up for a service that provides a download speed of 1 Mbps. Similarly, customer C2 has signed up for a service that allows a download speed of 512 Kbps. Traffic shaping allows the service provider to set per-user traffic limits by configuring policies at router R1 and ensures that the user gets what he/she pays for. When traffic is limited in this fashion, users can still access whatever they want, but the flows are smoothed out to the specified limit rather than attempting to use all or much of the total available network capacity. Again, the shaped traffic is indicated in Figure 23.3 using bold arrows flowing from router R1 to customers C1 and C2. ▲

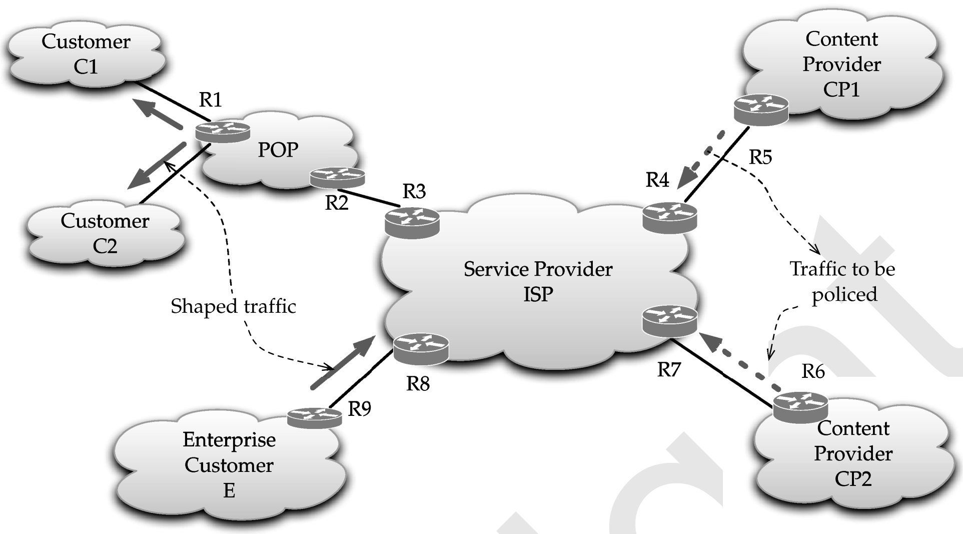

FIGURE 23.3 An ISP network.

Since shaping takes place at an egress interface in the router, it may be necessary to classify a packet to a flow earlier, possibly at the ingress interface. There are two predominant methods for traffic shaping:

- *Traffic smoothing:* This method eliminates bursts and presents a steady stream of traffic to the network, which can be implemented using a leaky bucket algorithm.
- *Traffic burst shaping:* This method shapes burst of predetermined size by averaging over a time window, which can be implemented using a token bucket algorithm.

Both schemes have different behavior and rate-limiting capabilities resulting in output streams with different characteristics. Most often in the literature, the leaky bucket and token bucket algorithms are discussed under the same name.

23.4.1 Leaky Bucket

A leaky bucket algorithm is primarily used to control the rate at which traffic enters the network. It provides a mechanism for smoothing bursty input traffic in a flow to present a steady stream into the network. In other words, the leaky bucket enforces a constant transmit rate regardless of the erratic burstiness in the input traffic of a flow.

The leaky bucket algorithm can be conceptually explained as follows. Imagine having a bucket per flow that has a hole at the bottom. An unregulated stream of packets arriving in the flow is placed in this bucket. The packets are drained slowly through the hole at the bottom and transmitted into the network at a constant rate of r bytes per sec. The bucket size (depth) is limited, say b bytes. When the rate of unregulated packets entering the bucket is

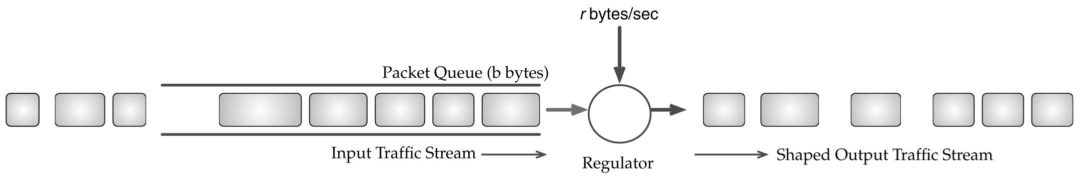

FIGURE 23.4 Leaky bucket algorithm.

greater than the drain rate of r, the bucket could be filled up. If a new packet arrives when the bucket is full, the entire packet is discarded. The leaky bucket is illustrated in Figure 23.4.

If the transmit rate is r bytes per sec, theoretically traffic should be injected at a rate of 1 byte every $1/r$th of a second. Since IP packets are an integral number of bytes, multiple bytes might have to be sent together approximating the theoretical rate, as measured over a longer interval. Consider an example where the transmit rate r is 1.2 Mbps. This implies that a byte needs to be sent every 6.7 μsec. Hence, a 1500-byte packet will be sent every 10 millisec or perhaps 500 bytes every 3.3 millisec, thus averaging 1.2 Mbps over a longer time interval.

The leaky bucket algorithm can be easily implemented using a bounded first-in, first-out (FIFO) queue, a timer, and a counter X. The timer expires every second and increments the counter by r. The size of the first packet in queue P is compared with the counter value X. If $X > P$, the counter is updated with $X = X - P$ and the packet is transmitted. Subsequent packets in the queue can be transmitted as long as the counter value X is greater than the size of the packets. When the counter value X is less than the next packet in the queue, transmission is stopped until the next second. In the next second, the counter value is updated with $X = r$ and the transmission of traffic continues.

If there are packets waiting in the queue when a new packet arrives, the size of the new packet P is added with the sum of the sizes S of all packets in the queue. If $P + S > b$, it represents a queue overflow (similar to bucket overflow) and the packet is discarded. Otherwise, the packet is queued.

Thus, the leaky bucket algorithm manages the flow of data into the network such that the packets are not forwarded at a rate greater than the network can or is willing to absorb. The bucket size b bounds the amount of delay that a packet can experience at this traffic shaper. The transmit rate r and bucket size b are typically user configurable.

A significant drawback of the leaky bucket algorithm is that it strictly enforces the average rate of a flow, no matter how bursty the traffic is. As many Internet applications are bursty in nature, it helps to allow some burstiness in a flow.

23.4.2 Token Bucket

A token bucket provides a mechanism that allows a desired level of burstiness within a flow by limiting its average rate as well as its maximum burst size. A token bucket can be viewed as an abstraction of rate of transfer expressed as a relationship between a committed burst size B, a committed information rate, CIR, and a time interval, T.

$$\text{CIR} = B/T,$$

where

- *Committed information rate (CIR)* specifies the amount of data that can be sent per-unit time interval and represents the long-term average rate at which a flow can send its data into the network. Sometimes this parameter is referred to as the *mean rate*. The mean rate is specified in bits per sec or bytes per sec as IP packets are of variable length.
- *Committed burst size (CBS)* specifies the maximum number of bytes that can be transmitted into the network in an extremely short interval of time. In theory, the committed burst size, as the time interval tends to zero, represents the number of bytes that can be instantaneously transmitted into the network. However, in practice, it is not possible to instantaneously send multiple bytes into the network since there is an upper bound on the physical transmission rate that cannot be exceeded.
- *Time interval (T)* specifies the time per burst.

According to the definition, the rate of the traffic sent over the network, over any integral multiple of the time interval, will not exceed the mean rate. However, within the interval, the rate can be arbitrarily fast. For instance, consider a flow that is constrained to send data at an average rate of 12,000 bits per sec. This flow can send 3000 bits within a time interval of 100 millisec. When considering this 100-millisec interval, it might appear that its average rate is 30,000 bits per sec. However, as long as the flow does not send more than 9000 bits in the second interval containing the 100 millisec in which it sent 3000 bits, it will not exceed the average of 12,000 bits per sec. Before examining the token bucket algorithm, let us consider an example of how a token bucket is specified.

Example 23.3 *Specifying a token bucket.*

Assume that the traffic needs to be sent into the network at a mean rate CIR of 2.4 Mbps. If a burst for a duration of 10 millisec (= 0.01 sec) needs to be sustained, the CBS can be calculated using the token bucket definition as

$$\text{CBS} = \frac{2{,}400{,}000 \text{ bits/sec} \times 0.01 \text{ sec}}{8 \text{ bits/byte}},$$

which yields 3000 bytes. Thus, the token rate is 300,000 (=2,400,000/8) bytes per sec, the CBS is 3000 bytes, and the token interval (T) is 10 millisec. Therefore, the token generator credits the token bucket with 3000 bytes worth of tokens every 10 millisec. This indicates that the conforming traffic will, in the worst case, come in 100 bursts per sec of 3000 bytes each and at a CIR not exceeding 2.4 Mbps. ▲

Algorithm

Based on the token bucket definition, an algorithm can be devised that controls the rate of the traffic in such a way that over a long time period the average allowed rate approaches the desired mean rate CIR asymptotically and over a short time interval the burst size of the traffic is upper bounded by bucket size CBS. The algorithm assigns a token bucket for each flow that consists of a bucket which can hold up to CBS tokens. Each token can be considered as

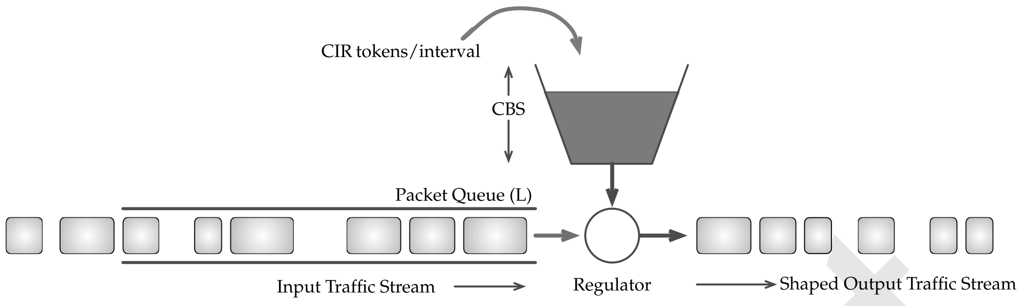

FIGURE 23.5 Token bucket algorithm.

a permission for the source to transmit a certain number of bits into the network. These CBS tokens in the bucket represent the allowed burst size. New tokens are added to the bucket at the rate of CIR tokens per token interval. If the bucket contains less than CBS tokens when a new token is generated, it is added to the bucket; otherwise, the newly generated token is dropped and the token bucket remains filled with CBS tokens. A conceptual illustration of the token bucket algorithm is shown in Figure 23.5.

Now, continuing with the algorithm, the packets arriving in a flow are placed into a packet queue that has a maximum length of L. If the flow delivers more packets than the queue can store, the excess packets are discarded. When a packet of size P bytes arrives in a flow, one of the following cases will apply.

- If the token bucket is full, P tokens are removed before the packet is transmitted into the network.
- If the token bucket is empty, the packet is queued until P tokens are accumulated in the bucket. Eventually, when the bucket contains P tokens, that many tokens are removed from the bucket and the packets are transmitted into the network.
- Finally, consider the case when the token bucket is partially filled with, say X tokens. If $P \leq X$, then P tokens are removed from the bucket and the packet is sent into the network. If $P > X$, the packet is queued and it waits until the remaining $P - X$ tokens are accumulated. Once the bucket accumulates the required P tokens, they are removed from the bucket and the packet is forwarded into the network.

If packets arrive in short bursts, up to CBS tokens would be available in the bucket, and, thus, up to CBS bytes would still get through. As a consequence, the maximum burst size is limited to at most CBS bytes for the flow. Furthermore, as the token replenishment rate is CIR tokens every token interval T, the maximum number of bytes that can enter the network over any time period of t is $\text{CBS} + t \times \text{CIR}/T$. Thus, the token replenishment rate, CIR, serves to limit the long-term average rate of bytes entering the network. The length of the packet queue bounds the delay incurred by a packet. Now let us consider an example that illustrates how traffic in a flow gets shaped using token bucket.

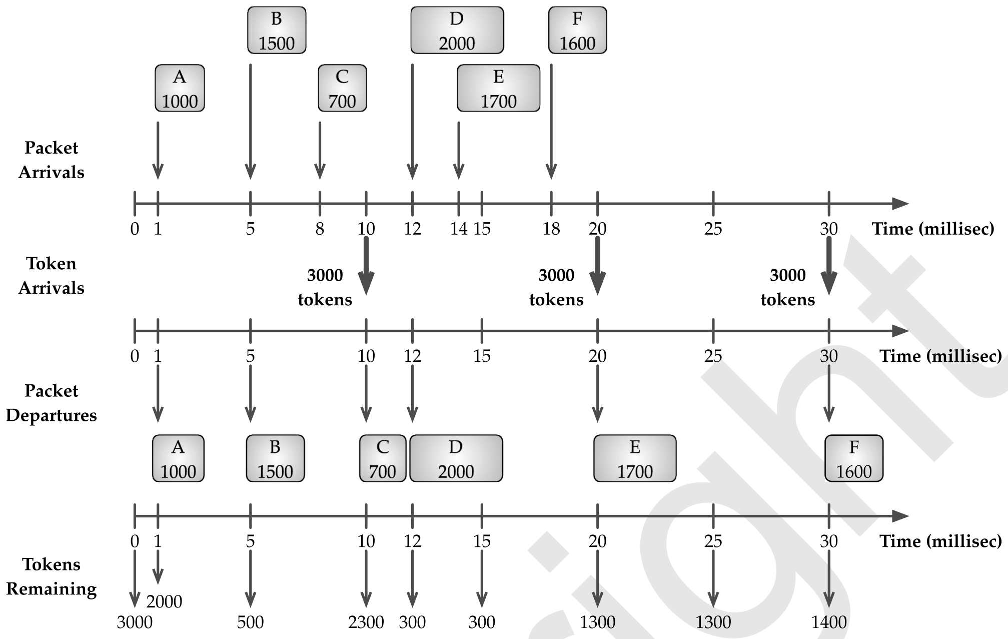

FIGURE 23.6 Shaping traffic using the token bucket algorithm.

Example 23.4 *Shaping traffic using token bucket.*

Consider the token bucket specification of Example 23.3. In this case, the CBS is 3000 bytes. Hence, the token bucket receives 3000 tokens every 10 millisec so that the CIR does not exceed 2.4 Mbps.

Now let us consider a sequence of packets of a flow as shown in Figure 23.6. It shows the time of arrival and departure for packets and the time of arrival of tokens into the bucket. The packets are referred to using the alphabetical letters A through F along with the size of the packet.

To start, assume that the token bucket contains 3000 tokens when time $t = 0$. At time $t = 1$ millisec, packet A of size 1000 bytes arrives. Since the the bucket contains enough tokens, the packet is immediately transmitted and 1000 tokens are removed from the bucket leaving only 2000 tokens. Now packet B of size 1500 bytes arrives at $t = 5$ millisec which is also transmitted immediately since 1500 tokens are available. After the packet B is transmitted, the bucket is left with 500 tokens. At $t = 8$ millisec, packet C of size 700 bytes arrives. Since there are not enough tokens in the bucket, packet C is queued.

A fresh set of 3000 tokens is credited to the bucket at $t = 10$ millisec. However, there are 500 tokens already and, thus, only 2500 of the new 3000 tokens are added so that the total does not exceed the burst size of 3000 tokens. The rest of the tokens are discarded. Now the packet C can be transmitted as the bucket has enough tokens. Thus, it departs at $t = 10$ millisec and the bucket is left with 2300 tokens after deducting 700 tokens for packet C. Packet D of size 2000 bytes that arrives at $t = 12$ millisec is immediately transmitted leaving only 300 tokens in the bucket. Now packet E of size 1700 bytes and F of size 1600 bytes, which arrive at $t =$ 14 millisec and $t = 18$ millisec, respectively, are queued since not enough tokens are available.

At $t = 20$ millisec another fresh set of 3000 tokens is credited to the bucket. Out of these, 300 tokens are discarded since 300 tokens from the previous interval are present in the bucket. Now packet E is transmitted, as sufficient tokens are available after a delay of 6 millisec. The remaining 1300 tokens are not enough for packet F. Thus, it has to wait until the next token interval when a new set of tokens arrives at $t = 30$ millisec. Finally, packet F is transmitted at $t = 30$ millisec after a delay of 12 millisec. We can see that a total of 8500 bytes was transmitted in an interval of 30 millisec and, thus, the rate of transfer is 2.26 Mbps ($= 8500 \times 8/(30 \times 10^{-3})$), which is less than the mean rate of 2.4 Mbps. ▲

Implementation

The token bucket algorithm can be easily implemented using a counter and a timer per flow. The timer expires every token interval T and increments the value of the counter by CIR tokens. This value is never allowed to go more than the CBS. When a packet arrives and if the counter value is greater than the packet size, it is transmitted, and the counter is decremented the packet size. If the packet size is greater than the counter value, the packet is queued until enough value is accumulated in the counter when the timer expires in subsequent intervals. The disadvantage of this approach is that it might not be scalable, as the number of flows increases and so does the number of timers.

Alternatively, the implementation uses a counter and a variable that stores the arrival time of the last packet. It uses the interarrival packet time to compute the number of tokens. Whenever the next packet arrives, the time between packet arrivals is calculated. If t_1 is the time of arrival of the previous packet and t_2 is the time of arrival of the current packet, then the number of tokens to be credited is calculated as $(t_2 - t_1) \times$ CIR. If there are N flows to be shaped, it requires $2N$ integers and, thus, the implementation scales linearly. We shall illustrate the use of this method in Example 23.6 when discussing traffic policing.

23.5 Traffic Policing

Traffic policing allows us to control the maximum rate of traffic for an individual flow or an aggregate of flows. It is typically used at the edge of the network to limit the traffic entering network. Let us illustrate the need for traffic policing using a few examples.

Example 23.5 *Limiting the data rate using traffic policing.*

There are several scenarios in which an ISP might have to limit the data rate:

- *Provision subrate delivery:* The enterprise customer E might require a bandwidth that might be a fraction of a T1 interface (384 Kbps). In such cases, the ISP might have to deliver the full T1 connection and use a traffic policing mechanism to restrict the bandwidth to what the customer has requested. This is because it might be less expensive than some of the alternate options like using Frame Relay. When the customer requests more bandwidth, the ISP can do an upgrade by simply modifying the policer to allow more traffic.

- *Rate control of P2P traffic:* Use of P2P applications like Kazaa and Bit torrent by residential customers C_1 and C_2 can consume a lot of bandwidth, affecting critical business applications running on the ISP network. This could incur huge penalties for the ISP if SLAs are not met. If the contents of these P2P applications are fetched from content providers CP_1 and CP_2, then the ISP needs to enable traffic policers at routers R_4 and R_7. These traffic policers distinguish the P2P traffic from normal traffic and restrict the rate of such traffic entering the ISP network. The P2P traffic can be either dropped or marked as lower priority before being admitted into the network.
- *Improved network security:* The ISP might need to police the ping traffic to a reasonable amount that would increase the overall security of the network from DOS attacks. The policing of ping traffic does not preclude someone from launching ICMP-based attacks, but to some extent drops a good portion of the attack traffic as it enters the ISP's network, thereby reducing its impact on the internal network resources. ▲

If policing determines that the traffic in a flow falls within the rate parameters (in-profile), then the packet is transmitted into the network. However, if the traffic exceeds the rate parameters (out-of-profile), then an action is applied based on the policy defined. The action can be either to drop packets immediately or transmit the packets to the network but marked with a lower priority. We refer to the action of dropping packets as *traffic policing* and the action of lowering the priority of out-of-profile packets as *packet marking*, which is discussed in detail in Section 23.6.

Traffic policing uses the same token bucket algorithm described in Section 23.4.2 for regulating traffic. Unlike shaping, where the token bucket algorithm uses a queue for storing packets when sufficient tokens are not available, traffic policing does not use a queue as it discards packets in such cases. The token bucket algorithm itself does not determine whether the packets must be queued or discarded. It just discards tokens when the bucket is full, and when there are no sufficient tokens available for a packet, it delegates the action of handling the packet depending on shaping or policing. Let us examine a sequence of packets to understand how the token bucket algorithm is used for traffic policing.

Example 23.6 *Policing traffic using token bucket.*

Consider policing the traffic shown in Example 23.4. Assume that there are 3000 tokens at $t = 0$. When packet A of size 1000 bytes arrives at $t = 1$ millisec, 300 (= 1 millisec $\times$ 3000/10 millisec) tokens are generated, which are discarded since the bucket is already full. Since there are enough tokens available in the bucket, the packet is transmitted and 1000 tokens are removed from the bucket, leaving 2000 tokens. Now packet B of size 1500 bytes arrives at $t = 5$ millisec, and it is 4 millisec since the last packet arrived. Hence, a new set of 1200 (= 4 millisec $\times$ 3000/10 millisec) tokens is credited. However, 200 tokens are discarded as the bucket overflowed after a total of 3000 tokens. As the bucket contains sufficient tokens, packet B is transmitted and 1500 tokens are left in the bucket (Figure 23.7).

The same procedure is repeated for packet C of size 700 bytes and packet D of size 2000 bytes. After packet D is transmitted at $t = 12$ millisec, the bucket is left with 900 tokens. When packet E arrives 2 millisec later, 600 (= 2 millisec $\times$ 3000/10 millisec) new tokens are generated. However, a total of 1500 tokens is not enough to transmit packet E and, thus,

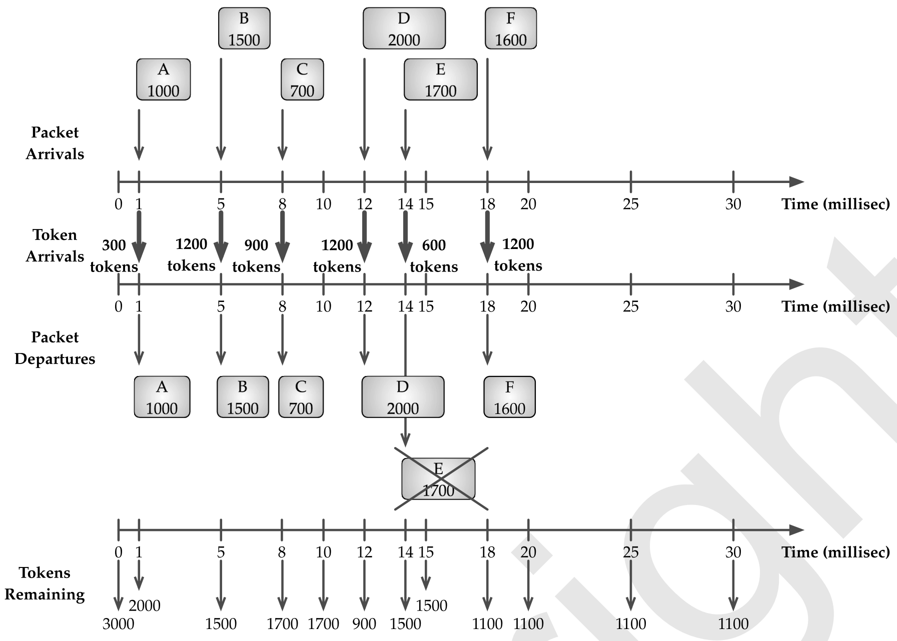

FIGURE 23.7 Policing traffic using the token bucket algorithm.

it is dropped. After 4 millisec, when packet F arrives at $t = 18$ millisec, a new set of 1200 ($= 4$ millisec $\times$ 3000/10 millisec) tokens is added to the bucket totaling 2700 tokens. This is enough to transmit packet F. ▲

23.5.1 Comparing Traffic Policing and Shaping

Even though shaping and policing control the rate of the traffic, there are substantial differences, which are summarized in [139]. The key difference is that traffic shaping not only limits the flows to a specified rate but also buffers the nonconforming traffic. The result of shaping is that the traffic is output at a smoothed rate. In contrast to shaping, policing propagates bursts and does not buffer nonconforming traffic; instead, it just simply drops the packet when the traffic rate exceeds the specified limits. The result is an output rate that appears as a sawtooth with crests and troughs. This is illustrated in Figure 23.8. The leftmost figure indicates the variation of input traffic rate over time, the middle figure shows the output rate of shaped traffic, and the rightmost figure shows the output rate of policed traffic.

Since shaping needs to buffer nonconformant traffic, it requires a queue with sufficient amount of memory. Typically, traffic shaping is applied for outbound traffic. However, since policing drops nonconformant traffic, it can be applied to inbound or outbound traffic in a router. Traffic shaping introduces delay as it buffers nonconforming traffic. However, policing does not introduce any additional delay or jitter in the traffic.

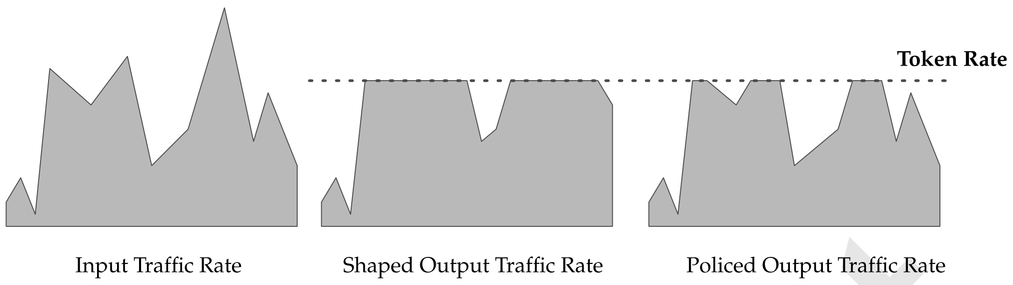

FIGURE 23.8 Comparing traffic policing and shaping.

23.6 Packet Marking

The action of lowering the priority of out-of-profile packets by modifying one or more header bits is referred to as *packet marking*. Marking out-of-profile packets serves as a hint to the next-hop downstream routers to handle them differently. For example, an access router that polices the traffic can mark a packet by changing its header bits so that core routers give the packet a higher probability of dropping during periods of congestion while they continue to deliver in-profile traffic.

While traffic policing takes a hard approach of dropping out-of-profile packets, packet marking uses a softer approach by postponing the decision to discard a packet to a downstream router, which is congested. As a result, when the available bandwidth is plentiful, service providers can deliver higher levels of service to all subscribers. At the same time, they are able to protect the shared network resources and meet the subscriber SLAs during periods of scarce bandwidth.

Packet marking can also be implemented using token bucket. When a packet arrives and if there are enough tokens available for the packet to be transmitted, it is considered as in-profile; otherwise, it is out-of-profile. If different rates need to be enforced for different flows then multiple token buckets with the appropriate bucket size and token replenishment rate can be run simultaneously. When a packet arrives, it is associated with a flow using packet classification based on the policies or rules configured in the router. Depending on the flow, the appropriate token bucket is chosen and the packet is marked as in-profile or out-of-profile. Based on the actions configured in the rules, the packets are subsequently dropped or marked. This is shown in Figure 23.9.

After marking, the packets are passed to output queueing and scheduling and finally transmitted on the output link. If the output link experiences congestion, then the priority is to drop the out-of-profile packets. A simple approach to drop more out-of-profile packets is to reduce their priority by segreating them from in-profile packets and assigning them to lower-priority queues. The in-profile packets are assigned to higher-priority queues. However, this can result in reordering of out-of-profile packets relative to in-profile packets. The reordering occurs when an out-of-profile packet is scheduled ahead of an in-profile packet. This scenario is shown in Figure 23.10. Hence, the receiving application will see the out-of-profile packet first, assuming that the intermediate routers did not drop the packet.

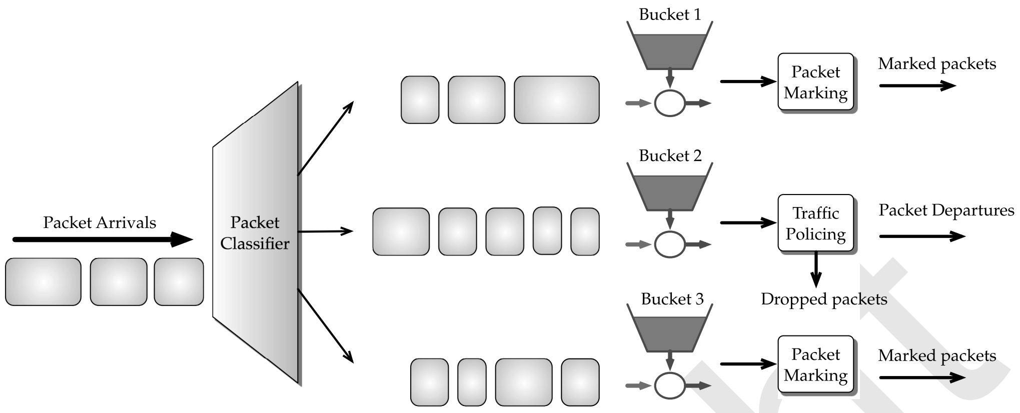

FIGURE 23.9 Use of multiple token buckets simultaneously.

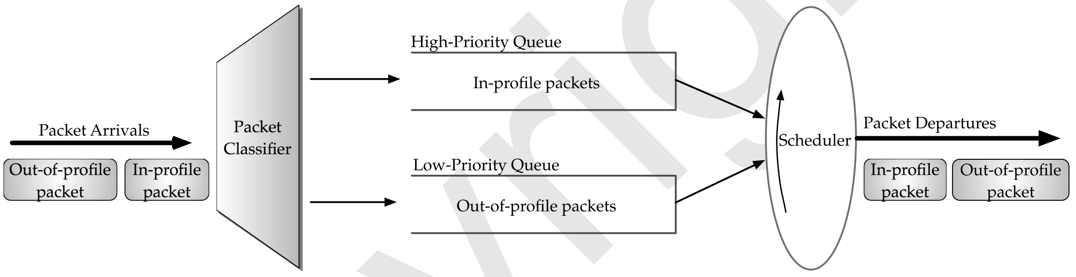

FIGURE 23.10 Reordering of in-profile and out-of-profile packets because of separate queues.

The preferred approach is to use a single queue and associate different RED drop probability for different types of packets. An aggressive RED drop probability is assigned for out-of-profile packets and a conservative RED drop probability for in-profile packets.

23.6.1 Graded Profiles

So far, our discussion assumed that a single average rate and a burst size are used to govern the traffic in a flow. However, this might be insufficient for expressing the traffic into different grades based on its temporal characteristics. Such grading of traffic allows the application of different types of marking or a combination of marking or policing for each grade. For instance, a graded profile might specify that the traffic exceeding a rate of M bytes per second is simply marked and if the excess traffic rate becomes greater than N bytes per second, it should be immediately discarded.

When the traffic is graded, colors can be used to describe marking of packets. That is, the color of the packet identifies whether it is conforming to the traffic profile. For example, a green packet means it is conforming to the committed rate of the traffic profile; a yellow packet means that it is not conforming to the committed rate, but meets the excess rate of the traffic profile; however, a red packet does not meet the committed nor the excess rates

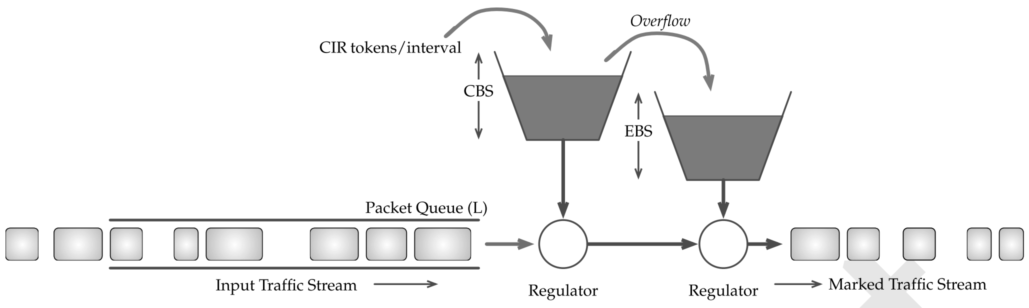

FIGURE 23.11 Single-rate three-color marking.

of the traffic profile. Then, green packets are processed as specified in the SLA and are not candidates for discarding. Yellow packets are candidates for discarding, typically, only if the network is congested. Red packets are immediately discarded.

Next we describe two marking algorithms: single-rate tricolor marking (srTCM) and two-rate tricolor marking (trTCM). For these, we need to use a few terms: committed information rate (CIR), committed burst size (CBS), excess information rate (EIR), and excess burst size (EBS). We have already discussed CIR and CBS. EIR specifies the average rate that is greater than or equal to the CIR; this is the maximum rate up to which packets are admitted into the network. EBS is the maximum number of bytes allowed for incoming packets. Packets are in-profile if they meet CIR and CBS ("CIR conformant") while they are out-of-profile if they meet EIR And EBS ("EIR conformant").

23.6.2 Single-Rate Tricolor Marking

A single-rate traffic profile uses three parameters, CIR, and the two burst sizes, a CBS and an EBS. This can be implemented by two token buckets C and E and both share' the common rate CIR. The maximum size of the token bucket C is CBS and the maximum size of the token bucket E is EBS. This is shown in Figure 23.11.

To start with, the token buckets C and E are initialized with CBS and EBS, respectively. Thereafter, tokens arrive at CIR times per second. The newly arrived token is added to bucket C if its token count is less than CBS. Otherwise, the token is added to bucket E, if it has less than EBS tokens. If buckets C and E already contain CBS and EBS tokens, the token is discarded.

Assume that the token buckets C and E contain P_C and P_E tokens, respectively. When a packet of size B bytes arrives, the packet is handled as follows:

- If $B \leq P_C$, the packet is colored green and the number of tokens in bucket C is decremented by B and the packet is transmitted into the network.
- If $B \leq P_E$, the packet is colored yellow and decrements the number of tokens in bucket E by the packet size B. The packet is allowed to enter the provider's network.
- If the packet is colored red, none of the existing tokens in either of the buckets is decremented.

Example 23.7 *Walk through the algorithm.*

For this example, assume that the committed rate is configured to be 2000 bytes per sec and the CBS of 2000 bytes. Similarly, the excess rate and excess burst size are also configured at 2000 bytes per sec and 2000 bytes, respectively.

The token buckets C and E start with their respective burst sizes of 2000 bytes each. If a 900 byte packet arrives, the packet conforms because enough bytes are available in the committed token bucket C. Hence, the packet is declared as CIR conformant, colored green, and then transmitted. At the same time, 900 bytes are removed from token bucket C, leaving only 1100 bytes.

If the next packet arrives 0.25 sec later, 500 (= 0.25×2000) bytes are added to token bucket C, leaving a total of 1600 bytes. If the next packet size is 1800 bytes, the packet does not conform, as only 1600 bytes are available in token bucket C.

The exceed token bucket, which started full at 2000 bytes (as specified by the excess burst size), is then checked for available bytes. Because enough bytes are available in token bucket E, the packet is colored yellow and admitted into the network. Subsequently, 1800 bytes are removed from the excess bucket E, leaving only 200 bytes.

If the next packet arrives 0.4 sec later, tokens equivalent to 800 (= 0.4×2000) bytes are generated. Of these 800 bytes, only 400 bytes are added to token bucket C as the maximum number of tokens is not allowed to exceed 2000 bytes. Since the remaining 400 bytes will overflow token bucket C, these are placed in token bucket E, leaving a total of 600 bytes.

If the arriving packet is 2000 bytes, the packet conforms because enough bytes are available in token bucket C. The packet is colored green, transmitted into the network, and 2000 bytes are removed from the conform token bucket (leaving 0 bytes).

If the next packet arrives 0.2 sec later, 400 (= 0.2×2000) bytes are added to the token bucket. Therefore, token bucket C now has 400 bytes. If the arriving packet is 800 bytes, the packet does not conform because only 400 bytes are available in bucket C. Similarly, the packet does not exceed because only 600 bytes are available in token bucket E. Therefore, the packet is colored red and discarded. ▲

23.6.3 Two-Rate Tricolor Marking

A two-rate traffic profile is described by four parameters: an EIR and its associated EBS, and a CIR and its associated CBS. EIR and CIR are measured in bytes per second and EIR must be greater or equal to CIR. Similarly, EBS and CBS are also measured in bytes and should be greater than zero. In addition, both of them should be configured to be equal to or greater than the size of the largest possible IP packet in the flow.

Two-rate tricolor marking can be used for ingress policing of a service where a peak rate needs to be enforced separately from a committed rate. Such a two-rate bandwidth profile can be also implemented by using two token buckets. One bucket, referred to as the committed or C bucket, is used to determine CIR-conformant and in-profile packets, while the other bucket, referred to as the *excess* or E bucket, is used to determine the EIR-conformant and out-of-profile packets.

Similar to srTCM, two token buckets C and E are used. The token buckets C and E are initialized with CBS and EBS, respectively. Thereafter, the token count in bucket C is

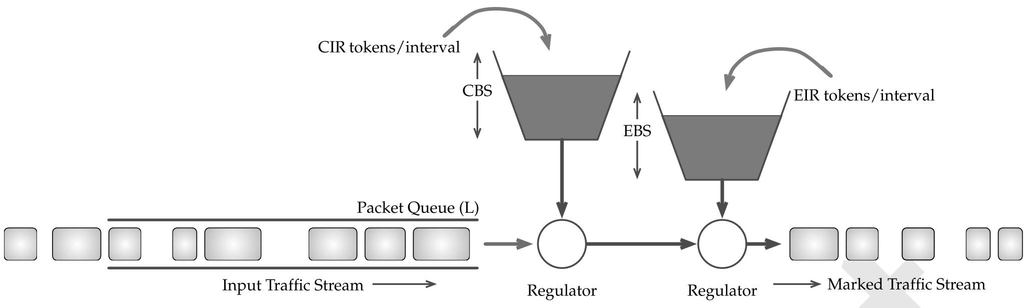

FIGURE 23.12 Two-rate three-color marking.

incremented by CIR times per second up to CBS. Similarly, the token count E is incremented by EIR times per second up to EBS. The two-rate three-color marking is shown in Figure 23.12.

When a packet of size B arrives and if buckets C and E contain the tokens P_C and P_E, respectively, the packet is handled as follows:

- If $P_E < B$, the packet is colored red and discarded.
- If $P_C < B$, the packet is colored yellow, admitted into the network, and B tokens are removed from bucket E.
- If the packet is colored green, it is transmitted into the network and the number of tokens in both the buckets is decremented by B.

23.7 Summary

In this chapter, we studied the importance of traffic shaping, policing, and packet marking as important tools available for the service provider to provide differentiated services in a cost-effective manner. We discussed how traffic shaping limits the rate of traffic sent into a network. We described two forms of traffic shaping: traffic smoothing and traffic burst shaping. We then studied in detail about how leaky bucket and token bucket algorithms can be used to regulate traffic.

We continued our discussion about traffic policing, which limits the rate of traffic by discarding excess packets. We then studied packet marking, which takes a softer approach by lowering the priority of packets and admitting them into the network under the assumption that the downstream routers will lower the priority of packets if congestion is imminent. We concluded the chapter with the discussion of two algorithms for packet marking.

Further Lookup

A detailed discussion about SLAs can be found in the book [719] devoted to this topic. Entire sections of Chapter 3 of [26] are dedicated to a discussion of traffic shaping, policing, and packet marking. The white paper [630] from Juniper Networks provides an excellent discussion of shaping and policing. The detailed documentation from Cisco Systems [138] provides an overview of shaping and policing. Another document from Cisco Systems [139] elaborates on the differences between shaping and policing.

Turner [706] first proposed the idea of the leaky bucket algorithm. Detailed descriptions about the leaky bucket and token bucket algorithms can be found in [683]. Single-rate three-color marking is described in RFC 2697 [291] and two-rate three-color marking in RFC 2698 [292].

An important direction for traffic shaping is stochastic traffic shaping (as opposed to deterministic traffic shaping); for an extensive discussion on stochastic traffic shaping, see [118].

Exercises

23.1. Explain two different ways to shape traffic.

23.2. Why is the idea of token bucket helpful in traffic policing?

23.3. Explain the connection between packet scheduling and traffic policing at a router.

23.4. Consider Example 23.4. Work through the example, if the token bucket size is changed to 2000 tokens.

23.5. How are service level agreements related to differentiated services?

23.6. What is the relation between traffic policing and traffic marking?